Population synthesis study of the binary Gaia white dwarf population

Treball de fi de grau
Grau en Enginyeria Física

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Author: Aina Ferrer i Burjachs [1]
Codirectors: Santiago Torres Gil [(1,2)] & Alberto Rebassa Mansergas [(1,2)]

[(1)]Departament de Física, Universitat Politècnica de Catalunya, c/Esteve Terrades 5, 08860 Castelldefels, Spain
[(2)]Institute for Space Studies of Catalonia, c/Gran Capità 2–4, Edif. Nexus 104, 08034 Barcelona, Spain

"I am going to look at the stars.

They are so far way and their light takes so long to reach us...

all we ever see of stars are their old pictures."

Dr. Manhattan - "Watchmen"

Contents

Acknowledgements

Firstly, I would like to thank Dr. Santiago Torres Gil and Dr. Alberto Rebassa Mansergas for all what they taught me, their advise corrections and the support during the realization of this project. Especially grateful for the opportunity that the results of my work go beyond this TFG. As well, I would also like to thank to Dr. Roberto Raddi for providing the models to calculate the *Gaia* magnitudes and make this project a reality.

I would also like to thank Dr. Jordi Jose Pont for introducing me into the fascinating world of astrophysics.

Als meus pares pel seu suport, ajuda i companyia. Al Link i a la Midna per ajudar-me a superar els mals dies. I en especial al meu avi Josep per fer-me sempre costat.

Abstract

We conduct a population synthesis study of the white dwarf population within 100 pc from the Sun. Such simulations are based on a detailed Monte Carlo code, which provides synthetic samples of both single and binary stars for different input parameters. Synthetic outcomes are compared with observed data obtained from the recent *Gaia* mission in its early data release 3. In a first step we compute the outcome of the synthetic sample in the *Gaia* magnitude system, introduce the photometric and astrometric errors according to *Gaia* prescriptions and apply the corresponding selection cuts. Then we analyze the general statistical properties, such as the number-counts and percentages of the complete range of sub-populations which contain at least one white dwarf component, i.e. single and merger white dwarfs, white dwarf plus main sequence binaries, and double white dwarfs; in its two modalities, resolved and unresolved systems. The comparison between the observed and the several outcomes of the simulations are mainly based of the fractions of the different sub-samples and their distribution within the Hertzsrpung-Russell diagram. To assist the statistical analysis of observed and synthetic distributions we introduce a generalization of the Kolmogorov-Smirnov test applied to two-dimensional density distributions. Our analysis reveals that any of the models analyzed in this project perfectly agrees with the observed percentages of all the different sub-populations. However, a flat or even a decreasing initial mass ratio distribution seems to be a reasonable first guess, while an increasing relation can be discarded. Further work is needed expanding the number of models and input parameters.

List of Figures

List of Tables

Introduction

White dwarfs (WD) are the most common stellar remnants and carry an incredible amount of valuable information about our Galaxy's past history, formation and evolution. WDs can be found not only isolated but also as part of binary systems (Althaus et al., 2010). These objects are of vital importance for the understanding of some of the top actual astrophysical problems, such as the age-metallicity relation of the Galaxy or the formation of Type Ia supernovae, among other examples. The *Gaia* mission has gathered an unprecedented wealth of information of our galaxy, in particular the third Gaia data release (Gaia Collaboration 2020) has identified about 350,000 high-confidence WD candidates. This vast amount of data is expected to help us constraining some of the previous mentioned unknowns.

The present work focuses on the study of a sample of single and WD binaries in the solar neighbourhood, in particular a volume of 100 pc radius. As we will explain in Chapter 3, this is thought to be the largest and most complete volume-limited sample of WDs available from *Gaia* observations (Torres et al., 2005; Jiménez-Esteban et al., 2018). The ultimate goal is to compare this observed data with the outcome of several simulations that we perform based on a Monte Carlo code that generates synthetic samples of both binary and single stars in the solar neighborhood. The code, developed by the GAA (Grup d'Astronomia i Astrofísica) at UPC in Castelldefels, has been extensively used in the study of several single and binary system populations and different Galactic components such as the thin and thick disk, the halo, the bulge and open and globular clusters (e.g. García-Berro et al., 1999, 2010; Torres et al., 2001, 2016, 2018, 2020; Camacho et al., 2014; Cojocaru et al., 2017). This long trajectory has consolidated the robustness of the code and guarantees the solidity of the results.

The population synthesis code makes use of a wide variety of assumptions and input parameters to build the Galactic model and to evolve its constituent stars. We will change only two of these key parameters: the initial mass ratio distribution (defined as the quotient between the mass of the primary and secondary stars), and the binary fraction (the proportion of binary and single systems); both will be explained in detail in Chapter 2. The decision on which of the assumed models yields synthetic data in better agreement with the observations will be based on the comparison of various characteristics of the populations, for instance the proportion of certain sub-populations or their stellar parameter distributions, as described in Chapter 5. In the statistical comparison between distributions, in particular the two dimensional distribution of objects within the Hertzsprung-Russell diagram, we will make use of a multivariate Kolmogorov-Smirnov test (Peacock 1983).

The project is organized as follows. In Chapter 1 we present a general outlook of the theoretical background on stellar and binary evolution, with special emphasis on WDs. We also provide information about the different Galactic components that constitute the solar neighbourhood. In Chapter 2 we provide a brief explanation of the population synthesis code to simulate the single and binary star populations. In Chapter 3 we present the observed sample and the outcome of the simulations and preform a preliminary comparison between them in order to decide which adjustments or corrections are required. In Chapter 4 we describe in more detail all the different sub-populations and analyse those with at least one WD. This allows us to familiarize with the different sub-populations and have a first approach on the relations between the observed and simulated samples. In Chapter 5 we compare different models with the observed sample in order to try to reach a conclusion on which one fits best with observed population. Finally, a summarize of the main results and future work is discuss in 5.3.

Chapter 1

Theoretical background: basic overview

In this Chapter we provide a brief introduction into stellar evolution, specially focused in understanding the phases from when a star is born until it becomes a white dwarf. In addition, we give a general description of the evolution of binary stars, also focused on white dwarfs as their stellar remnants. This will be of extreme use to build the population synthesis code described in the next Chapter.

1.1 Single stellar evolution

1.1.1 From ZAMS to the TPAGB phase

It is widely accepted to start counting the age of a star from the zero-age main-sequence (ZAMS), that is, once nuclear reactions begin to consume hydrogen (H) in its core (e.g., Hurley et al. 2000, Kippenhahn 2012). Previous stages take place in a relatively short time and the energy source is governed by gravitational contraction. Therefore, the ZAMS is considered the beginning of the star's evolution and the main-sequence (MS) stage is defined as the stage in which the H in the core burns into helium (He). Once there is no more H in the core the MS stage finishes and the star is composed by a He core and outer H-burning shells.

The post-MS evolution is characterized by an increase in the radius of the convective H-rich envelope of the star, decreasing its effective temperature ($T_{\rm eff}$), and consequently displacing the star horizontally into the right of the Hertzprung-Russell (HR) diagram (which displays the luminosity of a star as a function of its effective temperature; see Figure 1.1) and transforming it into a red giant (RG). Such process of expansion occurs quickly enough to leave a gap in the observed stars between the MS and RG branch known as the Hertzprung gap. In the RG phase the inner core shells contract increasing the temperature in the center and producing the He ignition. The core helium burning (CHeB) produces carbon (C), oxygen (O) and neon (Ne). During this process the star moves back near the MS branch until the He is finished in the core. Then a similar process as when the H was completely burned takes place, moving the star into the asymptotic giant branch (AGB). The star consists of a CO core surrounded by a He and H shells.

The AGB is generally divided into two parts, the early AGB (EGB) and the thermally pulsating ABG (TPAGB). In the EAGB the degenerate CO core is surrounded by a He-burning shell and a H shell. Eventually the He-burning layer caches up with the H shell, igniting it. Then the He shell becomes unstable, resulting in a giant release of luminosity that causes the star to expand and, as before, the H stops burning. The process can be repeated, the star contracts and the H shell reignites, He shell flash occurs and the subsequent expansion of the star extinguishes the burning

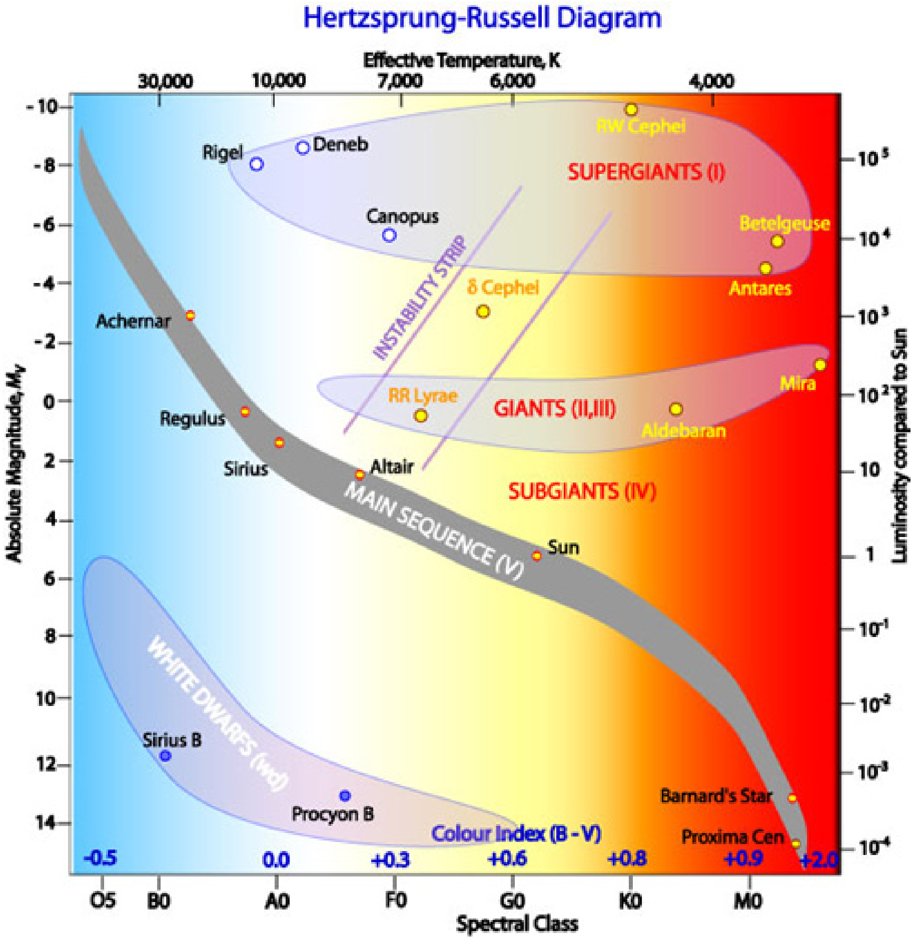


Figure 1.1: Hertzsprung-Russell diagram. The MS is clearly identified as a diagonal path, while white dwarfs are located below MS stars. From: R. Hollow, CSIRO.

shells. This is the TPAGB.

From the AGB on, the future of the star is determined by its mass: for relatively massive stars (above 8-10 M$_\odot$) the CO core reaches high enough temperatures to star a C-burning phase that, after several intermediate steps, it will lead into a supernova explosion, the remnant of which will form a neutron star (NS), or if it was massive enough, a black hole (BH). Lower mass stars do not reach C-fusing temperatures in their cores and evolve through a planetary nebula phase. The outer layers are ejected to the interstellar medium and the massive core is left behind forming a white dwarf (WD).

1.1.2 White dwarf remnant

WD stars are the more common final evolutionary stage. In fact, more than 97% of all MS stars have already become or will become a WD, as will our Sun (Althaus et al. 2010). Following the evolution through the AGB, low and intermediate mass stars begin to loss their envelope and stop any nuclear reactions in the core. Consequently, as they release their stored thermal energy, WDs will slowly cool down and their luminosity will fade with a typical time scale of 10 Gyr (as an illustrative example we show in Figure 1.2 the full evolutionary sequences from ZAMS to WD stage for different initial masses; Althaus et al. 2016). This fact made them a valuable source of information regarding the Galaxy's history (e.g., Isern 2019).

WDs are compact objects with a size comparable to Earth and a mass distribution that peaks at $\simeq 0.6 M_\odot$ and deeply declines towards lower and higher masses. The cores of WDs of masses lower

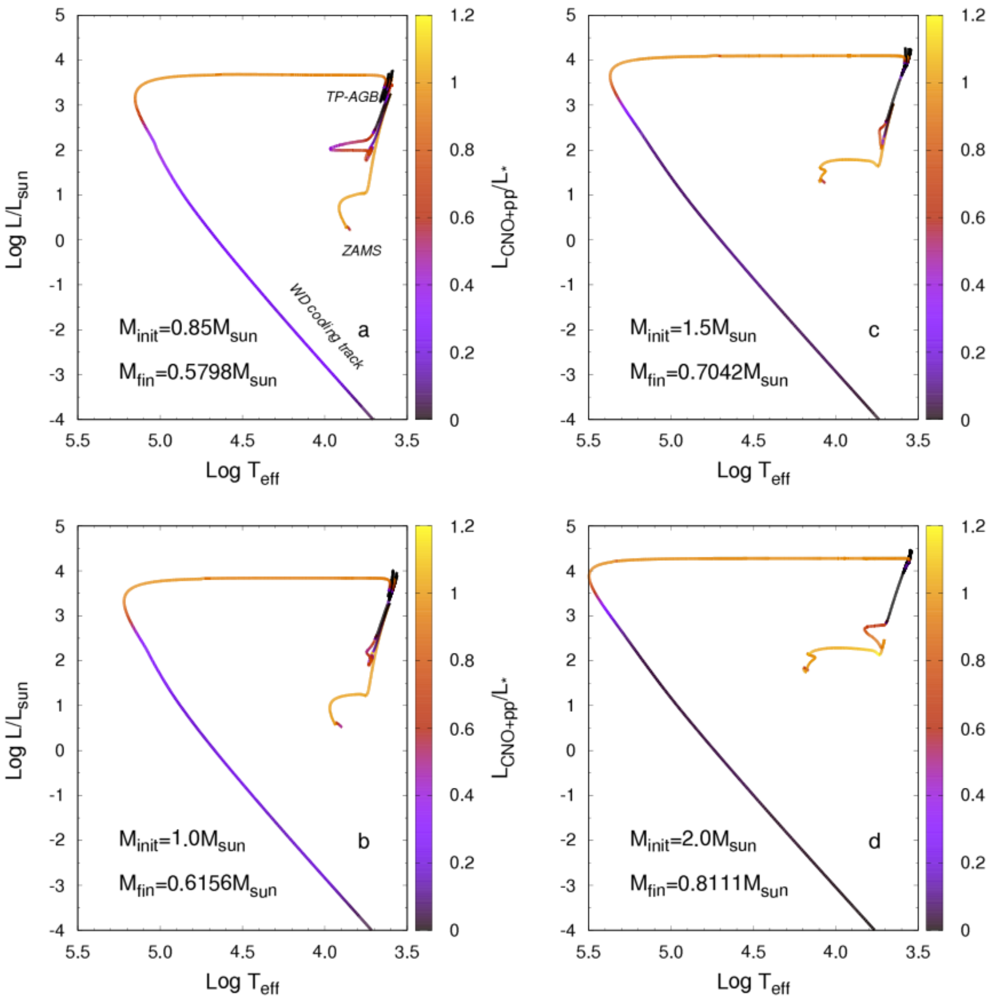


Figure 1.2: Theoretical evolutionary sequences of stars from ZAMS to WD with initial masses 0.85 $M_\odot$, 1.05 $M_\odot$, 1.5 $M_\odot$, and 2.0 $M_\odot$. The color scale on the right shows the fraction of the total luminosity from hydrogen burning. From: Althaus et al. (2016).

than $\simeq$0.45 $M_\odot$ are composed of He and they are believed to form when mass transfer interaction in binary systems truncate the RG phase (Rebassa Mansergas et al. 2011). From $\simeq$ 0.45 to $\simeq$ 1.05 $M_\odot$, the WDs have a CO core, and these are the most abundant. The progenitors of higher-mass WDs manage to ignite C in consecutive C-burning shell flashes that conclude with the exhaustion of C and leaving an ONe core. The mass of a WD can never be higher than a certain threshold, known as Chandrasekhar mass ($\simeq$ 1.4 $M_\odot$), for which no stable WDs are theoretically allowed to exist.

Independently of the type of core the WD has, all of them share the characteristic of having a thin layer of atmosphere. It is generally formed by an inner layer of He and an outer (even thinner) layer of H. For most of the WDs, $\simeq$ 80% of them, the spectral lines we detect are those from the outer H layer at the surface. These are the so called DA WDs, the other 20% that have lost the H envelope and are known as non-DA. Among the non-DA WDs there are several subcategories, which include the DO and DB spectral types, which feature ionized and neutral He lines, respectively. The evolution of DO WDs results in the recombination of the ionized He to transform it into a DB. Other subclasses include DQs, which show C atmospheres, DZs with metal rich spectral lines and DCs with no detectable spectral features.

1.2 Binary evolution

1.2.1 Basic ingredients

Binary systems evolve in a similar way to single stars if the different components do not interfere with each other, that is, if the orbit is large enough so that there is no mass transfer interaction between the two stars (larger than ≃10 AU; Hurley et al. 2002). In the opposite case, if the orbit is relatively close, there are mass transfer interactions between the two components of the system with repercussions in the evolution of the stars and the orbit they follow. In some occasions, even if the two components are considerably separated, some sort of interaction can still occur. In this section we describe briefly these interactions.

A possible outcome is when one of the stars of the system starts losing mass in form of a stellar wind and the companion accretes this material. Such mass loss represents a lost in the angular momentum of the star too, which is gained by the accreting star. In the case of eccentric obits, the process of mass loss and subsequent accretion leads to a transformation of the orbit into a more circular one. Another interaction experimented in binary systems are tidal frictions, which are widely believed to be the cause of the synchronisation between the stellar rotation and orbital motion.

If the binary system contains a star that has achieved the evolutionary state of supernova and explodes, becoming a BH or NS, it generates what is called a velocity kick, accelerating the binary system, in case it has survived the explosion. For those remaining systems, it is almost certain that the kick has changed their shape, as well as a loss of angular momentum due to mass loss, as stated before.

A very common process when the two MS stars in a binary are relatively close is mass transfer via Roche-lobe overflow. Mass transfer is generally initiated when the more massive star evolves into a RG or an AGB, although it can also be initiated by an excessive angular momentum loss that leads to a contraction of the orbit. Because the donor star is generally more massive, mass transfer is dynamically unstable and leads to the formation of a common envelope (CE) (Ivanova et al. 2013). Once the system enters this CE phase, the friction between the two stars and the material of the envelope reduces the orbital energy of the binary, thus reducing the orbital separation. This energy is deposited into the envelope and used to eventually eject it. Once the envelope is dissipated, the result is a close WDMS system with a short period. Another possibility may be that the cores coalesce into a single one and form a single star. Roche-lobe overflow in binary systems with an eccentric orbit, follows in most cases a circularization of the orbit. The outcome of such mass transfer depends on the type of stars that form the system and the radius-mass exponents.

Another channel of binary evolution that does not necessarily involve mass transfer is collisions. Collisions are considered to be a more violent process that coalescence, furthermore, coalescence comes from an advanced stage of a Roche-lobe overflow, while in a collision there is no need to have gone that process.

1.2.2 White dwarf binary systems

Given that in this project we aim at simulating the single and binary WD populations in the Galaxy, it is important to consider all the possible types of WD binaries and single WDs that we can observe.

In binary systems a WD can be found with another WD (we call these double WD systems or DWD) or with a MS (WDMS systems) (Toonen et al. 2017). Depending on their orbital separations, DWDs and WDMS can be resolved or unresolved, meaning that for a resolved system the angular separation is large enough to be able to observationally distinguish the two components of the system, contrary to the unresolved ones, where the angular separation is below a certain threshold that does not allow to detect the two stars separately, but as a single one. It is very likely that such unresolved systems have undergone mass transfer interactions that led to (at least) one CE phase. It is also worth noting here that mergers can take place in binary evolution, as explained in the previous sections. Thus, DWD mergers and WDs whose progenitors were binary systems that merged during their evolution (for example during the CE phase) would be observed as single WDs.

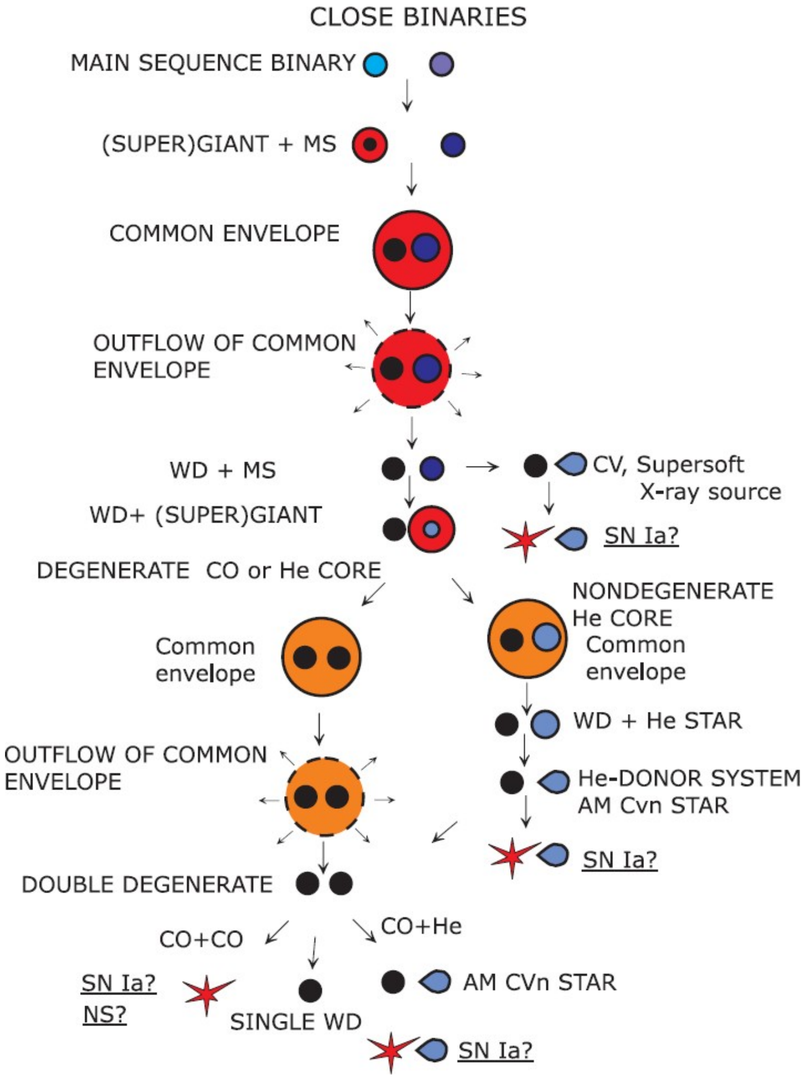


Figure 1.3: Evolutionary channels of close binary systems. From: Postnov & Yungelson (2006)

The evolutionary channels leading to one or another type of binaries containing one or two WDs are very diverse. In Figure 1.3 we show a typical evolution diagram illustrative of the different channels that results in close binary systems. The physics involved in the evolution of those systems implies the use of a wide set of parameters, such as the initial separation, the rate of mass transfers, the treatment of the CE phase, the angular momentum losses equation, and many others. Many of these parameters are poorly constrained. However, the analysis of a significant statistical large sample may provide some clues about them.

Finally, other types of binary systems including a WD are WD+BH or WD+NS binaries, however, in this work we will not take them into consideration since these systems are not observed within the volume we are considering.

1.3 The solar neighbourhood and the Galactic components

As explained before, WDs are generally dim objects, therefore, we see them at relatively close distances. For instance, for a relatively hot WD with an effective temperature of $T_{\rm eff} = 10,000\,$K its absolute magnitude is $M_{\rm G} \approx 12$, which implies that for a magnitude limiting survey of $G \approx 21$ such as *Gaia*, the maximum distance that we will be able to see this WD is $\simeq 630\,$pc. Recalling that the Sun is located at 8.5 kpc from the Galactic center (see the left panel of Figure 1.4), that implies we can only sample a relatively small region of the Galaxy around the Sun, generally called the solar neighborhood.

Thanks to the outcome of the *Gaia* mission, there are now $\simeq$ 350,000 high-confidence WD candidates identified to distances up to a few kpc (Gentile-Fusillo et al. 2021). However, the completeness of the observed sample peaks at a volume of radius 100 pc. Indeed, Jiménez-Esteban et al. (2018) found that the 100 pc sample has a completeness of $\simeq$97%, and about the 94% of them had a parallax error smaller than the 10%. For larger samples the completeness decreases considerably to $\simeq$60% for 250 pc and to $\simeq$22% for 500 pc.

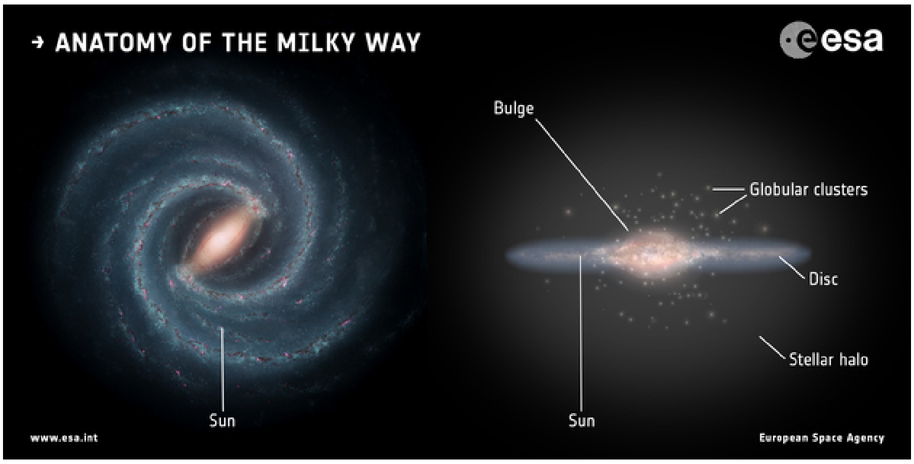


Figure 1.4: An artist's impression of our Galaxy, the Milky Way, showing the location of our Sun in one of its arms and the different components of the Galaxy: bulge, disk and stellar halo. From: Left: NASA/JPL-Caltech; right: ESA; layout: ESA/ATG medialab.

As shown in the right panel of Figure 1.4, our Galaxy is believed to be divided into different components. In the center we find the nucleus surrounded by a spherical bulge. The vast number of stars are in a flatten disk around the nucleus with a thin and a thick component. Finally, there is the halo, which is a region with a spheroid like shape around the main body of the Galaxy with a considerably extension around the center. Given that that the current work is restricted to the solar neighborhood, we will only consider stars placed in the thin and thick disks and the halo, as those are the only components visible from 100 pc from the Sun. The proportion of WDs that we can observed in each region is different, although dominated by the disk component, in particular the thick disk component. Torres et al. (2019) found this distribution to be 74-25-1% for the thin and thick disk and halo components, respectively. Stars in the different components have different properties, for instance thick disk and halo stars tend to be older and colder, and their metallicity is considerably lower than thin disk stars. The tangential velocity is another example, with halo stars being faster than thick disk stars, and these, in turn, being faster than stars in the thin disk.

Chapter 2

The population synthesis code

2.1 The GAA Monte Carlo simulator

Monte Carlo methods are a widely set of numerical calculation techniques based on random sampling with uncountable applications in science and engineering. Particularly in the Astrophysics field, these methods allow, among many other applications, to build a population of stars taking into account all the characteristics of its history, evolution and possible observational biases. This *artificial* population, oftenly call simulated or synthetic population, eventually can be compared with the *real* observed data. The result of this comparison, with the use of proper statistical techniques, allows us to derive a wealth of information about the population under study.

The GAA (Grup d'Astronomia i Astrofísica) at UPC in Castelldefels, Spain, has been pioneer in using the Monte Carlo techniques in the study of the WD population. The code, initially designed for the simulation of the single Galactic WD population (Torres, S., 2002), has later on incorporated a full treatment of binary systems (Camacho, J., 2014; Cojocaru E.R., 2016). A constant update of the cooling physics of WDs, through the use of the most innovative cooling sequences provided by the La Plata group has allow to apply the Monte Carlo code to a wide range of situations. For instance, the study of the thin and thick disk population of our Galaxy, the Galactic halo, the bulge or the population in open and globular clusters (e.g. García-Berro et al., 1999, 2004, 2010; Torres et al., 2001, 2002, 2015, 2016, 2018, 2020; Camacho et al., 2007, 2014; Cojocaru et al., 2017; Canals et al., 2018). Of relevant interest in recent years has been the studies devoted to the *Gaia* mission. As a few illustrative examples, we can mention the study of the capabilities of *Gaia* mission (Torres et al. 2005), or the recent analysis of the Gaia-DR2 100 pc white dwarf population (Jiménez-Esteban et al. 2018). The GAA Monte Carlo simulator has thus become a versatile and robust code, which is considered an international reference model in the simulation of the population of WDs.

2.2 General structure of the Monte Carlo code

The GAA population synthesis code used in this work can be divided in two sub-codes: one for the simulation of the single WD population and another one for the simulation of binary systems. This latter was initially based on the stellar evolution code (BSE) developed by Hurley et al. (2002) but incorporated several updates provided by Camacho, J. (2014), Cojocaru E.R. (2016) and Canals et al. (2018).

A general flowchart of the structure of the Monte Carlo code is shown in Figure 2.1. The code is initialized by a set of input parameters: the initial mass function (IMF), the star formation history (SFH), and the age-metallicity relation (AMR). These parameters, or strictly speaking distribution functions characterize each Galactic population. In recall that in this work we use three-component

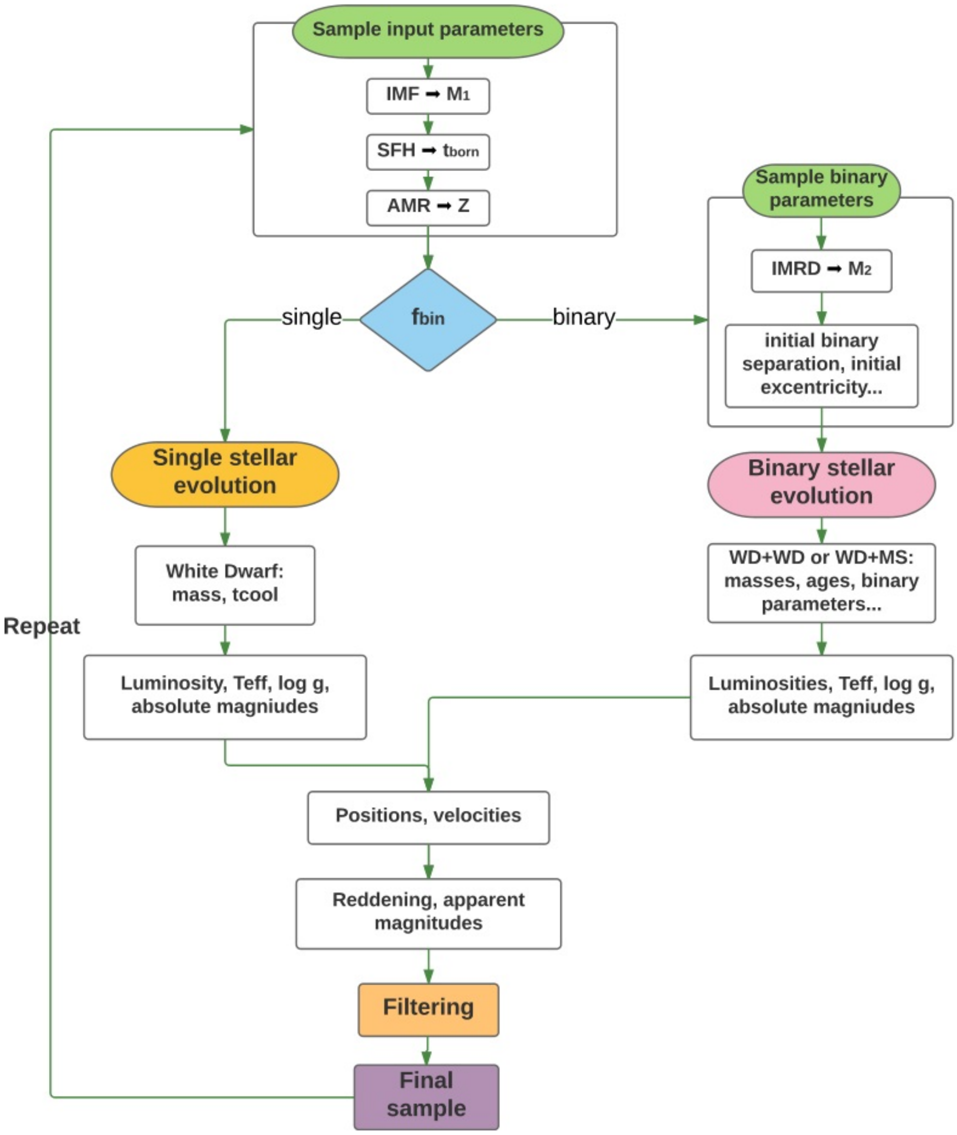

Sample input parameters
IMF ➡ M1
SFH ➡ tborn
AMR ➡ Z
fbin
single
binary
Sample binary parameters
IMRD ➡ M2
initial binary separation, initial excentricity...
Single stellar evolution
Binary stellar evolution
White Dwarf: mass, tcool
WD+WD or WD+MS: masses, ages, binary parameters...
Luminosity, Teff, log g, absolute magniudes
Luminosities, Teff, log g, absolute magniudes
Repeat
Positions, velocities
Reddening, apparent magnitudes
Filtering
Final sample

Figure 2.1: Simplified flowchart representation of the Monte Carlo population synthesis code. The acronyms used in the diagram stands for: IMF initial mass function, SFH star formation history, AMR age-metallicity relation, $f_{\rm bin}$ binary fraction, IMRD initial mass ratio distribution, WD white dwarf, MS main sequence, M_1 progenitor/initial primary mass, M_2 initial secondary mass, $t_{\rm born}$ age at which the star is formed, Z progenitor metallicity, $t_{\rm cool}$ white dwarf cooling time, $T_{\rm eff}$ effective temperature, and g surface gravity. From: Cojocaru E.R. (2016).

Galactic model that includes the thin and thick disk and the halo spheroid.

2.2.1 Sample input parameters

In the case of the thin disk population these are a disk age of 9.2 Gyr with a constant star formation rate. For the thick disk, stars are modelled according to a star formation rate peaked at 10 Gyr in the past. In the case of the halo population, the born time is chosen randomly according to a burst of constant star formation lasting 1 Gyr that happened 13.5 Gyr in the past. The age-metallicity relation for the different Galactic components is chosen to be a constant model with: a solar value of $Z = 0.02$ for the thin disk objects, a subsolar value of $Z = 0.01$ for thick disk stars, and even a lower value of $Z = 0.00173$ for the halo members. Finally, and regardless of the Galactic component, the initial masses of the MS stars are randomly chosen according to an IMF defined by Kroupa et al. (1993) which, for the range of masses (0.5-10 $\mathrm{M}_{\odot}$) that typically evolve into WDs, is equivalent to the Salpeter IMF (Salpeter, 1955) with the standard slope set to $\alpha = -2.35$.

Once this set of input parameters is specified, the code follows, according to the random sampling of the binary fraction parameter, two main branches: the single or the binary evolution. In what follows we briefly describe the main ingredients of each evolutionary path.

2.2.2 Single stellar evolution

If the single stellar evolution is adopted, the subsequent steps are quite straightforward. Once we know the initial mass and metallicity of the star we can derive its main-sequence life-time, $T_{\rm MS}$ (from BaSTI models[1]). Then, by computing $t_{\rm cool}$ (the WD cooling time) as

$$t_{\rm cool} = T_{\rm age} - T_{\rm MS} - t_{\rm born}, \tag{2.2.1}$$

where $T_{\rm age}$ is the age of the Galactic component, and $t_{\rm born}$ the born age, we know if the stars have had enough time to become WDs, that is, if $t_{\rm cool} > 0$. In that case, we adopt then the WD evolutionary sequences of La Plata group (Althaus et al. 2015; Camisassa et al. 2016, 2017; and references therein) to evolve the WD until present time and according to the specific metallicity and core composition (CO or ONe core for single WDs). Besides, the spectral type of the WD, being that DA or non-DA, is randomly chosen according to the known average percentages of 80% and 20% respectively.

With respect to the spatial distribution of stars, both thin and thick disk stars followed a double exponential spatial distribution, however, for the thin disk the scale height was set to 250 pc and the scale length to 2.6 kpc, while for the thick disk these take considerably higher values, 1.5 kpc and 3.5 kpc respectively. Halo stars, on the other hand, are distributed according to an isothermal sphere density. Velocity distributions can be also introduced at this point, however, we can skip this step provided that no specific analysis of the space velocities is going to be done in the present project.

Magnitudes and colors are then calculated on the specific photometric system under study; in our case, in the *Gaia* system. Given that this process is one of the goals of the present project, a detailed description of the procedure can be found in Chapter 3.

2.2.3 Binary stellar evolution

Contrary to the single stellar evolution case, binary evolution involves a more complex treatment given that physical interactions between objects may occur. As a result, an extra set of initial parameters need to be defined. If that is the case, once the mass of the primary MS is chosen according to the IMF of Kroupa et al. (1993), we also need to determine the mass of its companion (the secondary). That is perform according to an initial mass ratio distribution (IMRD) $n(q)$, with $q = M_2/M_1$ being the mass ratio, considering M_1 to be the primary and M_2 the secondary masses. In the current work we will take into consideration three different ways to define the IMRD: a flat distribution ($n(q) = 1$), a distribution proportional to the mass ratio ($n(q) \propto q$), and a distribution inversely proportional to the mass ratio ($n(q) \propto q^{-1}$). Initially, we will make use of the flat distribution, however in Chapter 5 we will discuss the properties of the other models, together with different initial binary fractions which initially takes a value of 50%.

The time at which the stars are born, the total possible ages for the different Galactic components, the spatial distributions and the evolution of the formed WDs in these binaries are the same as described in the previous section for single stars.

The initial separation between the components of the system is chosen accordingly to a logarithmic flat distribution $f(a) \propto ln(a)$, $3 \leq a/R_{\odot} \leq 10^6$ (Davis et al. (2010) and references

[1] http://basti.oa-teramo.inaf.it/

therein), where a stands for the initial semi-major axis of the binary system orbit. In turn, the initial eccentricities of the orbits, e, are randomly chosen according to a thermal distribution $g(e) = 2e, \quad 0.0 \leq e \leq 0.9$ (Heggie , 1975).

As explained in Chapter 1, binary systems can evolve with or without mass transfer episodes, that can be stable or not. An unstable mass transfer episode implies the system evolves through a CE phase. In the code, such stage is modeled according to the α-formalism presented in Tout et al. (1997), which is summarized below.

The CE efficiency parameter $\alpha_{\rm CE}$ describes how efficiently the release of orbital energy of the system is transformed into kinetic energy to eject the envelope. Therefore, one has $E_{\rm bind} = \alpha_{\rm CE}\Delta E_{\rm orb}$, where $\Delta E_{\rm orb}$ stands for the variation in the orbital energy and $E_{\rm bind}$ is the binding energy of the envelope of the primary star. This last parameter is calculated as follows:

$$E_{\rm bind} = -\frac{GM_{\rm donor}M_{\rm env}}{\lambda R_1}. \tag{2.2.2}$$

where $\mathrm{M_{donor}}$ is the mass of the primary star, $\mathrm{M_{env}}$ is the mass of the envelope and λ is the binding energy parameter that symbolizes the ratio between the approximate and exact value of the binding energy. It depends on several variables, such as the mass of the donor or the internal energy, $\alpha_{\rm int}$, that accounts for the fraction of internal energy that is used to expel the CE (including thermal, recombination and radiation energy; Han et al. 1995). Alternatively, the binding energy can be computed according to

$$E_{\rm bind} = \int_{M_{\rm core}}^{M_{\rm donor}} \left(-\frac{GM(r)}{r} + \alpha_{int}U_{int} \right) {\rm d}m \tag{2.2.3}$$

Finally, the code incorporates a set of prescriptions that account for the rest of the aspects of binary evolution such as: tidal evolution, stellar winds, gravitational radiation and magnetic braking, and angular momentum losses.

Chapter 3

The *Gaia* sample: callibrating the population synthesis model

In this chapter we present the observed *Gaia* sample and compare it with the outcome of our population synthesis model. Some initial corrections and effects need to be included in the simulated sample in order to mimic the observed one. For example, the astrometric and fotometric errors need to be implemented and calibrated.

3.1 The observed *Gaia* sample

The ESA mission *Gaia* has provided an unprecedented wealth of information of our Galaxy. In particular, the number of observed (high-confidence) WDs has dramatically increased to $\simeq 350\,000$. However, given that WDs are faint objects, they are difficult (or impossible) to observe at distances farther away than a few kpc. Moreover, observational biases dominate at such large distances. In order to avoid these problems we focus on a closer sample, 100 pc from the Sun. This is expected to be the largest and most complete volume-limited sample available from *Gaia* observations and, at the same time, it contains a reasonable large number of objects, for instance $\simeq 13\,000$ white dwarfs (Jiménez-Esteban et al. 2018) and $\simeq 300\,000$ M9-stars (Gaia Collaboration, 2021).

In Figure 3.1 we show the observational sample within 100 pc from the Sun obtained from *Gaia* on its early data release 3 (EDR3), the latest to date. It can be clearly identified the typical region of MS stars at the upper right side of the HR diagram with evolving red giants at the top of that track and WD stars at the lower left part. Additionally, a certain number of objects can be observed closer to the WD and MS tracks and, in particular, in between them. These last objects are of especial interest in our study because they probably represent WDMS binary systems.

As previously stated, the 100 pc sample can be assumed to be complete. Consequently we will use the number of objects found in the WD region as a normalization criteria for our simulations. Following Torres et al. (2019), this region is defined by $M_G > 2.95 \times (G_{\rm BP} - G_{\rm RP}) + 10.83$ for $(G_{\rm BP} - G_{\rm RP}) < 1.2$ and $M_G > 1.87 \times (G_{\rm BP} - G_{\rm RP}) + 12.16$ for $(G_{\rm BP} - G_{\rm RP}) > 1.2$ and up to $(G_{\rm BP} - G_{\rm RP}) = 2.0$. These cuts are illustrated as dashed lines in Figure 3.1. Additionally, we apply the following restrictions in order to select objects with good photometric and astrometric values:

i) $\varpi > 10\,$mas and $\varpi/\sigma_\varpi > 10$: to select objects below 100 pc with parallax errors under 10%.

ii) $F_{\rm BP}/\sigma_{F_{\rm BP}} > 10$, $F_{\rm RP}/\sigma_{F_{\rm RP}} > 10$ and $F_{\rm G}/\sigma_{F_{\rm G}} > 10$: to select objects with flux relative errors under 10%.

iii) RUWE > 2 or (ASTROMETRIC_EXCESS_NOISE$>$ 2 and ASTROMETRIC_EXCESS_NOISE_SIG$>$

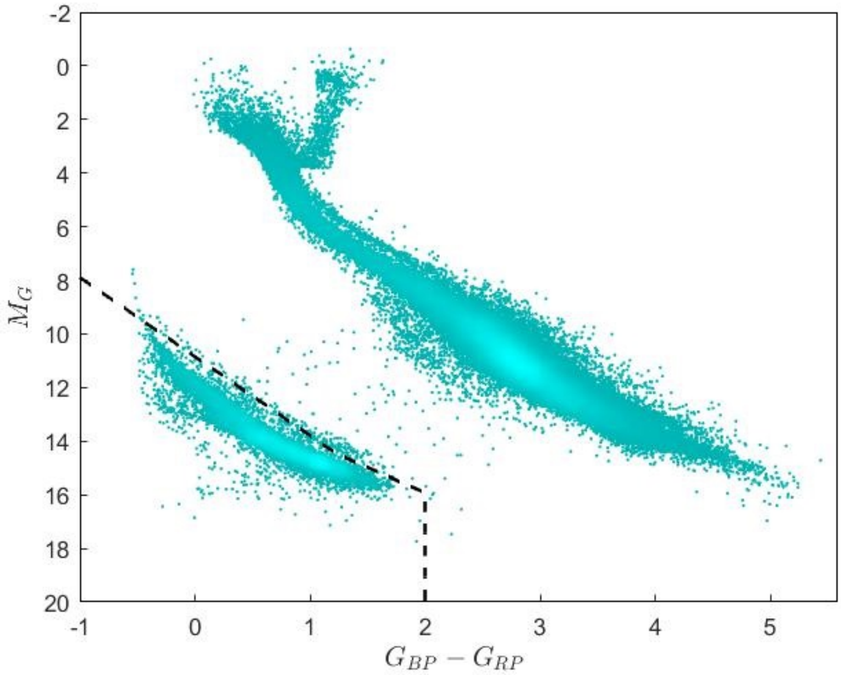


Figure 3.1: *Gaia* HR diagram of the observed 100 pc sample. It can be clearly identified the MS region at the upper and right side of the diagram, and the WD region on the lower left part.

2): to avoid contamination from nearby sources or bright background stars.

The final number of objects thus found in the WD region is 15753.

3.2 The population synthesis model

The output of the population synthesis model characterizes each star by some of its physical parameters such as the mass, the effective temperature ($\mathrm{T_{eff}}$) or the luminosity. In order to compare with the observational sample it is required to obtain the corresponding *Gaia* magnitudes (G, G_{BP} and G_{RP}). The first step we made to derive the magnitudes was to interpolate $\mathrm{T_{eff}}$ and the decimal logarithm of the surface gravity ($log_{10}(g) \equiv \log(g)$) from MIST models [1] (Dotter, A., 2016; Choi et al., 2016) . This was carried out by performing a bilinear interpolation of the input variables. In the case of MS stars the tables provide the bolometric correction for each magnitude. To obtain the magnitude it is required to perform the following calculation:

$$M_{\mathrm{G}} = 4.75 - 2.5 \cdot \log\left(\frac{L}{L_{\odot}}\right) - \mathrm{BC_G}, \tag{3.2.1}$$

where M_{G} is the absolute *Gaia* magnitude, L the luminosity of the star in solar units, $L_{\odot}$ the luminosity of the sun and $\mathrm{BC_G}$ the bolometric correction for the magnitude M_{G}. Equation 3.2.1 is also valid for the rest of *Gaia* magnitudes, i.e., G_{BP} and G_{RP}, but using the corresponding bolometric corrections. For WD stars, the MIST model table provides the magnitudes for the EDR3 passbands, $\mathrm{G_{EDR3}}$, $\mathrm{BP_{EDR3}}$ and $\mathrm{RP_{EDR3}}$, calculated at a fixed distance of 100 pc. To obtain the absolute magnitudes the following calculation is needed:

[1] `http://waps.cfa.harvard.edu/MIST/`

$$M_{\rm G} = G_{\rm EDR3} - 2.5 \cdot \log\left(\frac{R^2}{10^2}\right). \tag{3.2.2}$$

where R is the radius of the star in parsecs. The value of the radius is not provided by the simulation, but it can be calculated from the Stefan-Boltzman equation: $L = 4\pi\sigma R^2 T^4$, being L the luminosity, $\sigma = 5.670374 \cdot 10^{-8}\ m^{-2}K^{-2}$ the Stefan-Boltzman constant, T the effective temperature, and R the radius.

The value of $\rm T_{eff}$ is given directly by the simulation, however $\log(g)$ is not and therefore it has to be calculated as the logarithm of $g = {\rm G}M/R^2$, where G stands for the gravitational constant, M the mass of the star and R its radius expressed all of them in cgs units. This last value, as previously stated, is not provided by the simulation and it is calculated through the Stefan-Boltzmann equation.

In the case of the MS the metallicity of the star plays an in important role (for instance, in its life-time) and depending on its value the interpolation table should be adequately selected. As a consequence we preformed a double bilinear interpolation. On the other hand, metallicty plays only a minor role in the WD evolution, and this is the reason why we applied a simple bilinear interpolation for this case.

The previous procedure permits deriving the synthetic *Gaia* magnitudes for single stars. For WDMS and DWD binaries, once we know the magnitudes for each component, we determine whether the systems would be detected as two separated objects (resolved) or as a single one (unresolved). For that, we made use of the angular separation, s. The threshold chosen to distinguish those two cases is a limiting value of $s_{\rm lim} = 2\,$arcsec, meaning that, for $s > s_{\rm lim}$ we will consider the systems as resolved (in these cases the magnitudes are those directly obtained for each component in the same way as for single stars), otherwise we will regard them as unresolved. Although the *Gaia* mission is able to resolve systems with smaller angular separations (theoretically the limiting value is around $\sim 0.5\,$arcsec), this limits increases to 2 arcsec for samples selected with good astrometric and photometric parameters, as it is our case. Hence, for the unresolved systems, we obtain the combined *Gaia* magnitudes of the systems as the sum of fluxes of the individual stars according to:

$$m_f = -2.5 \cdot \log\left(10^{-0.4\cdot m_1} + 10^{-0.4\cdot m_2}\right), \tag{3.2.3}$$

where m_f is the final magnitude, (any of the three used in the *Gaia* system), and m_1 and m_2 the individual magnitudes of each star.

In Figure 3.2 we show the *Gaia* HR diagram containing all the different population of stars and binaries that we simulated: the upper right region contains the MS, the lower left one encloses the WD, while the area in between would be the unresolved WD+MS systems area.

3.3 Calibrating the synthetic sample

3.3.1 Adjusting the MS minimum mass

A first comparison between the observed sample and the simulated one reveals that the synthetic population of MS reached values of $G_{\rm BP} - G_{\rm RP}$ up to 6.5. Nonetheless, the maximum value of the observed MS is $\rm BP - RP \sim 5$. This discrepancy is a consequence of the minimum mass used in the simulated models. Theoretically the minimum mass required for a star to fuse hydrogen in its core is $\approx 0.08\rm M_\odot$ (see Kroupa et al. 1993 and references therein). However, the lowest mass of a certain sample of stars could be higher than this minimum mass, given that it depends on how these stars have locally been formed. In our case, we are considering the solar neighborhood up to 100 pc, and the analysis of the *Gaia* sample allows us to establish this limit. In order to do this, we represent

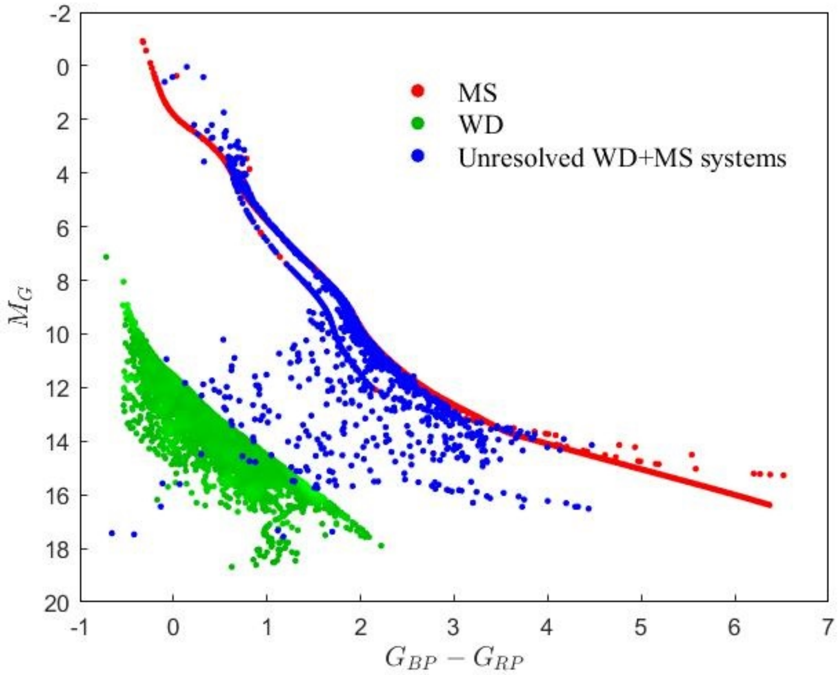


Figure 3.2: *Gaia* HR diagram of the different simulated populations when no photometric errors are taken into account: MS stars (including single MS stars and resolved MS from WDMS systems; red dots), WD stars (including single WDs, WDs that evolve from or as a result of mergers, resolved WDs from WDMS systems and resolved and unresolved DWD systems; green dots) and unresolved WDMS systems (blue dots).

in Figure 3.3 a plot of the mass of the single MS stars as a function of the color $G_{\rm BP} - G_{\rm RP}$. The plot reveals that a threshold for the mass of $0.095\,{\rm M}_{\odot}$ guarantees that the value of $G_{BP} - G_{RP}$ does not exceed 5. Such value for the minimum mass is a conservative guess, given that we need to take into account that photometric errors will spread the final color distribution. Further checks have been made using the distribution of $G_{\rm BP} - G_{\rm RP}$ to make sure that the limits of the simulated sample and the observed one indeed coincide, and therefore the simulated populations of MS stars are located in the same region as the observed ones. In figure 3.2 it can be observed that the simulated sample extends to the upper limit of $G_{\rm BP} - G_{\rm RP}$ that the observed population does not. Once the mass cut has been made (Figure 3.4), the upper edges of both populations coincide.

3.3.2 Adjusting the MS track

The initial comparison between the simulated and the observed samples reveals a displacement of the MS populations (thin and thick disk and halo), that is, the simulated ones are are located slightly lower than the observed ones in the HR diagram. This can be observed in both panels of Figure 3.4, where the MS tracks are visibly lower than the bulk of the observed population. Even for the case of the thin disk population (the track that is highest of the three) it does not match the peak of density of the observed sample and it is located below it. After several checks we could not find the source of the discrepancy, being a possible explanation the flux calibration criteria employed in the MIST tables. In consequence, we decided to perform an adjustment by hand as an *ad hoc* solution, waiting for future studies to provide a clearer explanation of this discrepancy. The adjustment is made looking into the $M_{\rm G}$ vs. $G_{\rm BP} - G_{\rm RP}$ and $M_{\rm G}$ vs. $G - G_{\rm RP}$ density maps that result from the simulated sample and comparing them with the observed one. We find that the following shifts should be applied to the MIST tables: $M_{\rm G}$+0.02, $M_{\rm BP}$+0.2 and $M_{\rm RP}$+0.05. It is worth saying that this artificial shift is specially important in the BP passband, while the other passbands are only marginally affected.

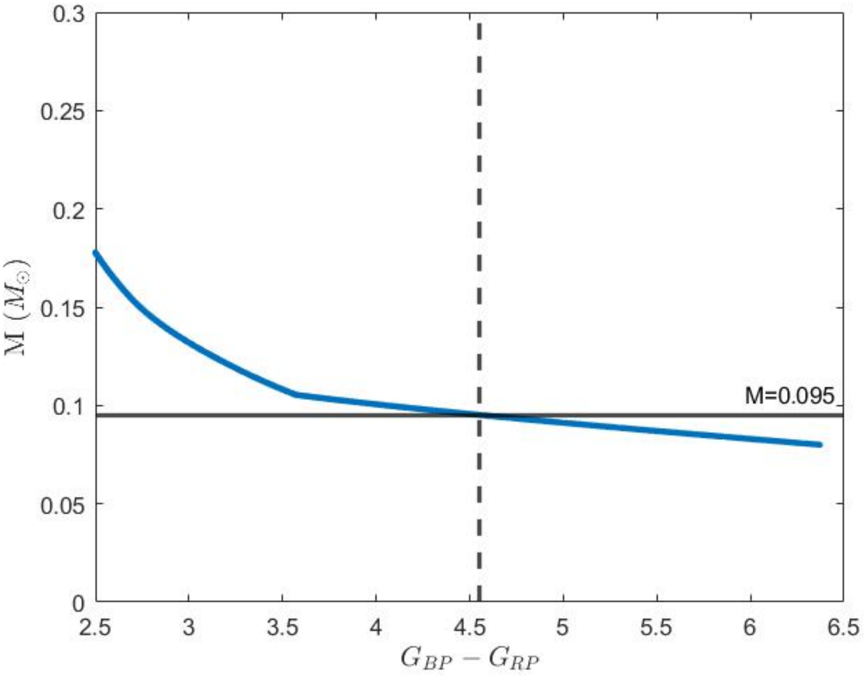


Figure 3.3: In blue the raw MS singles stars track; the black horizontal line represents the chosen threshold of $0.095\mathrm{M}_\odot$.

An alternative method to calculate the *Gaia* magnitudes for MS stars consists in applying the evolutionary models from PARSEC [2] (Bressan et al. 2012). The parameters used in this case were: the version 1.2S of the PARSEC tracks, without COLIBRI tracks, meaning no limit in Z, and the values for n_{inTPC} and η_{Reimers} as default ($n_{inTPC} = 10$, $\eta_{\mathrm{Reimers}} = 0.2$); the photometric system is the Gaia EDR3 with the OBC bolometric corrections; no circumstellar dust; null interstellar extinction ($A_v = 0.0$); long period variability; Kroupa initial mass function; the metal fraction Z according to the ones we need (see table 3.1); finally the output being the isochrone tables. In figure 3.5 we show (red lines) the MS track from top to bottom corresponding, respectively, to the three Galactic populations thin and thick disk and halo. The reference one, which is the thin disk track, perfectly fits the densest region of the observed sample. Therefore we can conclude that both populations, simulated and observed, coincide in their general shape (for a further checking see figure 3.6, where it is displayed a density map once dispersion in metallicity is added). It is worth saying that these new models give directly the absolutes magnitudes in *Gaia* filters, avoiding the need to calculate the bolometric correction as in the MIST models. This way we avoid the flux calibration problem and we do not need to add an artificial shift. Although the final population obtained through both models, PARSEC and MIST, is practically equivalent, the PARSEC models provide more satisfactory results since there is no need of an extra calibration. Henceforth, in what follows, we consider the PARSEC model for calculating the MS population tracks.

3.3.3 The effects of the metallicity dispersion

As previously stated, synthetic single MS stars follow straight and well defined tracks in the HR diagram (see figure 3.5). The three distinct tracks correspond to the three different metallicities adopted for each of the three Galactic components (see Table 3.1). However, it is well known that even in a certain population, for instance, the thin disk, there is a dispersion on the metallicity values at a given age (see e.g. Cassagrande et al. 2011). This dispersion in metallicity translates into a dispersion of points around the MS tracks, as reflected in the observed sample.

[2] `http://stev.oapd.inaf.it/cgi-bin/cmd`

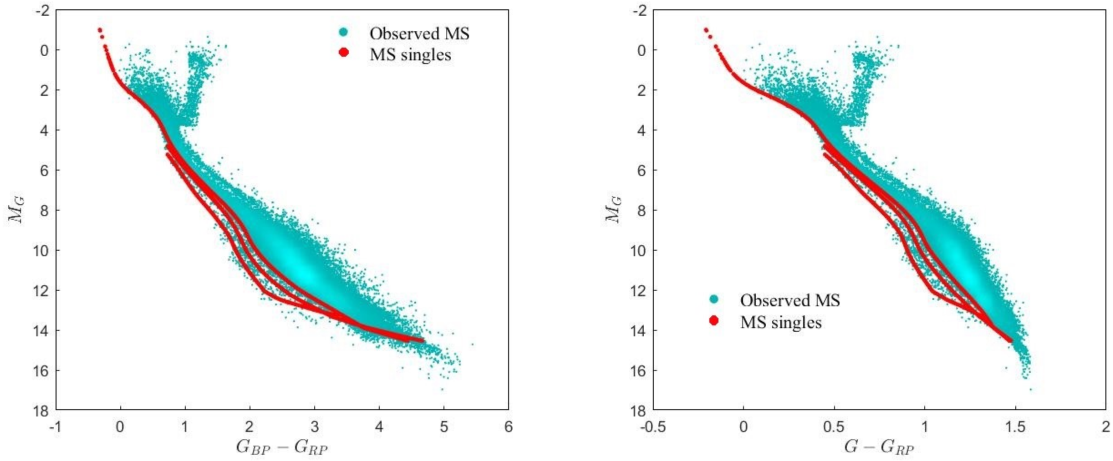


Figure 3.4: HR diagrams in *Gaia* magnitudes of M_G vs $G_{\rm BP} - G_{\rm RP}$ (left) and $M_{\rm G}$ vs $G - G_{\rm RP}$ of the observed MS star sample (in cyan) and the raw MS single simulated tracks (in red).

	Z	[Fe/H]	σ
Thin disk	0.02	0.073	0.005
Thick disk	0.01	-0.234	0.003
Halo	0.00173	-1.00	0.001

Table 3.1: Summary of the metallicity and the standard deviation used for each one of the three galactic components.

In order to mimic the observed sample, in which MS stars do not follow a single point track but a broaden area, we add a dispersion into the value of the metallicity. For such dispersion we adopted a Gaussian-like distribution with mean value zero (μ=0) and a different standard deviation for each Galactic component (see Table 3.1).

In Figure 3.6 it can be observed the result of adding the metalicity dispersion to the MS stars. MS tracks for the different Galactic populations increase their width, which represents a better resemblance with the observed sample. The chosen values of the standard deviation make sure of such resemblance; for lower values the simulated sample remains very near the tracks, while for higher values of sigma the simulated population spreads too much around the tracks. At the same time we have to consider that the addition of photometric and astrometric error (see the next section) will most likely introduce an extra broadening in the sample, therefore higher values of the standard deviation are not required.

As mentioned before, the effect of metallicity in the WD tracks is nearly negligible. Therefore, there is no need for introducing a dispersion due to metallicity in these tracks.

3.4 The effects of the photometric and astrometric errors

So far our synthetic stars do not contemplate any source of error. However, when comparing with observed data, physical and instrumental limitations introduce some physical errors in the measured values. These are specific of the process of observation and depend on the particularities of the instruments used. In our case, the astrometric and photometric errors have been introduced into the simulated populations according to the prescriptions described in the *Gaia* performance[3]. In both cases, we treat the errors as a Gaussian noise, in the same way as section 3.3.3 with $\mu = 0$ but adopting a standard deviation that is calculated for each star regardless the Galactic component to which it belongs. The procedure is to first compute the standard deviation and add

[3] `https://www.cosmos.esa.int/web/gaia/science-performance`

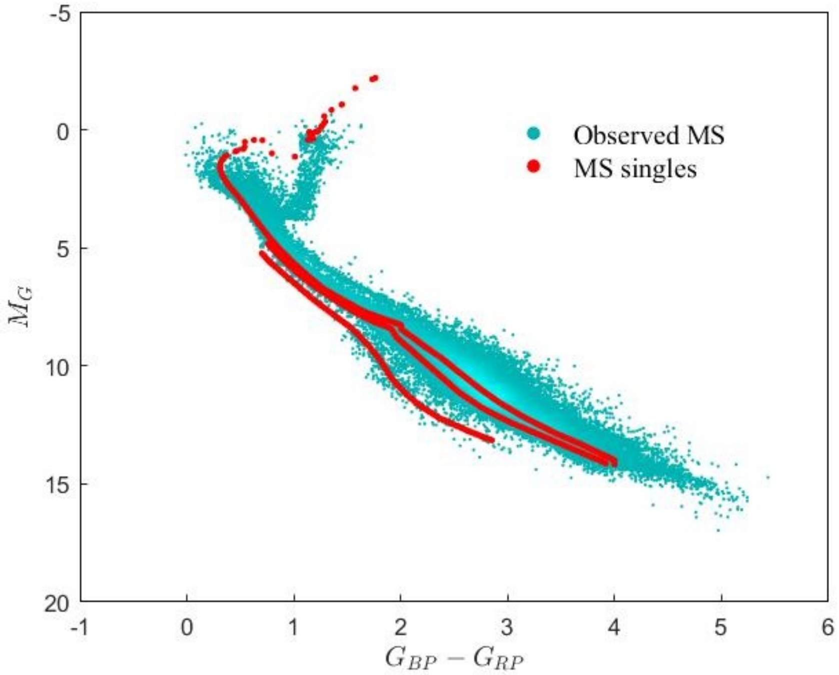


Figure 3.5: In red the raw single MS star tracks, in cyan the observed population of MS stars.

the corresponding astrometric error to the parallax. Next we compute the standard deviation and add the error to the three magnitudes (G, $G_{\rm BP}$ and $G_{\rm RP}$) to recalculate the apparent magnitudes. Finally we obtain the absolute magnitudes using the new parallax and apparent magnitude values.

The parallax standard deviation error is computed according to the *Gaia* performance as follows:

$$\sigma_{\varpi} = T_{\rm factor} \cdot (40 + 800 \cdot z + 30 \cdot z^2)^{0.5}, \tag{3.4.1}$$

where the standard deviation is given in μas, $T_{\rm factor}$ is a scale factor and z is an auxiliary variable defined as $z \equiv \max[10^{0.4\cdot(13-15)}, 10^{0.4\cdot(G-15)}]$. The choice of the scale factor $T_{\rm factor}$ is made using the criteria that the simulated sample has to look like the observational one, that is, to have a similar dispersion of points. For high values of $T_{\rm factor}$, the WD population spreads too much entering into the unresolved WD+MS area, and even reaching the MS tracks. In contrast, for values around unit, such $T_{\rm factor}$ provides a satisfactory configuration of the resulting populations that perfectly resembles the observational data. As the general shape does not vary significantly for vales of $T_{\rm factor}$ close to one, it has been chosen $T_{\rm factor} = 1$ as the most appropriate value.

In a similar way the standard deviation of the apparent magnitude G is computed as:

$$\sigma_G = 1.2 \cdot 10^{-3} \cdot (0.0001985 + 1.8633 \cdot z + 0.04895 \cdot z^2)^{0.5}, \tag{3.4.2}$$

where, on this occasion, the auxiliary variable z is $z \equiv \max[10^{0.4\cdot(12-15)}, 10^{0.4\cdot(G-15)}]$. Similarly, the standard deviation of magnitudes $G_{\rm BP}$ and $G_{\rm RP}$ are calculated as:

$$\sigma_{BP/RP} = 10^{-3} \cdot (10^{c_{BP/RP}} + 10^{b_{BP/RP}} \cdot z + 10^{a_{BP/RP}} \cdot z^2)^{0.5}. \tag{3.4.3}$$

but now with a value of z of $z \equiv \max[10^{0.4\cdot(11-15)}, 10^{0.4\cdot(G-15)}]$.

The prescription used by *Gaia* performance when computing the auxiliary variables $a_{\rm BP/RP}$, $b_{\rm BP/RP}$ and $c_{\rm BP/RP}$ in equation 3.4.3, are a function of the Johnson's color $(V - I_c)$. Our population

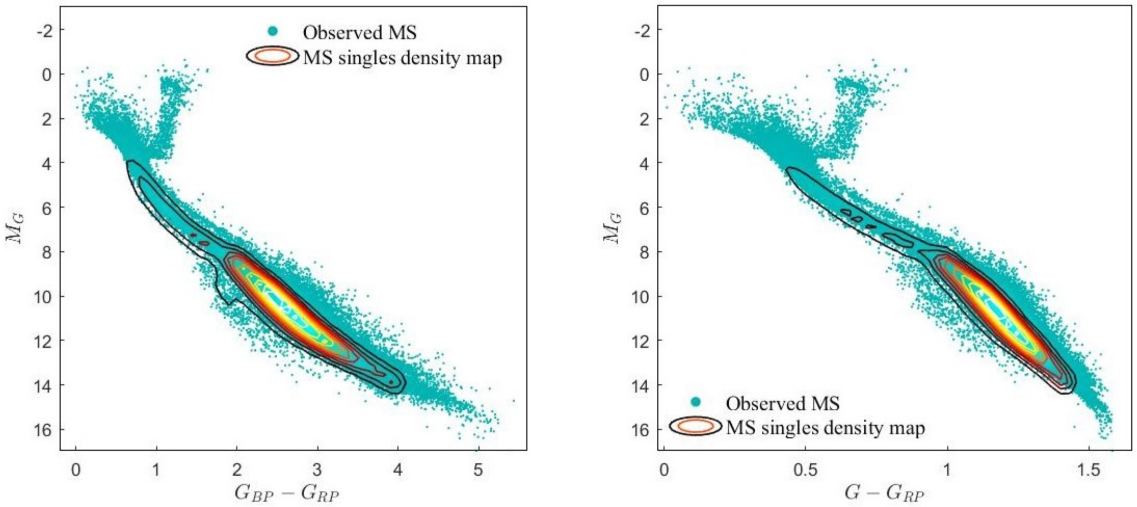


Figure 3.6: HR diagrams in *Gaia* magnitudes of M_G vs $G_{BP} - G_{RP}$ (left) and M_G vs $G - G_{RP}$ of the density map of the simulated MS singles population superposed to the observational sample.

synthesis model does not provide this, however we can made adopt the reasonable approximation that $V - I_c \sim G_{\mathrm{BP}} - G_{\mathrm{RP}}$ (table C.2. of Riello et al. 2021).

$$\begin{cases} a_{BP} = -0.000562(G_{\mathrm{BP}} - G_{\mathrm{RP}})^3 + 0.044390(G_{\mathrm{BP}} - G_{\mathrm{RP}})^2 + 0.355123(G_{\mathrm{BP}} - G_{\mathrm{RP}}) + 1.043270 \\ b_{BP} = -0.000400(G_{\mathrm{BP}} - G_{\mathrm{RP}})^3 + 0.018878(G_{\mathrm{BP}} - G_{\mathrm{RP}})^2 + 0.195768(G_{\mathrm{BP}} - G_{\mathrm{RP}}) + 1.465592 \\ c_{BP} = +0.000262(G_{\mathrm{BP}} - G_{\mathrm{RP}})^3 + 0.060769(G_{\mathrm{BP}} - G_{\mathrm{RP}})^2 - 0.205807(G_{\mathrm{BP}} - G_{\mathrm{RP}}) - 1.866968 \\ a_{RP} = -0.007597(G_{\mathrm{BP}} - G_{\mathrm{RP}})^3 + 0.114126(G_{\mathrm{BP}} - G_{\mathrm{RP}})^2 - 0.636628(G_{\mathrm{BP}} - G_{\mathrm{RP}}) + 1.615927 \\ b_{RP} = -0.003803(G_{\mathrm{BP}} - G_{\mathrm{RP}})^3 + 0.057112(G_{\mathrm{BP}} - G_{\mathrm{RP}})^2 - 0.318499(G_{\mathrm{BP}} - G_{\mathrm{RP}}) + 1.783906 \\ c_{RP} = -0.001923(G_{\mathrm{BP}} - G_{\mathrm{RP}})^3 + 0.027352(G_{\mathrm{BP}} - G_{\mathrm{RP}})^2 - 0.091569(G_{\mathrm{BP}} - G_{\mathrm{RP}}) - 3.042268 \end{cases} \tag{3.4.4}$$

Once the photometric and astrometric errors are added, we obtain the absolute G magnitudes, M_G, through the standard formula:

$$M_G = G + 5 + 5 \cdot \log(\varpi) \tag{3.4.5}$$

where G is the apparent magnitude and ϖ is the parallax expressed in arcsec. Similarly we obtain the apparent magnitudes for G_{BP} and G_{RP}.

Finally, we introduce a threshold to the value of apparent magnitude G given that observationally it has an upper limit of $G < 20.7$ (Gaia Collaboration 2021). Figure 3.7 shows our final synthetic population after applying the metallicity dispersion and the photometric and astrometric erros.

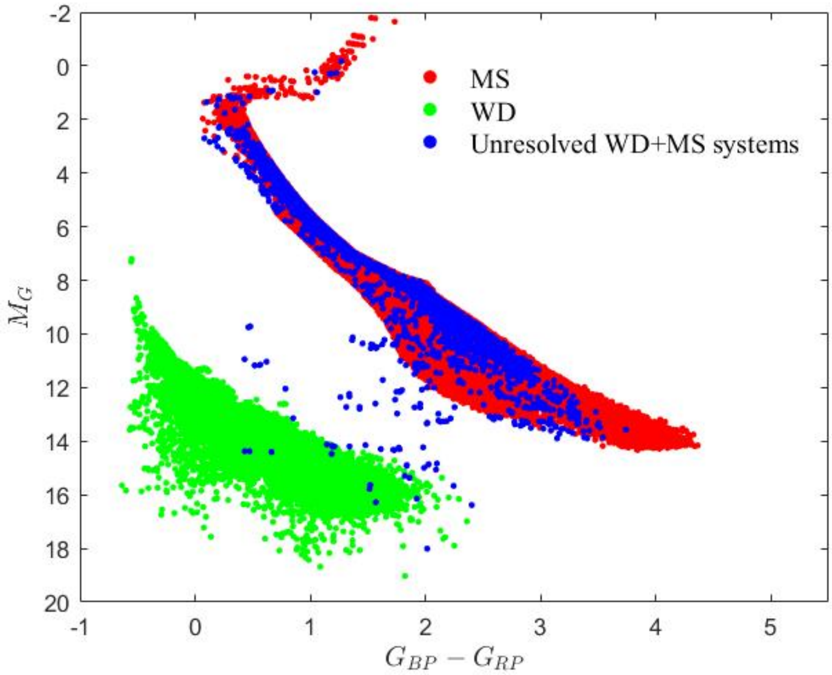


Figure 3.7: *Gaia* HR diagram of the synthetic population once the metallicity dispersion and the photometric and astrometric errors are added. Simulated stars belonging to the MS, WD and unresolved WD+MS systems are colored in red, green and blue, respectively.

Chapter 4

The white dwarf sub-populations

In Chapter 4 we analyzed the different sub-populations of single and binary systems containing at least one WD. It has to be emphasised that an incredible amount of information can be extracted from each simulation and one could study in detail every aspect of it, which is clearly beyond the scope of the present work. Because of this, here we broadly discuss the statistical properties of each sub-population with the aim of showing the potential of the results obtained for future works. To do so we chose a standard reference model for our binary population synthesis simulation. This reference model, which is the same used in Chapter 3, is not necessarily the one that better fits the observed sample, but uses the most commonly adopted assumptions. The discussion on which model best fits the observed samples is later addressed in Chapter 5. The simulated data is compared with currently observed populations extracted from the *Gaia* mission.

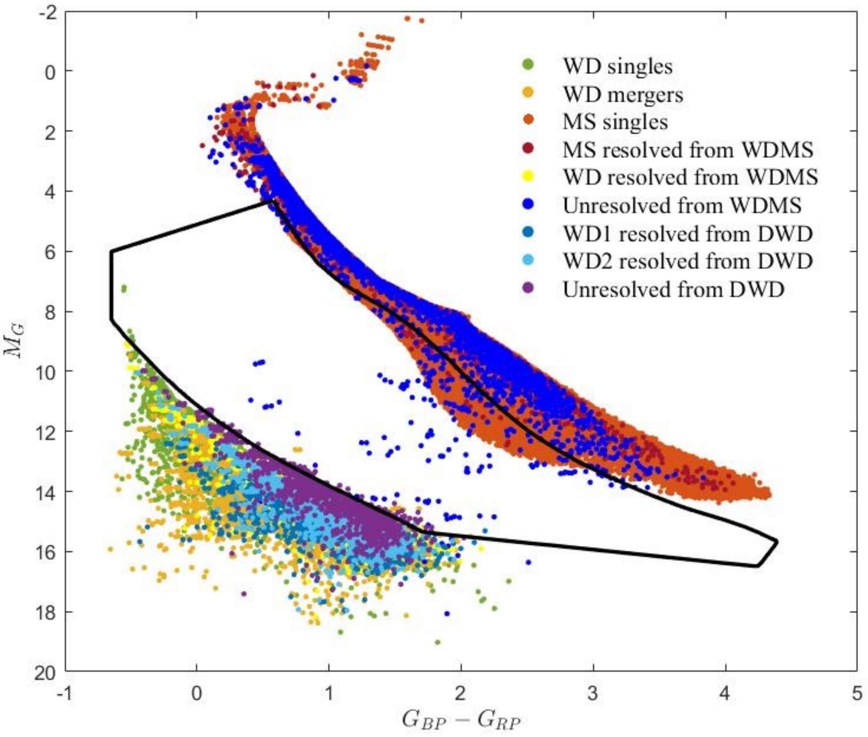


Figure 4.1: HR diagram in *Gaia* magnitudes of all stars populations.

4.1 The HR diagram regions: general counts

First of all, we analyze the location within the HR-diagram of the different sub-populations under study. Those are represented by different colors in Figure 4.1. Also, for a visual reference, the WDMS region defined by Rebassa-Mansergas et al. (in press) is marked as a solid black line. In view of this, three big regions or areas can be defined: the WDMS region itself, and the WD and MS regions, respectively, below and above the former region. The WD region is dominated by single and merger WDs. But it is also possible to find a sizeable fraction of assimilated WDs, that is, resolved and unresolved DWD, and resolved WDMS. The MS region is dominated by MS stars belonging to single or binary systems. Finally, the WDMS region is a bridge region in between the WD and MS regions. As its name indicates, the WDMS area has been defined in order to mainly locate unresolved WDMS systems, although some contamination of single low metallicity MS stars or unresolved DWD can occur.

Table 4.1 includes a summary of the number of objects of each sub-population that fall into the three general regions defined in the HR diagram. The values have been normalized to the value of the total observed objects in the WD area, i.e, 15753 objects. Doing so allows us to make a proper comparison between the observational and simulated populations.

	Total	MS region	WDMS region	WD region
WD singles	7841	0	4	7837
WD mergers	4311	0	13	4298
MS singles	104022	99334	4688	0
MS resolved	1177	1146	31	0
WD resolved	1177	0	0	1177
WDMS unresolved	1507	1361	139	7
WD 1	866	0	0	866
WD 2	866	0	1	865
DWD unresolved	755	0	53	702

Table 4.1: Summary of the number of objects of each sub-population falling on the different three general regions considered, once the normalization is made.

For a better understanding of the distribution of the different sub-populations across the HR diagram, Table 4.2 provides the percentages of the amount of objects of a certain sub-population that is placed within a determined region in relation to the total amount of stars of this sub-population (in blue). In other words, it indicates how a fixed sub-population is distributed along the three different regions. On the other hand, we show in red the proportion that represents a sub-population in relation to the total number of stars that are placed in a given region.

	Total	MS area		WDMS area		WD area	
WD singles	7841	0%	0%	0.08%	0.05%	49.75%	99.95%
WD mergers	4311	0%	0%	0.26%	0.3%	27.28%	99.70%
MS singles	104022	97.54%	95.50%	95.11%	4.50%	0%	0%
MS resolved	1177	1.12%	97.40%	0.63%	2.6%	0%	0%
WD resolved	1177	0%	0%	0%	0%	7.47%	100%
WDMS unresolved	1507	1.34%	90.32%	2.28%	9.22%	0.04%	0.46%
WD 1	866	0%	0%	0%	0%	5.50%	100%
WD 2	866	0%	0%	0.02%	0.12%	5.50%	100%
WDWD unresolved	755	0%	0%	1.08%	7.08%	4.46%	92.98%
Total	122521	101841		4928		15752	

Table 4.2: Summary of the percentages of the different sub-populations according to the three general regions once the normalization is made. In red, the percentage that the sub-population represents with respect to the total number of stars in the region. In blue, the percentage of the sub-population that falls into that region.

As expected, the different WD populations (singles, mergers, resolved components of a WDMS binary, resolved components of a DWD system or unresolved DWD systems) mainly fall into the WD area and in any case they are found within the MS zone. However, occasionally they may be placed into the WD+MS area, always a small fraction, in general lower that 1% except for the case of unresolved DWD systems in which the percentage rises to 7% (this issue will be discussed in section 4.4.2).

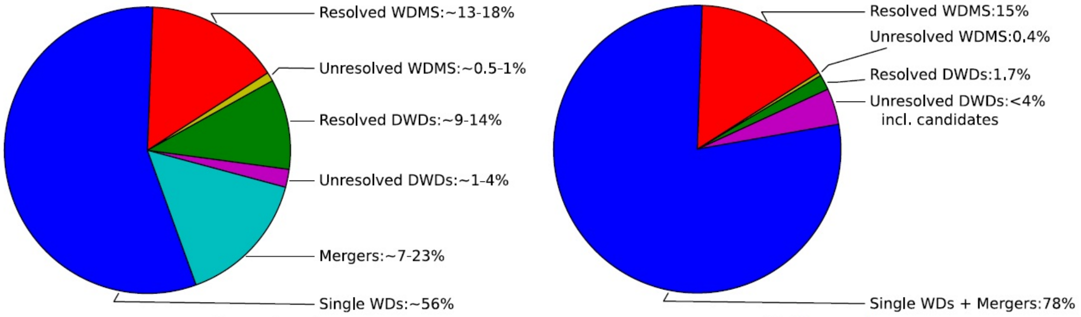
A similar situation occurs with the MS stars (singles and resolved components of WDMS systems); the majority of them are in the MS region with some objects that are also located within the WDMS area (none are found in the WD region, as expected). Interestingly, in the WDMS region the dominant population is that of single MS stars, being more than 95% of the total (see also Figure 4.1). This can be understood by looking at Table 4.3, which provides the number and percentage of single MS stars according to their Galactic membership within the MS and WDMS areas. We see that in the MS region the dominant stars are those which are members of the thin disk, representing almost all thin disk stars in this zone. In this region there are the majority of the thick disk stars as well. However, the halo stars, which are intrinsically less luminous, are generally located below the solar-metallicity MS track and therefore within the WDMS area.

	MS region			WDMS region		
Thin disk	83155	83.71%	99.95%	41	0.88%	0.05%
Thick disk	15300	15.40%	98.00%	311	6.63%	2.00%
Halo	879	0.89%	16.86%	4336	92.49%	83.14%

Table 4.3: Summary of the numbers and percentages of MS stars into the MS and WDMS region according to their Galactic membership, which can be used as a proxy for metallicity. In red the percentage that the stars with that membership represents for the total number of single MS stars in the area (black numbers). In blue the percentage of the stars with that membership that falls into that area.

4.2 Comparison with previous works

In Toonen et al. (2017) they made a similar statistical analysis of the different sub-populations as have been done here. Even though their sample had a high-level of completeness (80-90%), it suffered from a significant paucity in the number of objects ($\sim$ 130) given that the sample was restricted to only 20 pc from the Sun. In Figure 4.2 we plot a pie chat representing the fraction of the different sub-populations found in their observed sample (right panel) and their simulated predictions (left panel).



(b) Predicted

(a) Observed

Figure 4.2: Fractions found by Toonen et al. (2017) of the different sub-populations observed (right panel) and by their simulations (left panel) within the WD area for a 20 pc sample. From: Toonen et al. (2017).

The major discrepancy found by Toonen et al. (2017) concerns the resolved DWD population. 9-14% of this population is expected in the 20 pc sample, however, only 1.7% is observed. Even taking into account statistical errors and several prescriptions for the synthetic population models, the difference persists as an order of magnitude.

Now, returning to our analysis of the *Gaia* 100 pc sample, we show in Figure 4.3 a comparison of the fraction of the different simulated populations (left panel) with those found in the observed sample (right panel) in the WD region. In this particular case, the observational data has been derived from a common-proper motion analysis of the systems located in the WD region (Jiménez-Esteban et al. priv. comm.). In order to follow a similar analysis as done in Toonen et al. (2017) we count in this case the number of systems instead of individual objects. That is, we compute resolved DWD systems as one system, and not by its components, WD1 and WD2; resolved WDMS are considered according to the WD component of these resolved systems. We also need to take into account that observationally speaking we only have the information on the total number of objects within the WD area and the number of resolved DWD and WDMS systems but not, for instance, of the fraction of single WDs and merger WDs (which are in principle indistinguishable). Moreover, we also assumed that unresolved systems represents only a small fraction.

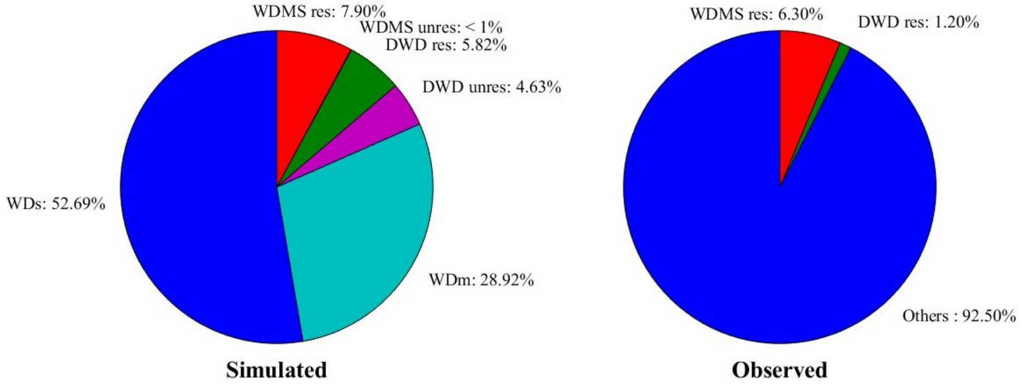


Figure 4.3: Simulated (left panel) and observed (right) fraction of the different sub-populations within the WD area for our 100 pc sample, being *Others* the sum of the WDs, WDm, WDMS unresolved and DWD unresolved populations.

Several discrepancies arise between the fractions of the observed and simulated populations. The first is the percentage of resolved DWD. While in the observed sample it remains as a value of 1.20%, in the simulated one it goes up to 5.82%, which is relatively larger. In the case of the resolved WDMS sub-population the difference is not as large as in the DWD situation, however it is not a perfect fit either. As previously commented, Toonen et al. (2017) found similar discrepancies for the observed and predicted samples for the 20 pc. In what follows we will analyse the different sub-populations one by one and in Chapter 5 we will analyse how the percentages of the predicted populations change for different models.

4.3 The WDMS sub-population

4.3.1 Unresolved WDMS systems

The first sub-population we will be looking at is the unresolved WDMS systems. The left panel of Figure 4.4 shows the location of these systems in the HR diagram (blue solid dots). The right panel shows the location of the two separated components (note that the simulation provide the information of both stars even though they are unresolved). It can be noticed that in the majority of the unresolved WDMS systems the dominant star is the MS, and therefore, such systems are placed in the MS track. Another interesting fact that we can observe from the right panel is a group of MS companions that concentrate at the right lower part, out of the MS track. Those systems, likely, have evolved through mass transfer episodes so that the MS star has lost a great amount of mass thus becoming a brown dwarf.

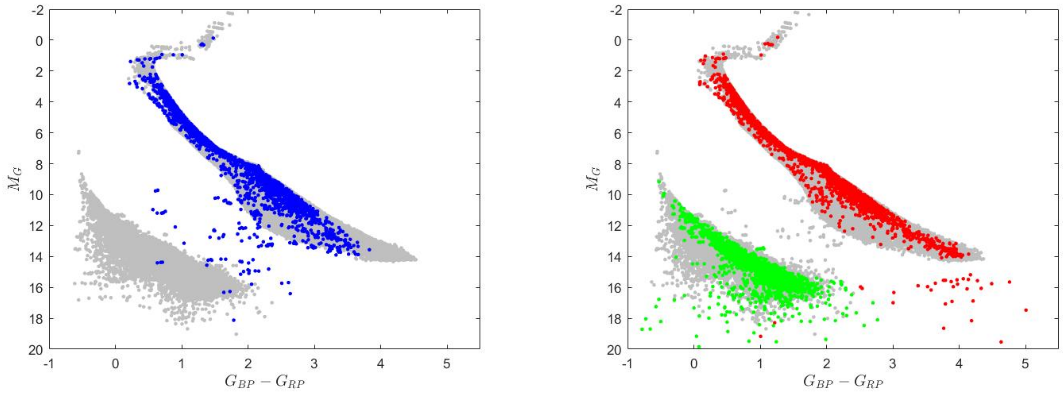


Figure 4.4: HR diagrams in *Gaia* magnitudes. Left panel: in blue the WDMS unresolved systems, in gray the other populations. Right panel: in green the WD component of the unresolved WDMS system, in red the MS component, in gray the other populations.

The left panel of Figure 4.5 shows the distribution of the unresolved WDMS population that are located inside the defined WDMS region (black solid lines), as well as the location of its individual components. While the WDs mostly remain in the WD area, some of the MS component fall within the WDMS region. Nonetheless, in both cases the stars remain close to the edges of their respective areas. The right panel shows the distribution of the components across the HR diagram in units of log($T_{\rm eff}$) vs log($L/L_\odot$) for both the simulated and the observed (Rebassa-Manserga et al. in press) WDMS unresolved populations inside the WDMS region. As it can be seen, the two populations follow a similar tendency.

The observed sample consists in a set of 112 unresolved WDMS binary candidates, of which 98 have reliable estimates of their parameters. According to our simulated population, we expect to find within the WDMS region 139 unresolved systems. This number represents the maximum number of unresolved WDMS objects that we could observationally find if our model is true. Assuming that this is the case, we can estimate the completeness of the observed sample, yielding a value of $\simeq 80 \pm 9$ per cent. Moreover, we can derived the space density for the 139 stars, obtaining a value of $3.3 \times 10^{-5}\,{\rm pc}^{-3}$, in agreement previous results (Schreiber & Gänsicke 2003). The unresolved WDMS objects within the WDMS region represent only the 9.22% of the total unresolved WDMS population (see Table 4.2), and, consequently, the total space density of these objects can be estimate as $3.6 \times 10^{-4}\,{\rm pc}^{-3}$. Finally, the unresolved WDMS population represents a $\sim 7.3\%$ of the WD space density calculated by Jiménez-Esteban et al. (2018).

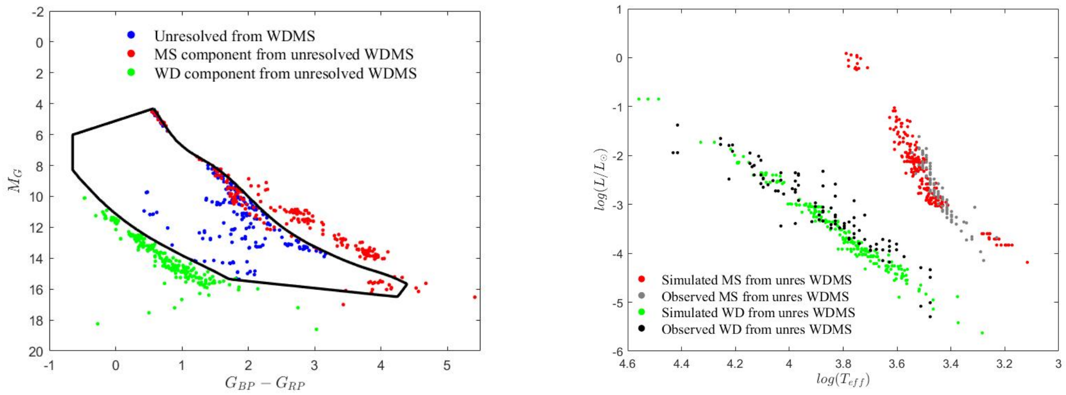


Figure 4.5: Left panel: HR diagram in *Gaia* magnitudes of the unresolved simulated WDMS populations and its components. Right panel: the HR diagram in log(T_{eff}) vs log(L) units of the components of the WDMS unresolved systems, simulated and observed.

A comparison between the density distributions of the simulated and observed samples belonging to this sub-population is shown in Figure 4.6. For this particular model there are notable differences between them, for instance there is an isolated cluster at the upper part of the area in the simulated population, at the edge between the WDMS and MS zones, that does not exist in the observed one. This can be explained as the region of the MS stars with spectral types F and G is highly dense for the unresolved WDMS population, and therefore some of them leak through the edge of the MS area into the WDMS one. For this model as well, the distribution does not occupy all the width of the WDMS region as the observed one does. However, in the observed population there is a perturbation in the lower part of the region that it is reproduced by this simulated model sample.

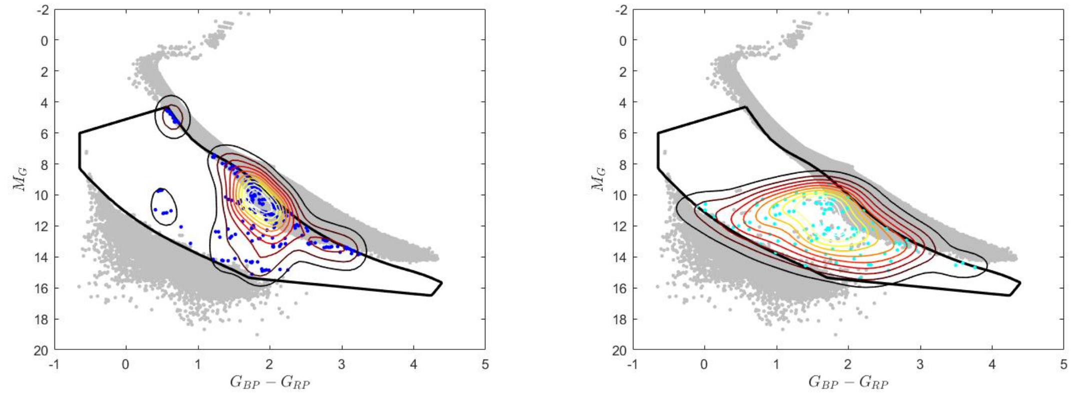


Figure 4.6: Density map of the unresolved WDMS systems within the WDMS area (black solid line). The left panel shows the simulated population, the right one the observed population.

Figures 4.7 and 4.8 provide histograms for different stellar parameters of the MS and WD components, respectively, that are part of unresolved WDMS systems. In the same figures it is displayed as well the data from the observed population. We can see that the distributions of those systems that fall inside the WDMS area agree reasonably well with the observed ones. For the MS components, those that are inside the area have an intrinsically lower masses and effective temperatures and a average surface gravity of 5 dex. With respect to the $T_{\rm eff}$ distribution, there is a gap in the simulated sample around 4, 500 to 5, 500 K that does not appear in the observed one. Moreover, the simulated distribution extends towards higher $T_{\rm eff}$. In a similar way, the mass of the simulated MS components inside the WDMS region extends to larger values. For the WD components of these systems, in general there is no visible pattern for the ones inside the WDMS region. The exception would be that systems with a high mass WD (or high surface gravity) component do not fall inside the region in both, simulated and observed, cases. Another peculiarity in the WD mass distribution histogram is that in the observed sample there is a considerably larger number of objects at around 0.2M$_\odot$ that are not present in the simulated sample.

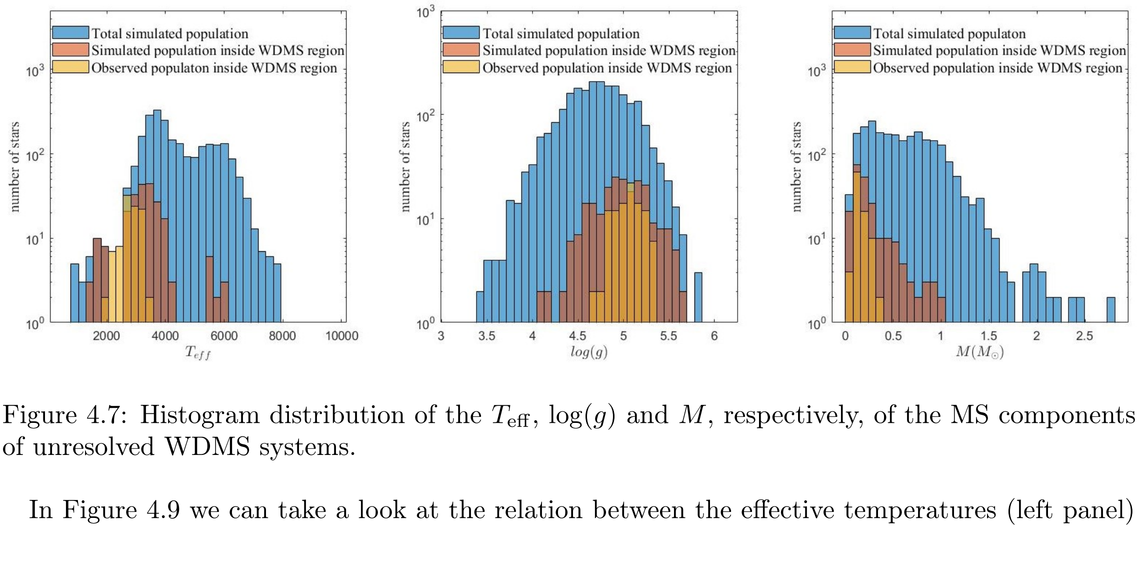


Figure 4.7: Histogram distribution of the $T_{\rm eff}$, $\log(g)$ and M, respectively, of the MS components of unresolved WDMS systems.

In Figure 4.9 we can take a look at the relation between the effective temperatures (left panel)

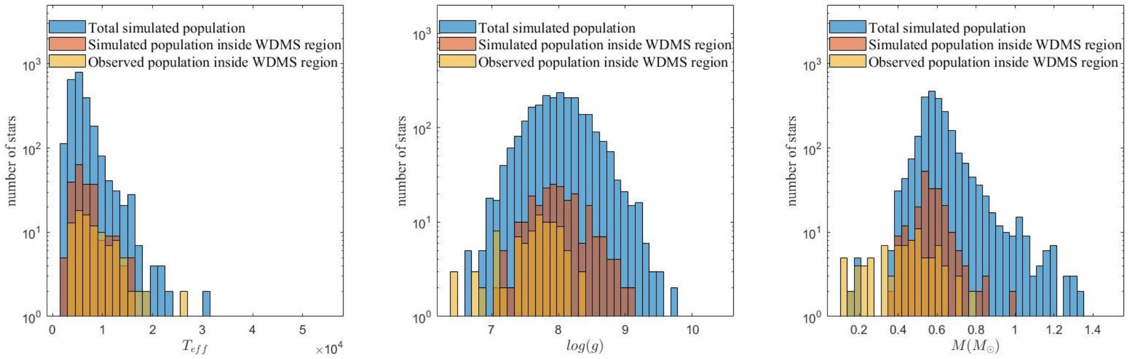


Figure 4.8: Histogram distribution of the $T_{\rm eff}$, $\log(g)$ and M, respectively, of the WD components of unresolved WDMS system

and masses (right panel) of the two components of the unresolved WDMS binary systems. Most of the systems have a WD $T_{\rm eff}$ between $5,000\,$K and $10,000\,$K with some values going up to $15,000\,$K, and a MS $T_{\rm eff}$ in the range between $3,000\,$K and $7,000\,$K. However, there are exceptions for this general tendency, for instance, there are some clusters formed for very low MS temperatures, that contain simulated systems inside and outside the WDMS region as well as some of the observed ones. In any case, it seems that the systems that fall inside this region, simulated or observed, have no specific range of temperatures for the WD component, and the simulated and observed populations agree reasonable well.

For the mass distribution, the bulk of systems is concentrated in a rectangular zone where the WD masses range from $0.4\,{\rm M}_\odot$ to $0.7\,{\rm M}_\odot$ and the MS companion between $0.1\,{\rm M}_\odot$ (as we defined the minimum in Chapter 3), and $1.5\,{\rm M}_\odot$. Again, there are some objects that do not follow the general tendency, for example those that have the WD companion with higher masses, up to near the Chandrasekhar limit, and the systems with MS component masses up to $2.5\,{\rm M}_\odot$. As explained before, the MS component tends to be less massive, while the WD mass takes all possible values. There are significant differences between the observed and the simulated populations that include the ones previously mentioned, i.e. the group of simulated systems with higher MS masses, and the observed systems with extremely low-mass WDs. At the same time there is a significant cluster of objects with low mass WDs that have MS companions with masses around $1.5\,{\rm M}_\odot$.

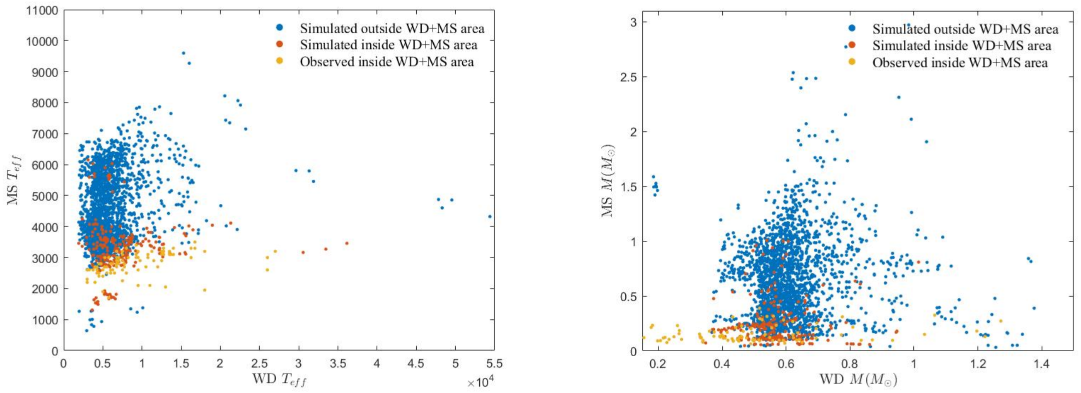


Figure 4.9: Left panel: WD $\rm T_{eff}$ vs. MS $\rm T_{eff}$ for WDMS unresolved systems. Right panel: the same as the left panel but for the component masses.

4.3.2 Resolved WDMS systems

The two components that are part of resolved WDMS systems are displayed in the left panel of Figure 4.10. In this case, the fact of being resolved translates into a considerably larger angular separation, therefore it is not likely that the components of such systems interacted with each other, or at least this interaction would be significantly less important than for the unresolved ones. Consequently, we can assume that the components evolved avoiding mass transfer episodes and fall exactly in the same regions as single stars are located. Also shown in the right panel of of Figure 4.10 is the observed population of resolved WDMS systems (cyan dots) derived from the common-proper motion analysis of the *Gaia* sample (Jiménez-Esteban et al. priv. comm). Although the general agreement of both samples, observed and simulated, is good some peculiarities can be detected. The most significant is the lack of observed cool and massive WDs (that is objects with $M_{\rm G} > 16$) that, on the other hand, seem to be very numerous in the simulated sample.

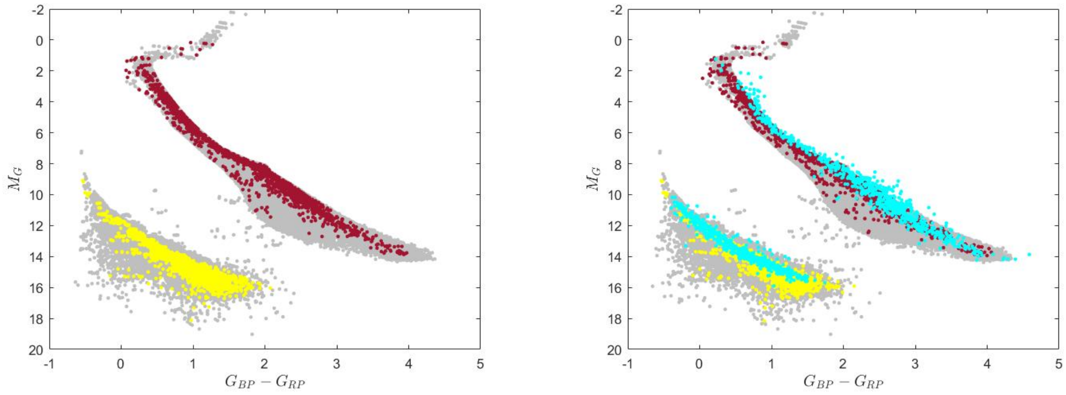


Figure 4.10: HR diagrams in *Gaia* magnitudes. Left panel: the MS stars (red) and WDs (yellow) that are part of resolved WDMS systems. In gray we show all the remaining sub-populations from our simulations. Right panel: superposed the observed *Gaia* population (cyan; (Jiménez-Esteban et al. priv. commm).

For comparison with the unresolved WDMS case, we show in Figure 4.11 the relation between the effective temperatures (left panel) and masses (right panel) of the two components. In this case, since there is not yet an estimation of the previous parameters for the observed sample, we only considered the simulated ones. At a first glance it seems that the general shape of the distributions are maintained with respect to the unresolved case. For the temperatures however there is no single MS component below 2,500 K, unlike the case of the unresolved systems. For the masses, one can see that no WD component reaches a value lower than $\sim 0.5\,{\rm M}_{\odot}$, as expected since there is no mass transfer episodes.

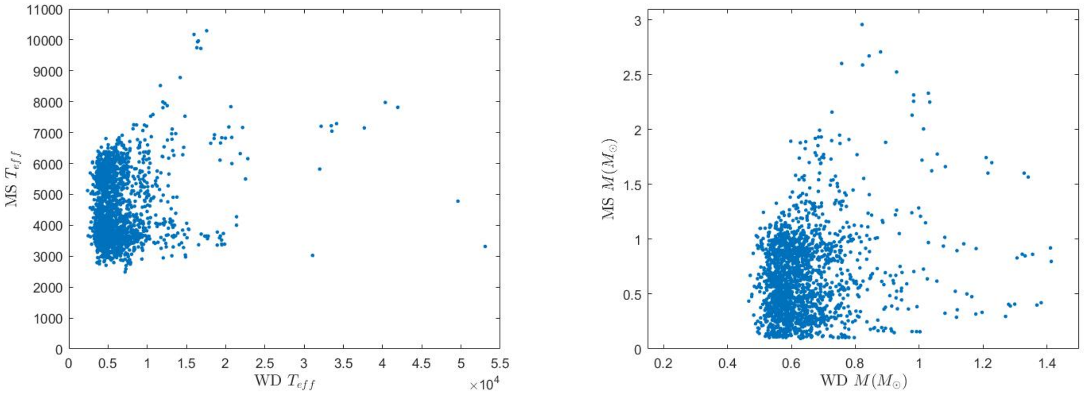


Figure 4.11: Left panel: WD $T_{\rm eff}$ vs. MS $T_{\rm eff}$ for WDMS resolved systems. Right panel: the same as the left panel but for the masses of the components.

4.4 The DWD sub-population

4.4.1 Unresolved DWD systems

The location of the unresolved DWD population in the HR diagram is shown in purple in the top panel of Figure 4.12. The vast majority of them (92.98%) lie in the WD region, while only an small fraction (7.08%) can be found in the WDMS region. In the bottom left and right panel of Figure 4.12 we take a closer look to the WD region and display the density maps of the simulated unresolved DWD and single WD population (WDs), respectively, superimposed to the observed WD sample. From these panels we can see that the unresolved DWD systems tend to have higher values of M_{G} than WDs, as expected because of the combined fluxes of the two WD components. However, the unresolved DWD sub-population is brighter, while the single WDs do not reach the upper limit in the WD region of the observed sample. At the same time, DWD systems tend to be cool objects, i.e., $M_{\mathrm{G}} > 14$ and $1 < G_{\mathrm{BP}} - G_{\mathrm{RP}} < 2$. This effect has been discussed by Bergeron et al. (2019) (see their figure 3). In their analysis, no WD atmospheric model, nor hydrogen pure nor helium pure models, of single WD is able to reproduce those values of magnitudes and colors. However, the concentration of DWD systems in that particular region of the HR diagram ($M_{\mathrm{G}} > 14$ and $1 < G_{\mathrm{BP}} - G_{\mathrm{RP}} < 2$) appears as a plausible explanation of the excess of objects discussed by Bergeron et al. (2019).

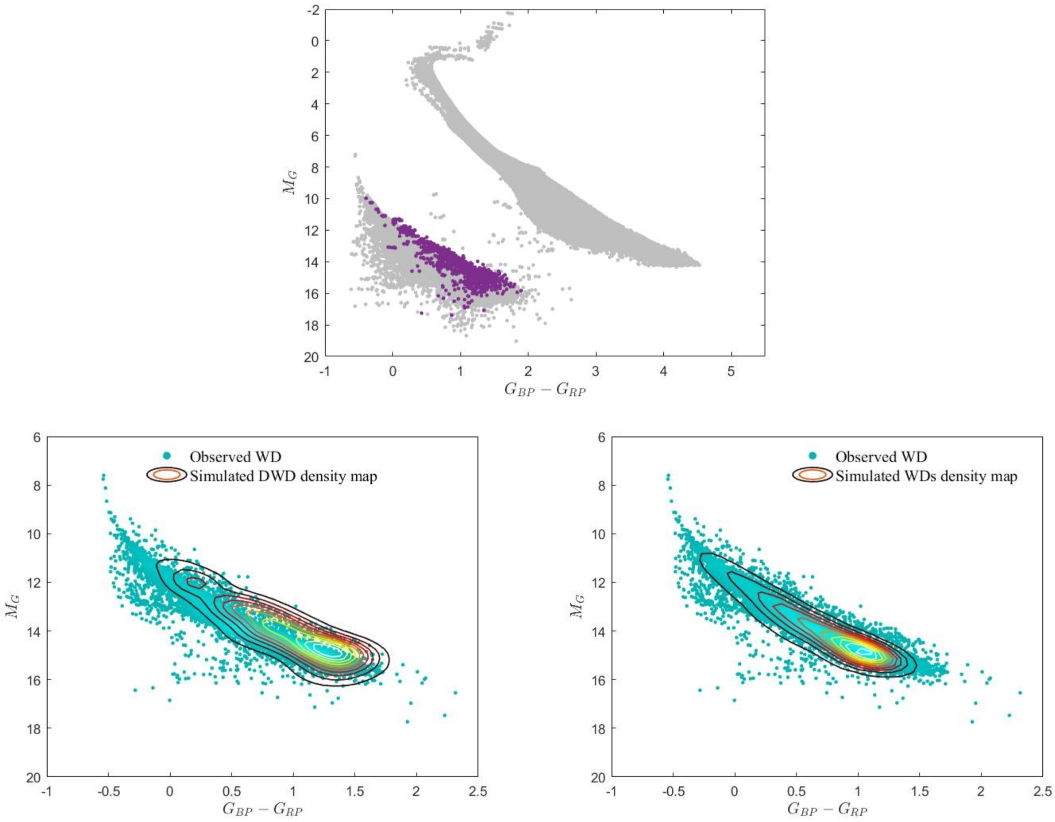


Figure 4.12: HR diagrams in *Gaia* magnitudes. In the top panel the unresolved DWD sub-population is shown in purple, in gray the other populations. In the bottom left panel we zoom-in to the WD region where we display the observed WD in cyan and the density map of the simulated unresolved DWD population. In the bottom right panel we display the observed WD in cyan and the density map of the simulated WDs.

4.4.2 Resolved DWD systems

For the previously analyzed resolved WDMS sub-population, it was easy to tag the two different components, i.e. WD and MS. However, for the resolved DWD case we use the terminology WD1 and WD2, where WD1 is the brighter and WD2 is the fainter component of the system. In other words, $M_{\mathrm{G}_{\mathrm{WD1}}} < M_{\mathrm{G}_{\mathrm{WD2}}}$.

In Figures 4.13 and 4.14 we show in the left panels the loci of the WD1 and WD2 components, respectively, and in the right panels, the simulated populations superposed to the observed one (Jiménez-Esteban, priv. comm.). The observed sample has considerably fewer objects, therefore it is expected some sort of discrepancy between the two samples. However, in both cases, WD1 and WD2, the simulated objects seem to be more spread over the HR diagram than the observed population.

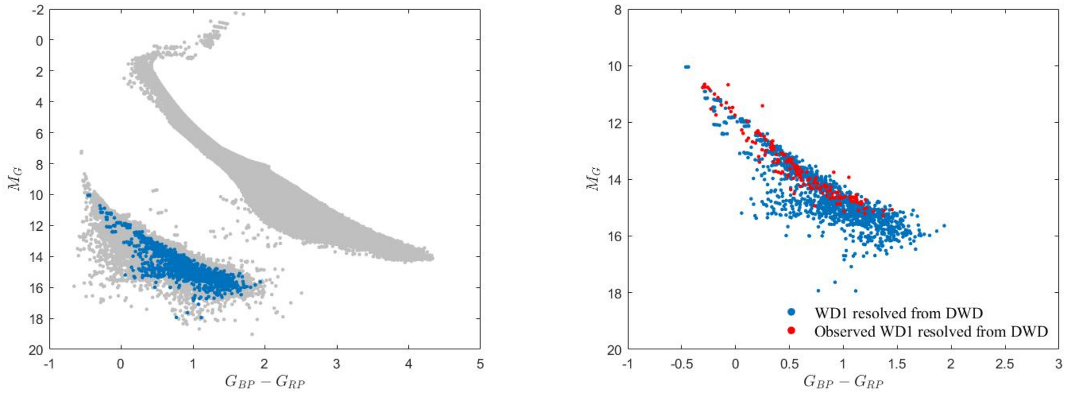


Figure 4.13: HR diagrams in *Gaia* magnitude. The left panel displays in dark blue the WD1 synthetic population, in gray the other populations. In the right panel the simulated WD1 sample is displayed together with the observed WD1 population.

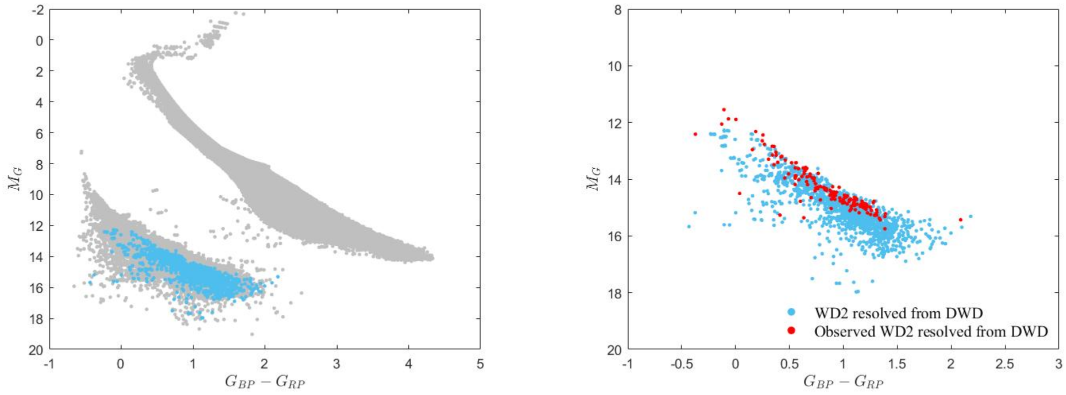


Figure 4.14: HR diagrams in *Gaia* magnitude. The left panel displays in light blue the WD2 synthetic population, in gray the other populations. In the right panel the simulated WD2 sample is displayed together with the observed WD2 population.

Figure 4.15 represents the relation between the T_{eff} of the two components in our resolved DWD systems for both the simulated population and the observed one. It can be seen that both populations follow the same tendency. That is, for lower values of the effective temperature (under $10,000\,\mathrm{K}$ approximately) the two components tend to have similar T_{eff}, whilst the dispersion around equal temperatures increases for higher temperature values.

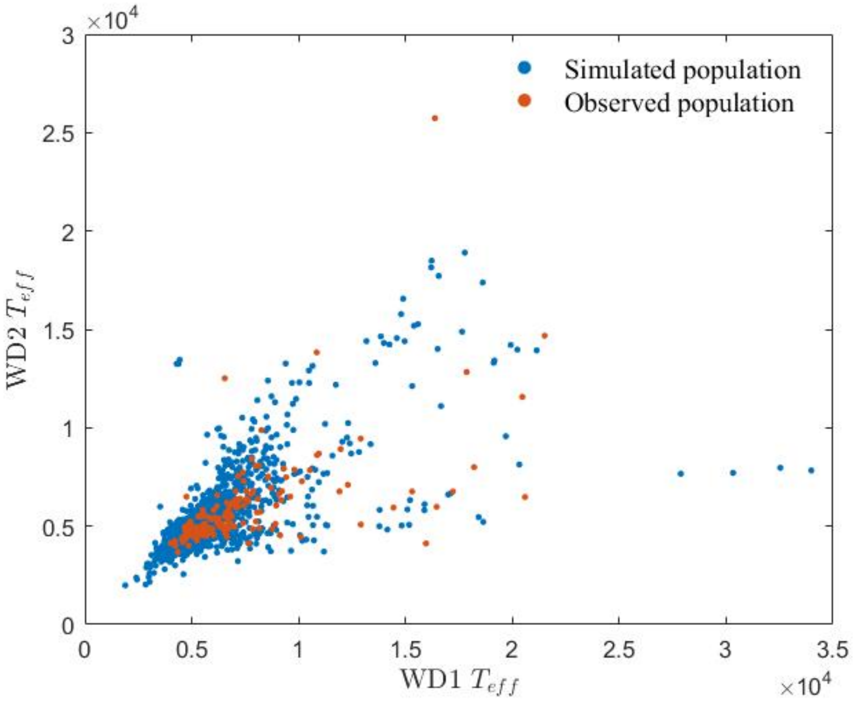


Figure 4.15: The relation between the effective temperatures of WD1 and WD2 that are part of resolved DWD systems.

4.5 The single WD population

The two possible types of genuine single WD populations are shown in figure 4.16, the left panel corresponds to purely single WDs and the right one to single WDs that arise as a result of mergers, i.e. the WDm.

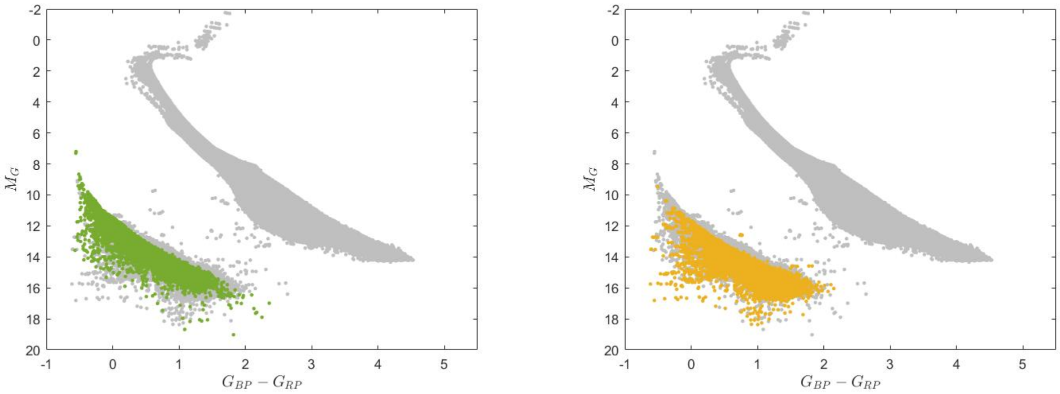


Figure 4.16: HR diagrams in *Gaia* magnitudes. In the left panel we show in green the purely single WDs, in gray the other populations. In the right panel we show in orange the single WDs that result from mergers, in gray the other populations.

Observationally it is not possible to distinguish between WDs that came from a merger of a binary system or that have been always isolated, as there is not footprint that allow to decipher the evolutionary history of the progenitor of the WD. It is possible, though, to make such distinction in our simulations. For this reason in Figure 4.17 we compare the mass distributions of the single WD observed data from Jiménez-Esteban et al. (2018) with the simulated mass distributions, including a breakdown into purely single and mergers for the last sample. The mass distribution histogram has been normalized to the observed sample. At a firs glance it may seem that the simulated WD masses dangerously extend towards values too close to 1.5M$_\odot$, but in any case they exceed the Chandrasekhar mass limit.

As expected, the simulated purely-single WDs tend to be less massive than the WDm, even so, both populations occupy the same rang of masses, from ∼0.4M$_\odot$ up to the limit. To give some numbers, the peak for the isolated WDs is placed at 0.53 M$_\odot$ and for the WDm it is a little bit higher, at 0.59 M$_\odot$. An analogous situation occurs with the mean value of the two populations, which are 0.640 M$_\odot$ and 0.7531 M$_\odot$ for the WDs and WDm stars, respectively. Therefore, as one may expect, the total simulated population, being the sum of the isolated WDs and the WDm, the peak is somewhere in between, at 0.55 M$_\odot$, as it is the main value, 0.6845 M$_\odot$. It is reasonable to expect that the values of the total population are closer to the ones for the isolated WD stars as the number of WDs is considerably higher than the WDm (see table 4.1). When comparing the total simulated population with the observed one, the maximum of the distribution falls into the same bin, and the observed mean value is very similar, 0.6886 M$_\odot$. Nonetheless there is a rather shocking difference between them, the observed population has a second peak for higher masses, at ∼0.75M$_\odot$. This phenomena was already reported by Jiménez-Esteban et al. (2018) and is still a matter of debate.

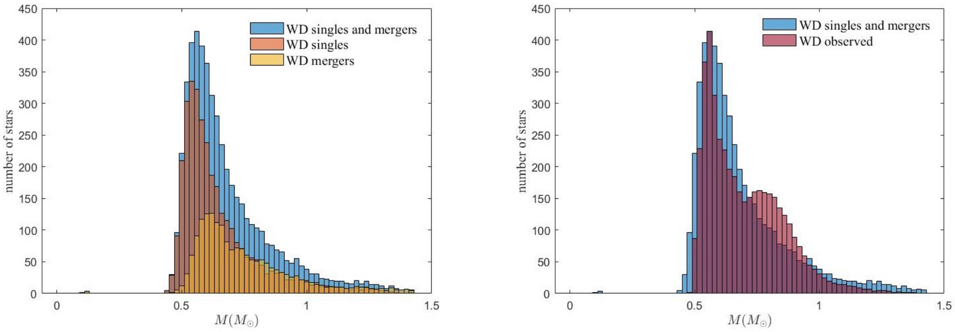


Figure 4.17: Left panel: mass histogram distribution of the single WDs, merger WDs populations and the two populations together. Right panel: mass histogram of the simulated single and merger WD population, superposed the observed sample from Jiménez-Esteban et al. (2018).

Chapter 5

Fitting the binary population models

So far, we have only considered one model in our simulations, with the parameters of the binary fraction fixed to 50% and the IMRD as a flat distribution, as specified in Chapter 2. However, those are not necessarily the ones that provide the best fit to the observed population. In this Chapter we present the results for different values of the previous mentioned parameters. To be precise, we take a closer look at the results obtained when assuming two other IMRD, as well as when using a wide range of values for the binary fraction. Then we discuss which one best agrees with the observational data. To compare the different models with the observed population we focus on analysing the number of unresolved WDMS that are in the WDMS region, as well as the shape of the corresponding stellar parameter distributions, and the fraction of resolved DWD systems and resolved WDMS in the WD region.

5.1 Varying the IMRD

As stated in Chapter 2, there are mainly three different definitions for the IMRD, $n(q) = 1$, $n(q) \propto q$ and $n(q) \propto q^{-1}$. The model presented so far was build according to the flat distribution ($n(q) = 1$). Henceforth we refer to it as Model 1. We name the $n(q) \propto q$ model as Model 2 and the $n(q) \propto q^{-1}$ model as Model 3. For now we keep the same binary fraction of 50%.

As we did for Model 1 in Chapter 4, we summarize the information of these two new models in Tables 5.1 and 5.2 respectively for Models 2 and 3. From them we can see that there are some general features that remain unchanged. The WD populations are mostly in the WD region with some of them entering the WDMS area, and the MS populations are placed in the MS zone with some also falling within WDMS area. In other words, the distribution across the three main regions in the HR diagram remain practically constant, as it can be verified at looking to the blue percentages from Tables 4.2, 5.1 and 5.2. Nonetheless, there are significant differences in the total number of objects belonging to each sub-population (percentages in red), that directly affect the composition of a given region. The exception is the MS region, which maintains a predominance of single MS stars with some MS from resolved WDMS systems and WDMS unresolved systems that represents just ∼1% each one.

Despite the fact that in all three models the majority of objects within the WDMS region is dominated by single (halo) MS stars (more than the 89%), there is an important change in the percentage of unresolved WDMS that are in this region, due to the differences in the number of these systems in the area among the three models. Model 1 had 139 unresolved WDMS inside; Model 2 has about one third of that, 53; and Model 3, 404, which is three times more that the number of Model 1. This tendency goes in line with the total number of unresolved WDMS, therefore it is reasonable that the more stars there are, the more fall into this zone. Given that the number of simulated stars from Model 1 is closer to the observed one, Models 2 and Models 3 seem, a priori, less favoured. In any case, the percentage of the WD populations within this area

	Total	MS region			WDMS region			WD region		
WD singles	8453	0	0%	0%	9	0.23%	0.11%	8444	53.60%	99.89%
WD mergers	3712	0	0%	0%	8	0.20%	0.22%	3704	23.51%	99.78%
MS singles	85536	81701	98.03%	95.52%	3835	96.07%	4.48%	0	0%	0%
MS resolved	689	674	0.81%	97.82%	15	0.38%	2.18%	0	0%	0%
WD resolved	689	0	0%	0%	0	0%	0%	689	4.37%	100%
WDMS unres	1027	968	1.16%	94.26%	53	1.33%	5.16%	6	0.04%	0.58%
WD 1	1048	0	0%	0%	1	0.02%	0.10%	1047	6.65%	99.90%
WD 2	1048	0	0%	0%	1	0.02%	0.10%	1047	6.65%	99.90%
DWD unres	886	0	0%	0%	70	1.75%	7.90%	816	5.18%	92.10%
Total	103088	83343			3992			15753		

Table 5.1: Summary of the percentages of the different sub-populations according to the three general regions once the normalization is made for Model 2. In red, the percentage that the sub-population represents with respect to the total number of stars in that region. In blue the percentage of a given sub-population that falls into that region.

always represents less than the 3% of stars in the area.

	Total	MS region			WDMS region			WD region		
WD singles	9219	0	0%	0%	7	0.15%	0.08%	9212	58.48%	99.92%
WD mergers	3889	0	0%	0%	10	0.22%	0.26%	3879	24.62%	99.74%
MS singles	93053	88867	96.61%	95.50%	4187	89.26%	4.50%	0	0%	0%
MS resolved	1626	1571	1.71%	96.62%	55	1.17%	3.38%	0	0%	0%
WD resolved	1626	0	0%	0%	3	0.06%	0.18%	1623	10.30%	99.82%
WDMS unres	2000	1544	1.68%	77.20%	404	8.61%	20.20%	52	0.33%	2.60%
WD 1	369	0	0%	0%	0	0%	0%	369	2.34%	100%
WD 2	369	0	0%	0%	1	0.02%	0.27%	368	2.34%	99.73%
DWD unres	275	0	0%	0%	24	0.51%	8.73%	251	1.59%	91.27%
Total	112426	91982			4691			15754		

Table 5.2: Same as Table 5.1 but for Model 3.

The fraction of the different populations considered inside the WD region vary considerably depending on the adopted IMRD model. The most notable changes are the ones suffered by the resolved WDs that are part of WDMS systems and the DWD systems, which is a direct result of the variation of the total number of these systems. The variations in the number of resolved WDMS systems follows the same pattern as the unresolved WDMS ones, meaning that Model 2 has the lowest amount, Model 3 the highest, and Model 1 lies somewhere in between. For the DWD systems, resolved or not, we see the opposite trend: Model 2 is the one with most DWD systems and Model 3 the lowest. Consequently, the fraction for those populations inside the WD region changes according to these tendencies. On the other hand, purely-single WDs and WDm always represents about 1/2 and 1/4 of the population in the WD area, respectively. Similarly, for all three models, the number of unresolved WDMS inside this area remains extremely low, $<1\%$, which means that only for a small fraction of unresolved WDMS the WD component is the one that dominates.

In table 5.3 we summarize the information of the fractions of the different populations that are located within the WD region for the three models as well as for the observed ones. The information for Model 1 is the same of the diagram in Figure 4.3. As explained in Chapter 4, the only observational values that we have access to, within the WD region, are the ones for the resolved WDMS and DWD sub-populations.

Figure 5.1 shows the density maps of the unresolved WDMS inside the WDMS region for the two new models. It is obvious that both distributions are extremely different from each other. Model 2 exhibits two well defined peaks in the density map (like Model 1), a cluster in the upper part of the area (but more populated) and another one for $G_{\rm BP} - G_{\rm RP} \approx 1.75$, both in the frontier

	Observed	Model 1	Model 2	Model 3
WDMS res	6.30%	7.90%	4.69%	10.55%
DWD res	1.20%	5.82%	7.12%	2.40%
WD singles	92.50%	52.69%	57.42%	59.87%
WD mergers		28.92%	25.18%	25.21%
WDMS unres		0.04%	0.04%	0.34%
DWD unres		4.63%	5.55%	1.63%
Total others	92.50%	86.28%	88.19%	87.05%

Table 5.3: The fractions of the different populations in the WD region for the observed data and the simulations of Models 1, 2 and 3, being *Total others* the sum of the WDs, WDm, WDMS unresolved and DWD unresolved populations

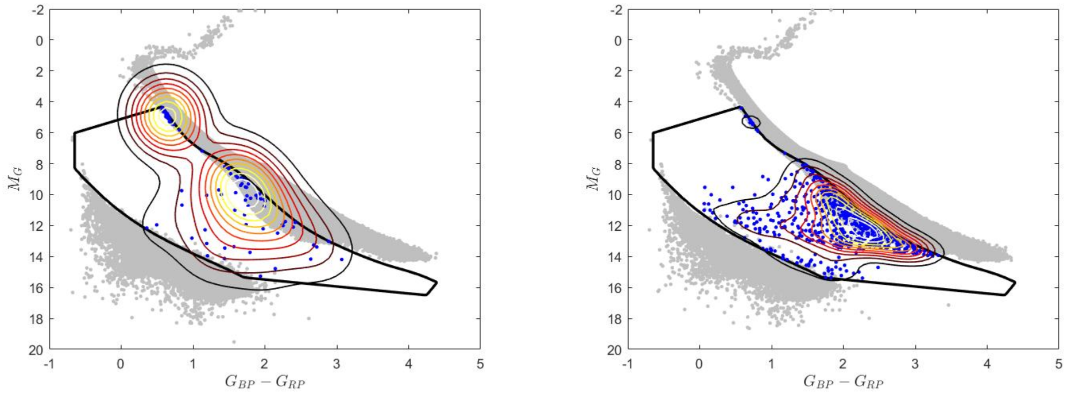


Figure 5.1: Density maps of the unresolved WDMS systems in the WDMS region for Models 2 (left panel) and 3 (right panel).

with the MS region. On the other hand, Model 3 presents a more spread distribution across the WDMS region, which has a higher resemblance to the observed distribution. An upper cluster is still present, however, it is a minority compared to the rest. In both cases, it seems that the distributions tend to be closer to the MS limit, while in the observed case, it is quite centered in the area.

5.2 Varying the binary fraction

In this section we analyse how the changes in the assumed binary fraction affect the outcome of the simulations. Given that Model 1 provides a numbers of binaries and single stars that are in better (although not full) agreement with the observed values, we adopt $n(q) = 1$ as our reference model when varying the binary fraction. The binary fractions that we considered are 30%, 40%, 60% and 70% in addition to the 50% that we previously presented. For all those values, the distribution in the MS region remains practically unaltered in comparison with the original assumed value of 50%. However, there is a significant change in the number of unresolved WDMS that are placed inside the WDMS region, with this number increasing as the binary fraction increases. The 30% model includes 69 stars in the area, 100 the 40% one, 173 the 60% binary fraction model, and finally, 200 the 70% one. This behaviour should be expected since an increase in the binary fraction translates into an increase in the number of WDMS systems that fall into that area.

As shown in table 5.4, the percentages of the different sub-populations that are located in the WD region change also according to the assumed binary fraction. As expected, there is a clear tendency for increasing the proportion of binary systems, WDMS or DWD, resolved or not, with the binary fraction. The WDm population follows this pattern as well due to the fact that these WDs are formed as a result of a merger of a binary system, therefore an increase in the binary

	Observed	30% model	40% model	50% model	60% model	70% model
WDMS res	6.30%	3.74%	5.34%	7.90%	7.04%	10.30%
DWD res	1.20%	3.21%	4.14%	5.82%	6.37%	7.59%
WD singles	92.50%	76.68%	67.90%	52.69%	49.70%	37.92%
WD mergers		14.12%	19.53%	28.92%	31.90%	38.05%
WDMS unres		0.07%	0.09%	0.04%	0.10%	0.15%
DWD unres		2.18%	3.00%	4.63%	4.89%	5.99%
Total others	92.50%	93.05	90.52	86.28	86.54	82.11

Table 5.4: Fraction of the different sub-populations in the WD region for the observed data and the simulations of the different binary fractions from 30% to 70% in steps of 10%, being *Total others* the sum of the WDs, WDm, WDMS unresolved and DWD unresolved populations.

fraction will also result in a growth in the number of WDm stars. The only sub-population that shows an inverse relation with the binary fraction is that of purely-single WDs, as it is the only population that involved a single star from the beginning of its evolution. Comparing the proportion of resolved WDMS systems to the observed one, we see that the values are in better agreement when assuming the 40% or 50% binary fractions. Meanwhile, all the simulations give a percentage of the resolved DWD systems that is larger than the observed one, independently of the assumed binary fraction.

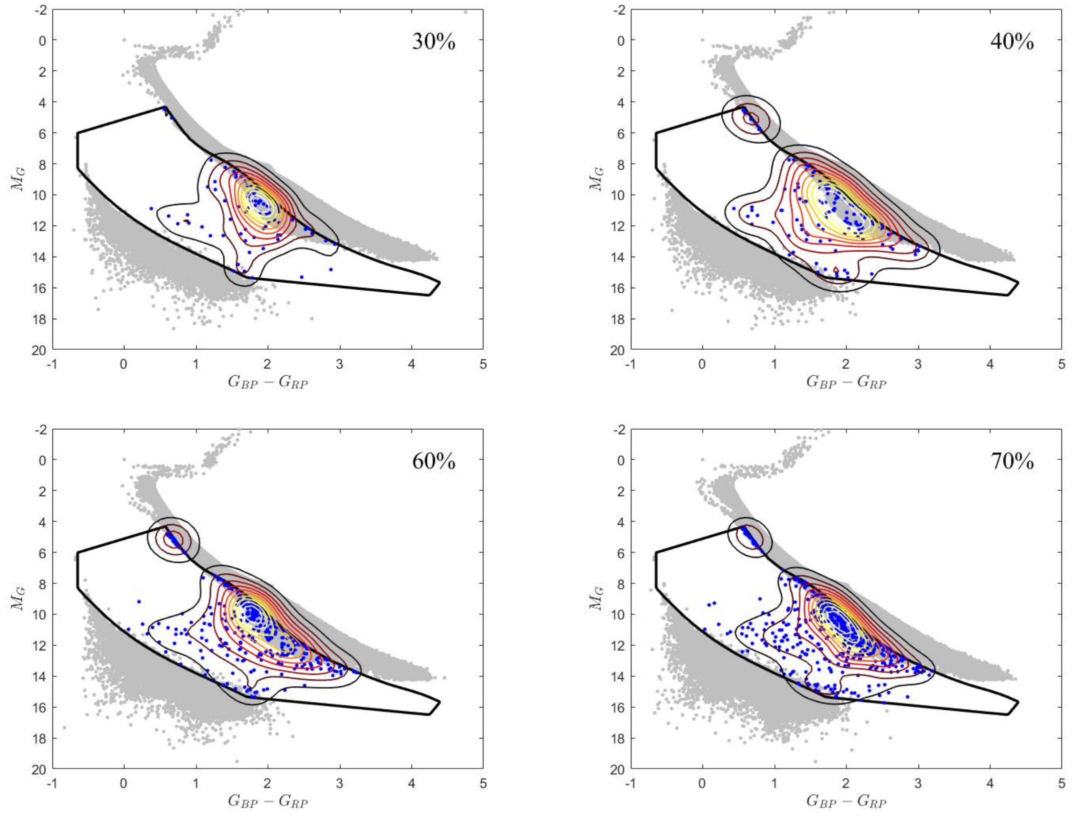


Figure 5.2: Density maps of the unresolved WDMS systems in the WDMS region for different binary fractions (top left 30%; top right 40%; bottom left 60%; bottom right 70%.

In Figure 5.2 we compare the distributions of the unresolved WDMS systems inside the WDMS region of the HR diagram for the different binary fraction models. The main difference between them is that, as the binary fraction increases, so does the number of objects and the cluster in the top right gains importance. In all cases most of the objects are located near the MS area, however not always the synthetic objects reach the other extreme of the zone, that is, the frontier with the WD region.

5.3 A first approach to the best fitting model

To summarise, the different models we presented are: Model 1, with $n(q) = 1$ and binary fraction 50%; Model 2, with $n(q) \propto q$ and binary fraction 50%; Model 3, with $n(q) \propto q^{-1}$ and a binary fraction of 50%; 30% model, with $n(q) = 1$ and 30% of binary fraction; 40% model, with $n(q) = 1$ and 40% of binary fraction; 50% model and $n(q) = 1$, which is the the same as Model 1; 60% model, with $n(q) = 1$ and 60% of binary fraction; and 70% model, with $n(q) = 1$ and 70% of binary fraction.

With all the information presented in this chapter we find that there is no straightforward way for choosing the model that best fits with the observed sample. For instance, Model 3 has the closest proportion of resolved DWD systems to the observed one, however, the number of unresolved WDMS systems in the WDMS region is four times the number observed. On the other hand, Model 1 provides a reasonably close value to the observed one for the unresolved WDMS systems in the WDMS zone and the percentage of resolved WDMS in the WD area, but the fraction of resolved DWD is considerably higher and the upper cluster seems more populated that the one in Model 3. Nonetheless, we can confidently conclude that Model 2 does not present a good resemblance with the observed population, as the distribution is too different from the observed one, there are too few unresolved WDMS and too many resolved DWD systems in the WD area.

We stated in Section 5.2 that higher values of the binary fraction result in higher values of the fraction of resolved DWD and WDMS systems in the WD area as well as in more unresolved WDMS placed in the WDMS region. By comparing these percentages to the observed ones, we consider that the best fit is the one with a binary fraction of 40%. This model yields 100 unresolved WDMS inside the WDMS area, which agrees very well with the 98 observed, and the proportion of resolved WDMS in the WD area has a difference of less than 1%. Of course, this model can not yet be considered as optimal because there are issues such as the high fraction of DWD systems that are not reconciled with the observations.

For a further check we preformed a two-dimensional two-sample Kolmogorov-Smirnov test, which allowed us to make a more exhaustive and quantitative comparison between the different distributions, for instance, of unresolved WDMS systems placed in the WDMS region, than simply looking at their shapes. This test gives the largest absolute difference between the two cumulative probability distributions, therefore a lower value means a better resemblance between the two distributions.

The results of the test, summarized in Table 5.5, pointed out that the best fit is Model 1 with a 40% of binary fraction, being a close second the 30% case. Model 3 obtained the third best results in the Kolmogorov-Smirnov test, probably because of the low number of stars in the upper cluster, and the higher spread of the main group across the area.

	KS
Model 1	0.3150
Model 2	0.5241
Model 3	0.2961
30% model	0.2913
40% model	0.2871
60% model	0.3108
70% model	0.3355

Table 5.5: Values of the outcome of the Kolmogorov-Smirnov test for the different models, being KS the largest absolute difference between the two cumulative probability distributions.

In the present work we have discussed how the outcome of the simulations change depending on only two parameters. This is providing us a first guess of the general tendencies regarding the changes in the numbers and fractions of the different populations across the HR diagram. So far, Model 3 exhibits a nice distribution of the stars in the WDMS region and has the lowest value of the resolved DWD proportion, but the values of the unresolved WDMS systems in the WDMS area and the resolved WDMS in the WD area remain too high. If we lower the binary fraction there is a chance that those values decrease to match with the observed ones. However, lowering the fraction would not be a realistic solution. So other parameters that characterize the modeling of the binary population should be taken into account, such as the common-envelope efficiency, initial separation distribution, mass losses, among others. A thorough analysis involving such parameters and comparison between the simulated and observed distributions is beyond the present work, although is currently underway and will be part of a forthcoming publication.

Conclusions

We have conducted a population synthesis study of the white dwarf population within 100 pc from the Sun based on a detailed Monte Carlo code, which provided synthetic samples of both single and binary stars for different input parameters.

We firstly derived the *Gaia* magnitudes from the output parameters of the simulated samples. Two models were considered: MIST and PARSEC. Despite some issues with the MIST models presumably due to flux calibration of the bolometric corrections, that imply a shift in the HR-diagram of the position of the MS trackfor, we presented a temporal solution to this situation, which consisted in preforming an adjustment by hand. Afterwards we found an alternative method, based on PARSEC models, to derive the *Gaia* magnitudes which did not exhibit such visible displacement.

Other adjustments were required in order to compare the observed and simulated populations. For instance the addition of a metallicity dispersion for the MS stars, as the simulation outcome drew three perfectly distinct tracks in the MS region of the HR diagram, which is not realistic when comparing to the observed data. Another important effect we had to consider was the inclusion of photometric and astrometric errors in all simulated sub-populations, to take into account the instrumental limitations and physical errors affecting the observed sample, following the *Gaia* prescriptions. The implemented dispersion and inclusion of photometric and astrometric errors broadened the MS sub-populations achieving a very good resemblance with the observed MS data in the HR diagram.

Once the synthetic populations were adequately calibrated, the simulated data proved to be an excellent source of information. Among the most relevant results we derived an estimation of the completeness of the unresolved WDMS population. From the observed sample, consisting in 112 unresolved WDMS binary candidates and the expected number of objects in our synthetic population, we were able to derive a completeness value of $\simeq 80 \pm 9$ per cent, yielding a number density of these objects of $3.3 \times 10^{-5}\,\mathrm{pc}^{-3}$. These results have been summarized and included in Rebassa-Mansergas et al. (2021), in press.

The comparison between the outcome of our population synthesis model and the observed sub-populations showed a satisfactory degree of resemblance between them. Nonetheless, some important differences raised between the two samples. A clear example is the mass distribution of the single WD population, see section 4.5, were the observed distribution exhibits a second peak for higher masses that could to be explained with the information provided by the simulations. Another significant discrepancy is found in the fraction of each sub-population placed in the WD region of the HR diagram. While we obtained reasonably close values for the resolved WDMS sub-population, in the case of resolved DWDs the observed sample represented a considerably smaller fraction than the simulated one. Both issues were reported in previous works and still remain as a matter of debate.

Finally, our last objective was to determine the optimal parameters of our population synthesis model which resulted in a better fitting with the observed *Gaia* samples. When searching for the best fitting model, we focused on comparing the value of simulated versus observed fraction of resolved WDMS and DWD systems in the WD region and the number of unresolved WDMS systems that are located in the WDMS area as well as on their distribution across this region. For the last one we made use of a two-dimensional two-sample Kolmogorov-Smirnov test in order to obtain a more quantitative estimation than a simple visual comparison. The models available to compare were three with different IMRD ($n(q) = 1$, $n(q) \propto q$ and $n(q) \propto q^{-1}$) with the same binary fraction of 50%; and four more models with the same IMRD as the $n(q) = 1$ but changing the binary fractions to 30%, 40%, 60% and 70%.

In view of the results we can conclude that there is not a single model that fits all the different observed percentages of the white dwarf sub-populations. Regarding the IMRD, it seems obvious that the directly proportional IMRD does not provide a satisfactory result, as most of its features differ from the observed sample, specially the distribution of the unresolved WDMS sub-population across the WDMS region. With respecto to Model 1 ($n(q) = 1$) and Model 3 ($n(q) \propto q^{-1}$) not all results are completely positive or negative, as one model's strengths are the other's weaknesses. Therefore there is no clear decision in which of them fits best with the observed sample. Similarly, when changing the binary fraction, the 40% model obtained the most optimal results in all parameters except for the fraction of resolved DWDs in the WD area, which has been the one with more problems to fit. This change of the binary fraction allowed us to draw the general tendencies and showed us that, besides the variation in the proportion of the different binary populations, there is an important difference in the ratio of WDs and WDm.

The current project is limited to seven different outcomes of the simulations, which only consider the variation of two parameters, and always one at the time. For future works we suggest a study of a wider range of models, which take into consideration other parameter modification. For instance, the variation of the different parameters that rule the CE phase ($\alpha_{\rm CE}$, $\alpha_{\rm int}$ and λ) would probably have a huge impact not only in the calculated fraction of the different sub-populations in the WD area or the number of unresolved WDMS that falls in the WDMS region, but also in the evolution of the close binary systems and their distribution across the HR diagram. To give another example, the initial separation is likely to influence the ratio of resolved and unresolved, and consequently, all the parameters we used to discuss the fitting.

References

Althaus, L. G., Córsico, A. H., Isern, J., García-Berro, E. 2010, Astron Astrophys Rev, **18**, 471–566

Althaus, L. G., Camisassa, M., Miller Bertolami, M., García-Berro, E., 2015, A&A, **476**, A9

Althaus, L. G., De Gerónimo, F., Córsico, A., Torres, S., García-Berro, E., 2016, A&A, **597**, A67

Bergeron, P., Dufour, P., Fontaine, G., Coutu, S., Blouin, S., Genest-Beaulieu, C., Bédard, A., Rolland, B., 2019, ApJ, **876**, 67B

Camacho, J., Torres, S., Isern, J., Althaus, L., García-Berro, E., 2007, A&A**471**, 151-158

Camacho,J., 2014, “Monte Carlo simulations of the population of single and binary white dwarfs of our Galaxy", PhD thesis, UPC

Camacho, J., Torres, S., García-Berro, E., Zorotovic, M., Schreiber, M. R., , Rebassa-Mansergas, A., Nebot-Gómez Morán, A., Gänsicke, B. T., 2014, A&A, **566**, A86

Camisassa, M. E., Althaus, L.G., Córsico, A.H., Vinyoles, N., Serenelli, A.M., Isern, J., Miller Bertolami, M.M., García-Berro, E., 2016, ApJ, **823**, 158

Camisassa, M. E., Althaus, L.G., Rohrmann, R., García-Berro, E., Torres, S., Córsico, A.H., Wachlin, F.C., 2017, ApJ, **839**, 1-11

Canals, P., Torres, S., Soker, N., 2018, MNRAS, **480**, 4519–4525

Choi, J., Dotter, A., Conroy, C., Cantiello, M., Paxton, B., Johnson, B. D., 2016, ApJ, **823**, 102

Cojocaru E.R., 2016, “Population synthesis studies of the white dwarfs of the galactic disk and halo”, PhD Thesis, UPC

R. Cojocaru, A. Rebassa-Mansergas, S. Torres, E. García-Berro, 2017, MNRAS, **470**, 1442-1452

Davis P. J., Kolb U., Willems B., 2010, MNRAS, **403**, 179

Dotter A., 2016, ApJS, **222**, 8

García-Berro, E. Torres, S. Isern, J. and Burkert, A., 1999, MNRAS, **302**, 173–188

García-Berro, E. Torres, S. Isern, J. and Burkert, A., 2004, A&A, **418**, 53–65

García-Berro, E., Torres, S. Althaus, L. G., Renedo, I., Lorén-Aguilar, P., Córsico, A. H., Rohrmann, R. D. Salaris, M. and Isern, J., 2010, Nature, **465**, 194-196

Gaia Collaboration, A. G. A. Brown, A. Vallenari, T. Prusti, J.H.J. de Bruijne, et al., 2020, A&A, **649**, A1

Gaia Collaboration, Smart, R. L., Sarro, L. M., Rybizki, J., et al., 2021, A&A, **649**, A6

Riello, M., De Angeli, F., Evans, D. W., et al. 2021, A&A, **649**, A3

Gentile Fusillo, N. P., Tremblay, P. -E., Cukanovaite, E., Vorontseva, A., Lallement, R., Hollands, M., Gänsicke, B. T., Burdge, K. B., McCleery, J., Jordan, S., submitted to MNRAS, arxiv2106.07669

Han, Z., Podsiadlowski, P., Eggleton, P. P., 1995, MNRAS, **272**, 800-820

Heggie, D. C., 1975, MNRAS, **173**, 729–787.

Hurley, J. R., Pols, O. R., Tout, C. A., 2000, MNRAS, **315**, 543–569

Hurley, J. R., Tout, C. A., Pols, O. R., 2002, A&A, **329**, 897–928

Isern, J., 2019, ApJ, **878**, L11

Ivanova, N., Justham, S., Chen, X., De Marco, O., Fryer, C. L., Gaburov, E., Ge, H., Glebbeek, E., Han, Z., Li, X. -D., Lu, G., Marsh, T., Podsiadlowski, P., Potter, A., Soker, N., Taam, R., Tauris, T. M., van den Heuvel, E. P. J., Webbink, R. F., 2013, A&ARv, **21**, 59

Jiménez-Esteban, F. M., Torres, S., Rebassa-Mansergas, A., Skorobogatov, G., Solano, E., Cantero, C, Rodrigo, C., 2018, MNRAS, **480**, 4505-4518

Kippenhahn, R., Weigert, A., Weiss A., 2012, *Stellar Structure and Evolution*, 2nd edition, Springer-Verlag Berlin Heidelberg

Kroupa, P., Tout, C. A., Gilmore, G. 1993, MNRAS, **262**, 545

Peacock, J. A., 1983, MNRAS, **202**, 615–627

Postnov, K.A., Yungelson, L.R., 2006, Living Rev. Relativ. 9, 6.

Rebassa-Mansergas, A., Nebot Gómez-Morán, A., Schreiber, M. R., Girven, J., Gänsicke, B. T., 2011, MNRAS, **413**, 1121-1131

Salpeter E. E., 1955, ApJ, **121**, 161

Schreiber M. R., Gänsicke B. T., 2003,A&A, **406**, 305

Toonen, S.,Hollands, M., Gänsicke, B. T., Boekholt, T., 2017, A&A, **602**, A16

Torres, S., García-Berro, E., Burkert, A., Isern, J., 2001, MNRAS, **328**, 492–500

Torres, S., García-Berro, E., Burkert, A., Isern, J., 2002, MNRAS, **336**, 971–978

Torres, S., 2002, "Simulación de Monte Carlo de la población de enanas blancas de la Galaxia", PhD thesis, UPC

Torres, S., García-Berro, E., Isern, J., Figueras, F., 2005, MNRAS, **360**, 1381–1392

Torres, S., García-Berro, E., Althaus, L. G., and Camisassa, M. E. 2015, A&A, **581**, A90

Torres, S. and García-Berro, E., 2016, A&A, **588**, A35

Torres, S., García-Berro, E., Cojocaru, R., and Calamida, A., 2018, MNRAS, **476**, 1654-1662

Torres, S., Cantero, C., Rebassa-Mansergas, A., Skorobogatov, G., Jiménez-Esteban, F. M., Solano, E., 2019, MNRAS, **485**, 5573–5589

Torres, S., Rebassa-Mansergas, A., Camisassa, M.E., and Raddi, R., 2020, MNRAS, **502**, 1753–1767

Tout, C. A., Aarseth, S. J., Pols, O. R., Eggleton, P. P., 1997, MNRAS, **291**, 732-748

Appendix

Matlab code

```matlab
close all
clear all


%% Load data

% Load the data and change the variables name

% MS+WD binary systems
% mWD(Msun) mMS(Msun) log10(Teff_wd) log10(Teff_MS) log10(L_wd/Lsun)
% log10(L_MS/Lsun)  s(arc sec) par(mas)  index

load('MSWDfb40.out');      MSWD=MSWDfb40;

% WD mergers
% mWD(Msun) log10(Teff) log10( L/Lsun) par(mas) index

load('WDmergersfb40.out'); WDmergers=WDmergersfb40;

% WD singles
% mWD(Msun) Teff logg R(Rsun) G BP RP par(mas) index

load('WDsinglesfb40.out'); WDsingles=WDsinglesfb40;

% WD+WD binary systems
% mWD1(Msun) mWD2(Msun) log10(Teff1) log10(Teff2) log10(L1/Lsun)
% log10(L2/Lsun)  s(arc sec) par(mas) index

load('WDWDfb40.out');     WDWD=WDWDfb40;

% MS singles
% mMS(Msun) log10(Teff) log10(R) log10(L) par index

load('MSfb40.out');       MS=MSfb40;

% Load tables for WD and MS
% Teff log g G_dr3 BP_dr3 RP_dr3
load('WD_Gaia_magn.txt')

% Z logL logTe logg G BP RP
load('PARSEC0p001.txt');
load('PARSEC0p005.txt');
load('PARSEC0p010.txt');
load('PARSEC0p015.txt');
load('PARSEC0p020.txt');
load('PARSEC0p025.txt');
load('PARSEC0p030.txt');
load('PARSEC0p035.txt');
load('PARSEC0p040.txt');
load('PARSEC0p045.txt');


%% Reorganize data for WD

```

```
% Find the values of Teff and logg for which the Gaia magnitudes are available
TeffWD=unique(WD_Gaia_magn(:,1)); loggsWD=unique(WD_Gaia_magn(:,2));

% For some Teff the values of the Gaia magnitudes are available for a range
% of logg from 5-9.5 others from 6-9.5, therefore we need to organize the
% tables into two groups

G_dr3=reshape(WD_Gaia_magn(1:1596,3),length(loggsWD),[]);
G_dr3_2=reshape(WD_Gaia_magn(1597:end,3),length(loggsWD(5:end)),[]);
BP_dr3=reshape(WD_Gaia_magn(1:1596,4),length(loggsWD),[]);
BP_dr3_2=reshape(WD_Gaia_magn(1597:end,4),length(loggsWD(5:end)),[]);
RP_dr3=reshape(WD_Gaia_magn(1:1596,5),length(loggsWD),[]);
RP_dr3_2=reshape(WD_Gaia_magn(1597:end,5),length(loggsWD(5:end)),[]);


%% Reorganize data for MS

% To interpolate Matlab require that the vectors do not include repeated
% values and that they are sorted

[P01a,i01a,¬]=unique(PARSEC0p001(:,2)); G01a=PARSEC0p001(i01a,5);
BP01a=PARSEC0p001(i01a,6);              RP01a=PARSEC0p001(i01a,7);
[P01,i01]=sort(P01a);                   G01=G01a(i01);
BP01=BP01a(i01);                        RP01=RP01a(i01);
[P05a,i05a,¬]=unique(PARSEC0p005(:,2)); G05a=PARSEC0p005(i05a,5);
BP05a=PARSEC0p005(i05a,6);              RP05a=PARSEC0p005(i05a,7);
[P05,i05]=sort(P05a);                   G05=G05a(i05);
BP05=BP05a(i05);                        RP05=RP05a(i05);
[P10a,i10a,¬]=unique(PARSEC0p010(:,2)); G10a=PARSEC0p010(i10a,5);
BP10a=PARSEC0p010(i10a,6);              RP10a=PARSEC0p010(i10a,7);
[P10,i10]=sort(P10a);                   G10=G10a(i10);
BP10=BP10a(i10);                        RP10=RP10a(i10);
[P15a,i15a,¬]=unique(PARSEC0p015(:,2)); G15a=PARSEC0p015(i15a,5);
BP15a=PARSEC0p015(i15a,6);              RP15a=PARSEC0p015(i15a,7);
[P15,i15]=sort(P15a);                   G15=G15a(i15);
BP15=BP15a(i15);                        RP15=RP15a(i15);
[P20a,i20a,¬]=unique(PARSEC0p020(:,2)); G20a=PARSEC0p020(i20a,5);
BP20a=PARSEC0p020(i20a,6);              RP20a=PARSEC0p020(i20a,7);
[P20,i20]=sort(P20a);                   G20=G20a(i20);
BP20=BP20a(i20);                        RP20=RP20a(i20);
[P25a,i25a,¬]=unique(PARSEC0p025(:,2)); G25a=PARSEC0p025(i25a,5);
BP25a=PARSEC0p025(i25a,6);              RP25a=PARSEC0p025(i25a,7);
[P25,i25]=sort(P25a);                   G25=G25a(i25);
BP25=BP25a(i25);                        RP25=RP25a(i25);
[P30a,i30a,¬]=unique(PARSEC0p030(:,2)); G30a=PARSEC0p030(i30a,5);
BP30a=PARSEC0p030(i30a,6);              RP30a=PARSEC0p030(i30a,7);
[P30,i30]=sort(P30a);                   G30=G30a(i30);
BP30=BP30a(i30);                        RP30=RP30a(i30);
[P35a,i35a,¬]=unique(PARSEC0p035(:,2)); G35a=PARSEC0p035(i35a,5);
BP35a=PARSEC0p035(i35a,6);              RP35a=PARSEC0p035(i35a,7);
[P35,i35]=sort(P35a);                   G35=G35a(i35);
BP35=BP35a(i35);                        RP35=RP35a(i35);
[P40a,i40a,¬]=unique(PARSEC0p040(:,2)); G40a=PARSEC0p040(i40a,5);
BP40a=PARSEC0p040(i40a,6);              RP40a=PARSEC0p040(i40a,7);
[P40,i40]=sort(P40a);                   G40=G40a(i40);
BP40=BP40a(i40);                        RP40=RP40a(i40);
[P45a,i45a,¬]=unique(PARSEC0p045(:,2)); G45a=PARSEC0p045(i45a,5);
BP45a=PARSEC0p045(i45a,6);              RP45a=PARSEC0p045(i45a,7);
[P45,i45]=sort(P45a);                   G45=G45a(i45);
BP45=BP45a(i45);                        RP45=RP45a(i45);


%% Some constants

% g=GM/R^2, L=sigm*4*pi*R^2*T^4 -> R^2=L/(sigm*pi*4*T^4) now g in Si units
% [g]=m/s*100cm/1m -->g_cgs=g_si*100

G=6.67430e-11;    sigm=5.670374e-8; Ls=3.828e26; Msun=1.9885e30; Rs=6.96e8;
% m^3 kg^-1 s^-2; W m^-2 K^-4;      W;           kg;             m;

pc=3.0857e16;

% Threshold for the angular separation
```

```
s_lim=2;

% Values for the metalicities acording to the index
% index =0 thin; =1 thick, =2 halo
Z0=0.02; Z1=0.01; Z2=0.00173;



%% WD+WD systems

% Calculate the logg

loggWD1=log10(100.*G.*Msun.*WDWD(:,1)./(10.^(WDWD(:,5))*Ls./...
    (4.*pi.*sigm.*(10.^(WDWD(:,3))).^4)));
loggWD2=log10(100.*G.*Msun.*WDWD(:,2)./(10.^(WDWD(:,6))*Ls./...
    (4.*pi.*sigm.*(10.^(WDWD(:,4))).^4)));
G_dr3_WD1=zeros(1,length(WDWD(:,1)));
BP_dr3_WD1=zeros(1,length(WDWD(:,1)));
RP_dr3_WD1=zeros(1,length(WDWD(:,1)));

G_dr3_WD2=zeros(1,length(WDWD(:,1)));
BP_dr3_WD2=zeros(1,length(WDWD(:,1)));
RP_dr3_WD2=zeros(1,length(WDWD(:,1)));

% Interpolation to find Gaia magnitudes
for ii=1:length(WDWD(:,1))
    % WD1 component
    if 10^(WDWD(ii,3))≤TeffWD(84)
        G_dr3_WD1(ii)=interp2(TeffWD(1:84),loggsWD,G_dr3,...
            10^(WDWD(ii,3)),loggWD1(ii),'makima');
        BP_dr3_WD1(ii)=interp2(TeffWD(1:84),loggsWD,BP_dr3,...
            10^(WDWD(ii,3)),loggWD1(ii),'makima');
        RP_dr3_WD1(ii)=interp2(TeffWD(1:84),loggsWD,RP_dr3,...
            10^(WDWD(ii,3)),loggWD1(ii),'makima');
    else
        G_dr3_WD1(ii)=interp2(TeffWD(85:end),loggsWD(5:end),G_dr3_2,...
            10^(WDWD(ii,3)),loggWD1(ii),'makima');
        BP_dr3_WD1(ii)=interp2(TeffWD(85:end),loggsWD(5:end),BP_dr3_2,...
            10^(WDWD(ii,3)),loggWD1(ii),'makima');
        RP_dr3_WD1(ii)=interp2(TeffWD(85:end),loggsWD(5:end),RP_dr3_2,...
            10^(WDWD(ii,3)),loggWD1(ii),'makima');
    end
    %WD2 component
    if 10^(WDWD(ii,4))≤TeffWD(84)
        G_dr3_WD2(ii)=interp2(TeffWD(1:84),loggsWD,G_dr3,...
            10^(WDWD(ii,4)),loggWD2(ii),'makima');
        BP_dr3_WD2(ii)=interp2(TeffWD(1:84),loggsWD,BP_dr3,...
            10^(WDWD(ii,4)),loggWD2(ii),'makima');
        RP_dr3_WD2(ii)=interp2(TeffWD(1:84),loggsWD,RP_dr3,...
            10^(WDWD(ii,4)),loggWD2(ii),'makima');
    else
        G_dr3_WD2(ii)=interp2(TeffWD(85:end),loggsWD(5:end),G_dr3_2,...
            10^(WDWD(ii,4)),loggWD2(ii),'makima');
        BP_dr3_WD2(ii)=interp2(TeffWD(85:end),loggsWD(5:end),BP_dr3_2,...
            10^(WDWD(ii,4)),loggWD2(ii),'makima');
        RP_dr3_WD2(ii)=interp2(TeffWD(85:end),loggsWD(5:end),RP_dr3_2,...
            10^(WDWD(ii,4)),loggWD2(ii),'makima');
    end
end

% Calculate the absolute magnitudes
G_abs_WD1=G_dr3_WD1-2.5.*log10((10.^(WDWD(:,5))*Ls./(4.*pi.*sigm.*...
    (10.^(WDWD(:,3))).^4))/(10*pc)^2)';
BP_abs_WD1=BP_dr3_WD1-2.5.*log10((10.^(WDWD(:,5))*Ls./(4.*pi.*sigm.*...
    (10.^(WDWD(:,3))).^4))/(10*pc)^2)';
RP_abs_WD1=RP_dr3_WD1-2.5.*log10((10.^(WDWD(:,5))*Ls./(4.*pi.*sigm.*...
    (10.^(WDWD(:,3))).^4))/(10*pc)^2)';

G_abs_WD2=G_dr3_WD2-2.5.*log10((10.^(WDWD(:,6))*Ls./(4.*pi.*sigm.*...
    (10.^(WDWD(:,4))).^4))/(10*pc)^2)';
BP_abs_WD2=BP_dr3_WD2-2.5.*log10((10.^(WDWD(:,6))*Ls./(4.*pi.*sigm.*...
    (10.^(WDWD(:,4))).^4))/(10*pc)^2)';
RP_abs_WD2=RP_dr3_WD2-2.5.*log10((10.^(WDWD(:,6))*Ls./(4.*pi.*sigm.*...
    (10.^(WDWD(:,4))).^4))/(10*pc)^2)';

% Sort WD1 and WD2 according to MG_{WD1}<MG_{WD2}
G_abs_WD1_aux=G_abs_WD1; BP_abs_WD1_aux=BP_abs_WD1;
RP_abs_WD1_aux=RP_abs_WD1;
G_abs_WD2_aux=G_abs_WD2; BP_abs_WD2_aux=BP_abs_WD2;
RP_abs_WD2_aux=RP_abs_WD2;
WDWD_aux=WDWD;

for ii=1:WDWD(:,1)
    % If it is not fulfilled change WD1 and WD2
    if G_abs_WD1_aux(ii)>G_abs_WD2_aux(ii)
       G_abs_WD1(ii)=G_abs_WD2_aux(ii); BP_abs_WD1(ii)=BP_abs_WD2_aux(ii);
       RP_abs_WD1(ii)=RP_abs_WD2_aux(ii);
       G_abs_WD2(ii)=G_abs_WD1_aux(ii); BP_abs_WD2(ii)=BP_abs_WD1_aux(ii);
       RP_abs_WD2(ii)=RP_abs_WD1_aux(ii);
       WDWD(ii,1)=WDWD_aux(ii,2); WDWD(ii,2)=WDWD_aux(ii,1);
       WDWD(ii,3)=WDWD_aux(ii,4); WDWD(ii,4)=WDWD_aux(ii,3);
       WDWD(ii,5)=WDWD_aux(ii,6); WDWD(ii,6)=WDWD_aux(ii,5);
    end
end


G_WDWD=[]; BP_WDWD=[]; RP_WDWD=[]; Teff_WDWD=[]; logg_WDWD=[]; par_WDWD=[];
index_WDWD=[]; L_WDWD=[]; M_WDWD=[];
G_WD1=[];  BP_WD1=[];  RP_WD1=[];  Teff_WD1=[];  logg_WD1=[];  par_WD1=[];
index_WD1=[];  L_WD1=[];  M_WD1=[];
G_WD2=[];  BP_WD2=[];  RP_WD2=[];  Teff_WD2=[];  logg_WD2=[];  par_WD2=[];
index_WD2=[];  L_WD2=[];  M_WD2=[];

% Check if the system is resolved or not and save variables
for ii=1:length(WDWD(:,1))
    if WDWD(ii,7)<s_lim
        G_WDWD=[G_WDWD -2.5*log10(10^(-G_abs_WD1(ii)*0.4)+10^...
            (-G_abs_WD2(ii)*0.4))];
        BP_WDWD=[BP_WDWD -2.5*log10(10^(-BP_abs_WD1(ii)*0.4)+10^...
            (-BP_abs_WD2(ii)*0.4))];
        RP_WDWD=[RP_WDWD -2.5*log10(10^(-RP_abs_WD1(ii)*0.4)+10^...
            (-RP_abs_WD2(ii)*0.4))];
        Teff_WDWD=[Teff_WDWD; 10^WDWD(ii,3),10^WDWD(ii,4)];
        logg_WDWD=[logg_WDWD; loggWD1(ii), loggWD2(ii)];
        L_WDWD=[L_WDWD; WDWD(ii,5),WDWD(ii,6)];
        M_WDWD=[M_WDWD; WDWD(ii,1),WDWD(ii,2)];
        par_WDWD=[par_WDWD WDWD(ii,8)];
        index_WDWD=[index_WDWD WDWD(ii,9)];
    else
        G_WD1=[G_WD1 G_abs_WD1(ii)]; BP_WD1=[BP_WD1 BP_abs_WD1(ii)];
        RP_WD1=[RP_WD1 RP_abs_WD1(ii)];
        G_WD2=[G_WD2 G_abs_WD2(ii)]; BP_WD2=[BP_WD2 BP_abs_WD2(ii)];
        RP_WD2=[RP_WD2 RP_abs_WD2(ii)];
        Teff_WD1=[Teff_WD1 10^WDWD(ii,3)];
        Teff_WD2=[Teff_WD2 10^WDWD(ii,4)];
        logg_WD1=[logg_WD1 loggWD1(ii)];
        logg_WD2=[logg_WD2 loggWD2(ii)];
        par_WD1=[par_WD1 WDWD(ii,8)];
        par_WD2=[par_WD2 WDWD(ii,8)];
        L_WD1=[L_WD1 WDWD(ii,5)];
        L_WD2=[L_WD2 WDWD(ii,6)];
        M_WD1=[M_WD1 WDWD(ii,1)];
        M_WD2=[M_WD2 WDWD(ii,2)];
        index_WD1=[index_WD1 WDWD(ii,9)];
        index_WD2=[index_WD2 WDWD(ii,9)];
    end
end



%% MS+WD systems

% Calculate the logg

loggWD=log10(100.*G.*Msun.*MSWD(:,1)./(10.^(MSWD(:,5))*Ls./...
    (4.*pi.*sigm.*(10.^(MSWD(:,3))).^4)));
```

```
loggMS=log10(100.*G.*Msun.*MSWD(:,2)./(10.^(MSWD(:,6))*Ls./...
    (4.*pi.*sigm.*(10.^(MSWD(:,4))).^4)));
G_dr3_WD=zeros(1,length(MSWD(:,1))); BP_dr3_WD=zeros(1,length(MSWD(:,1)));
RP_dr3_WD=zeros(1,length(MSWD(:,1)));
G_dr3_MS=zeros(1,length(MSWD(:,1))); BP_dr3_MS=zeros(1,length(MSWD(:,1)));
RP_dr3_MS =zeros(1,length(MSWD(:,1)));

% Interpolation to find Gaia magnitudes
for ii=1:length(MSWD(:,1))
    %WD component
    if 10^(MSWD(ii,3))≤TeffWD(84)
        G_dr3_WD(ii)=interp2(TeffWD(1:84),loggsWD,G_dr3,10^...
            (MSWD(ii,3)),loggWD(ii),'makima');
        BP_dr3_WD(ii)=interp2(TeffWD(1:84),loggsWD,BP_dr3,10^...
            (MSWD(ii,3)),loggWD(ii),'makima');
        RP_dr3_WD(ii)=interp2(TeffWD(1:84),loggsWD,RP_dr3,10^...
            (MSWD(ii,3)),loggWD(ii),'makima');
    else
        G_dr3_WD(ii)=interp2(TeffWD(85:end),loggsWD(5:end),G_dr3_2,10^...
            (MSWD(ii,3)),loggWD(ii),'makima');
        BP_dr3_WD(ii)=interp2(TeffWD(85:end),loggsWD(5:end),BP_dr3_2,10^...
            (MSWD(ii,3)),loggWD(ii),'makima');
        RP_dr3_WD(ii)=interp2(TeffWD(85:end),loggsWD(5:end),RP_dr3_2,10^...
            (MSWD(ii,3)),loggWD(ii),'makima');
    end
    % MS component
    % Metallicity dispersion
    if MS(ii,6)==0
        Z=abs(Z0+normrnd(0,0.005));
    elseif MS(ii,6)==1
        Z=abs(Z1+normrnd(0,0.003));
    elseif MS(ii,6)==2
        Z=abs(Z2+normrnd(0,0.001));
    end
    if Z≤0.005                % 0.001/0.005
        G_dr3_MS(ii)=interp1([0.001 0.005],[interp1(P01,G01,MSWD(ii,6),...
            'linear','extrap') interp1(P05,G05,MSWD(ii,6),...
            'linear','extrap')],Z,'linear','extrap');
        BP_dr3_MS(ii)=interp1([0.001 0.005],[interp1(P01,BP01,MSWD(ii,6),...
            'linear','extrap') interp1(P05,BP05,MSWD(ii,6),...
            'linear','extrap')],Z,'linear','extrap');
        RP_dr3_MS(ii)=interp1([0.001 0.005],[interp1(P01,RP01,MSWD(ii,6),...
            'linear','extrap') interp1(P05,RP05,MSWD(ii,6),...
            'linear','extrap')],Z,'linear','extrap');
    elseif Z>0.005 && Z≤0.010 % 0.005/0.010
        G_dr3_MS(ii)=interp1([0.01 0.005],[interp1(P10,G10,MSWD(ii,6),...
            'linear','extrap') interp1(P05,G05,MSWD(ii,6),...
            'linear','extrap')],Z,'linear','extrap');
        BP_dr3_MS(ii)=interp1([0.01 0.005],[interp1(P10,BP10,MSWD(ii,6),...
            'linear','extrap') interp1(P05,BP05,MSWD(ii,6),...
            'linear','extrap')],Z,'linear','extrap');
        RP_dr3_MS(ii)=interp1([0.01 0.005],[interp1(P10,RP10,MSWD(ii,6),...
            'linear','extrap') interp1(P05,RP05,MSWD(ii,6),...
            'linear','extrap')],Z,'linear','extrap');
    elseif Z>0.010 && Z≤0.015 % 0.010/0.015
        G_dr3_MS(ii)=interp1([0.01 0.015],[interp1(P10,G10,MSWD(ii,6),...
            'linear','extrap') interp1(P15,G15,MSWD(ii,6),...
            'linear','extrap')],Z,'linear','extrap');
        BP_dr3_MS(ii)=interp1([0.01 0.015],[interp1(P10,BP10,MSWD(ii,6),...
            'linear','extrap') interp1(P15,BP15,MSWD(ii,6),...
            'linear','extrap')],Z,'linear','extrap');
        RP_dr3_MS(ii)=interp1([0.01 0.015],[interp1(P10,RP10,MSWD(ii,6),...
            'linear','extrap') interp1(P15,RP15,MSWD(ii,6),...
            'linear','extrap')],Z,'linear','extrap');
    elseif Z>0.015 && Z≤0.020 % 0.015/0.020
        G_dr3_MS(ii)=interp1([0.02 0.015],[interp1(P20,G20,MSWD(ii,6),...
            'linear','extrap') interp1(P15,G15,MSWD(ii,6),...
            'linear','extrap')],Z,'linear','extrap');
        BP_dr3_MS(ii)=interp1([0.02 0.015],[interp1(P20,BP20,MSWD(ii,6),...
            'linear','extrap') interp1(P15,BP15,MSWD(ii,6),...
            'linear','extrap')],Z,'linear','extrap');
        RP_dr3_MS(ii)=interp1([0.02 0.015],[interp1(P20,RP20,MSWD(ii,6),...
            'linear','extrap') interp1(P15,RP15,MSWD(ii,6),...
            'linear','extrap')],Z,'linear','extrap');
    elseif Z>0.020 && Z≤0.025 % 0.020/0.025
        G_dr3_MS(ii)=interp1([0.02 0.025],[interp1(P20,G20,MSWD(ii,6),...
            'linear','extrap') interp1(P25,G25,MSWD(ii,6),...
            'linear','extrap')],Z,'linear','extrap');
        BP_dr3_MS(ii)=interp1([0.02 0.025],[interp1(P20,BP20,MSWD(ii,6),...
            'linear','extrap') interp1(P25,BP25,MSWD(ii,6),...
            'linear','extrap')],Z,'linear','extrap');
        RP_dr3_MS(ii)=interp1([0.02 0.025],[interp1(P20,RP20,MSWD(ii,6),...
            'linear','extrap') interp1(P25,RP25,MSWD(ii,6),...
            'linear','extrap')],Z,'linear','extrap');
    elseif Z>0.025 && Z≤0.030 % 0.025/0.030
        G_dr3_MS(ii)=interp1([0.03 0.025],[interp1(P30,G30,MSWD(ii,6),...
            'linear','extrap') interp1(P35,G35,MSWD(ii,6),...
            'linear','extrap')],Z,'linear','extrap');
        BP_dr3_MS(ii)=interp1([0.03 0.025],[interp1(P30,BP30,MSWD(ii,6),...
            'linear','extrap') interp1(P35,BP35,MSWD(ii,6),...
            'linear','extrap')],Z,'linear','extrap');
        RP_dr3_MS(ii)=interp1([0.03 0.025],[interp1(P30,RP30,MSWD(ii,6),...
            'linear','extrap') interp1(P35,RP35,MSWD(ii,6),...
            'linear','extrap')],Z,'linear','extrap');
    elseif Z>0.030 && Z≤0.035 % 0.030/0.035
        G_dr3_MS(ii)=interp1([0.03 0.035],[interp1(P30,G30,MSWD(ii,6),...
            'linear','extrap') interp1(P35,G35,MSWD(ii,6),...
            'linear','extrap')],Z,'linear','extrap');
        BP_dr3_MS(ii)=interp1([0.03 0.035],[interp1(P30,BP30,MSWD(ii,6),...
            'linear','extrap') interp1(P35,BP35,MSWD(ii,6),...
            'linear','extrap')],Z,'linear','extrap');
        RP_dr3_MS(ii)=interp1([0.03 0.035],[interp1(P30,RP30,MSWD(ii,6),...
            'linear','extrap') interp1(P35,RP35,MSWD(ii,6),...
            'linear','extrap')],Z,'linear','extrap');
    elseif Z>0.035 && Z≤0.040 % 0.035/0.040
        G_dr3_MS(ii)=interp1([0.04 0.035],[interp1(P40,G40,MSWD(ii,6),...
            'linear','extrap') interp1(P35,G35,MSWD(ii,6),...
            'linear','extrap')],Z,'linear','extrap');
        BP_dr3_MS(ii)=interp1([0.04 0.035],[interp1(P40,BP40,MSWD(ii,6),...
            'linear','extrap') interp1(P35,BP35,MSWD(ii,6),...
            'linear','extrap')],Z,'linear','extrap');
        RP_dr3_MS(ii)=interp1([0.04 0.035],[interp1(P40,RP40,MSWD(ii,6),...
            'linear','extrap') interp1(P35,RP35,MSWD(ii,6),...
            'linear','extrap')],Z,'linear','extrap');
    elseif Z>0.040             % 0.040/0.045
        G_dr3_MS(ii)=interp1([0.04 0.045],[interp1(P40,G40,MSWD(ii,6),...
            'linear','extrap') interp1(P45,G45,MSWD(ii,6),...
            'linear','extrap')],Z,'linear','extrap');
        BP_dr3_MS(ii)=interp1([0.04 0.045],[interp1(P40,BP40,MSWD(ii,6),...
            'linear','extrap') interp1(P45,BP45,MSWD(ii,6),...
            'linear','extrap')],Z,'linear','extrap');
        RP_dr3_MS(ii)=interp1([0.04 0.045],[interp1(P40,RP40,MSWD(ii,6),...
            'linear','extrap') interp1(P45,RP45,MSWD(ii,6),...
            'linear','extrap')],Z,'linear','extrap');
    end
end

% Calculate the absolute magnitudes of the WD component
G_abs_WD=G_dr3_WD-2.5.*log10((10.^(MSWD(:,5))*Ls./(4.*pi.*sigm.*...
    (10.^(MSWD(:,3))).^4))/(10*pc)^2)';
BP_abs_WD=BP_dr3_WD-2.5.*log10((10.^(MSWD(:,5))*Ls./(4.*pi.*sigm.*...
    (10.^(MSWD(:,3))).^4))/(10*pc)^2)';
RP_abs_WD=RP_dr3_WD-2.5.*log10((10.^(MSWD(:,5))*Ls./(4.*pi.*sigm.*...
    (10.^(MSWD(:,3))).^4))/(10*pc)^2)';

% Rename the absolute magnitudes of the MS component
G_abs_MS=G_dr3_MS';
BP_abs_MS=BP_dr3_MS';
RP_abs_MS=RP_dr3_MS';

G_MSWD=[];      BP_MSWD=[];     RP_MSWD=[]; Teff_MSWD=[]; logg_MSWD=[];
index_MSWD=[];  par_MSWD=[];    L_MSWD=[];  M_MSWD=[];
G_MS=[];        BP_MS=[];       RP_MS=[];   Teff_MS=[];   logg_MS=[];
index_MS=[];    par_MS=[];      L_MS=[];    M_MS=[];
G_WD=[];        BP_WD=[];       RP_WD=[];   Teff_WD=[];   logg_WD=[];
index_WD=[];    par_WD=[];      L_WD=[];    M_WD=[];
```

```

G_MSWD_MS_WD=[]; BP_MSWD_MS_WD=[]; RP_MSWD_MS_WD=[];

% Check if the system is resolved or not and save variables
for ii=1:length(MSWD(:,1))
    if MSWD(ii,7)<s_lim
        G_MSWD=[G_MSWD -2.5*log10(10^(-G_abs_MS(ii)*0.4)+10^...
            (-G_abs_WD(ii)*0.4))];
        BP_MSWD=[BP_MSWD -2.5*log10(10^(-BP_abs_MS(ii)*0.4)+10^...
            (-BP_abs_WD(ii)*0.4))];
        RP_MSWD=[RP_MSWD -2.5*log10(10^(-RP_abs_MS(ii)*0.4)+10^...
            (-RP_abs_WD(ii)*0.4))];
        G_MSWD_MS_WD=[G_MSWD_MS_WD; G_abs_MS(ii) G_abs_WD(ii)];
        BP_MSWD_MS_WD=[BP_MSWD_MS_WD; BP_abs_MS(ii) BP_abs_WD(ii)];
        RP_MSWD_MS_WD=[RP_MSWD_MS_WD;RP_abs_MS(ii) RP_abs_WD(ii)];
        Teff_MSWD=[Teff_MSWD; 10^MSWD(ii,4),10^MSWD(ii,3)];
        logg_MSWD=[logg_MSWD; loggMS(ii), loggWD(ii)];
        L_MSWD=[L_MSWD; MSWD(ii,6), MSWD(ii,5)];
        M_MSWD=[M_MSWD; MSWD(ii,2), MSWD(ii,1)];
        index_MSWD=[index_MSWD MSWD(ii,9)];
        par_MSWD=[par_MSWD MSWD(ii,8)];
    else
        G_MS=[G_MS G_abs_MS(ii)]; BP_MS=[BP_MS BP_abs_MS(ii)];
        RP_MS=[RP_MS RP_abs_MS(ii)];
        G_WD=[G_WD G_abs_WD(ii)]; BP_WD=[BP_WD BP_abs_WD(ii)];
        RP_WD=[RP_WD RP_abs_WD(ii)];
        Teff_MS=[Teff_MS 10^MSWD(ii,4)]; Teff_WD=[Teff_WD 10^MSWD(ii,3)];
        index_MS=[index_MS MSWD(ii,9)];  index_WD=[index_WD MSWD(ii,9)];
        par_MS=[par_MS MSWD(ii,8)];      par_WD=[par_WD MSWD(ii,8)];
        L_MS=[L_MS MSWD(ii,6)];          L_WD=[L_WD MSWD(ii,5)];
        M_MS=[M_MS MSWD(ii,2)];          M_WD=[M_WD MSWD(ii,1)];
        logg_MS=[logg_MS loggMS(ii)];    logg_WD=[logg_WD loggWD(ii)];
    end
end



%% WD mergers

% Calculate the logg

loggWDm=log10(100.*G.*Msun.*WDmergers(:,1)./(10.^(WDmergers(:,3))*Ls./...
    (4.*pi.*sigm.*(10.^(WDmergers(:,2))).^4)));
G_dr3_WDm=zeros(1,length(WDmergers(:,1)));
BP_dr3_WD1=zeros(1,length(WDmergers(:,1)));
RP_dr3_WD1=zeros(1,length(WDmergers(:,1)));

% Interpolation to find Gaia magnitudes
for ii=1:length(WDmergers(:,1))
    if 10^(WDmergers(ii,2))≤TeffWD(84)
        G_dr3_WDm(ii)=interp2(TeffWD(1:84),loggsWD,G_dr3,10^...
            (WDmergers(ii,2)),loggWDm(ii),'makima');
        BP_dr3_WDm(ii)=interp2(TeffWD(1:84),loggsWD,BP_dr3,10^...
            (WDmergers(ii,2)),loggWDm(ii),'makima');
        RP_dr3_WDm(ii)=interp2(TeffWD(1:84),loggsWD,RP_dr3,10^...
            (WDmergers(ii,2)),loggWDm(ii),'makima');
    else
        G_dr3_WDm(ii)=interp2(TeffWD(85:end),loggsWD(5:end),G_dr3_2,...
            10^(WDmergers(ii,2)),loggWDm(ii),'makima');
        BP_dr3_WDm(ii)=interp2(TeffWD(85:end),loggsWD(5:end),BP_dr3_2,...
            10^(WDmergers(ii,2)),loggWDm(ii),'makima');
        RP_dr3_WDm(ii)=interp2(TeffWD(85:end),loggsWD(5:end),RP_dr3_2,...
            10^(WDmergers(ii,2)),loggWDm(ii),'makima');
    end
end

% Calculate the absolute magnitudes
G_WDm=G_dr3_WDm-2.5.*log10((10.^(WDmergers(:,3))*Ls./(4.*pi.*sigm.*...
    (10.^(WDmergers(:,2))).^4))/(10*pc)^2)';
BP_WDm=BP_dr3_WDm-2.5.*log10((10.^(WDmergers(:,3))*Ls./(4.*pi.*sigm.*...
    (10.^(WDmergers(:,2))).^4))/(10*pc)^2)';
RP_WDm=RP_dr3_WDm-2.5.*log10((10.^(WDmergers(:,3))*Ls./(4.*pi.*sigm.*...
    (10.^(WDmergers(:,2))).^4))/(10*pc)^2)';
```

```

% Save variables
Teff_WDm=10.^WDmergers(:,2);
logg_WDm=loggWDm;
par_WDm=WDmergers(:,4);
index_WDm=WDmergers(:,5);
L_WDm=WDmergers(:,3);
M_WDm=WDmergers(:,1);



%% WD singles

% Calculate the absolute magnitudes

par_WDs=WDsingles(:,8)/1000;
G_abs_WDs=WDsingles(:,5)+5+5*log10(par_WDs);
BP_abs_WDs=WDsingles(:,6)+5+5*log10(par_WDs);
RP_abs_WDs=WDsingles(:,7)+5+5*log10(par_WDs);

% Save variables
Teff_WDs=10.^WDsingles(:,2);
logg_WDs=WDsingles(:,3);
index_WDs=WDsingles(:,9);
L_WDs=log10((4.*pi.*sigm.*(WDsingles(:,4).*Rs).^2.*(10.^WDsingles(:,2)).^4)/Ls);
M_WDs=WDsingles(:,1);



%% MS singles

% Calculate the logg

loggMSs=log10(100.*G.*Msun.*MS(:,1)./(10.^(MS(:,3)).*Rs).^2);
G_MSs=zeros(1,length(MS(:,1))); BP_MSs=zeros(1,length(MS(:,1)));
RP_MSs=zeros(1,length(MS(:,1)));

% Interpolation to find gaia mag
for ii=1:length(MS(:,1))
    % Metallicity disperssion
    if MS(ii,6)==0
        Z=abs(Z0+normrnd(0,0.005));
    elseif MS(ii,6)==1
        Z=abs(Z1+normrnd(0,0.003));
    elseif MS(ii,6)==2
        Z=abs(Z2+normrnd(0,0.001));
    end
    if Z≤0.005                 % 0.001/0.005
        G_MSs(ii)=interp1([0.001 0.005],[interp1(P01,G01,MS(ii,4),...
            'linear','extrap') interp1(P05,G05,MS(ii,4),...
            'linear','extrap')],Z,'linear','extrap');
        BP_MSs(ii)=interp1([0.001 0.005],[interp1(P01,BP01,MS(ii,4),...
            'linear','extrap') interp1(P05,BP05,MS(ii,4),...
            'linear','extrap')],Z,'linear','extrap');
        RP_MSs(ii)=interp1([0.001 0.005],[interp1(P01,RP01,MS(ii,4),...
            'linear','extrap') interp1(P05,RP05,MS(ii,4),...
            'linear','extrap')],Z,'linear','extrap');
    elseif Z>0.005 && Z≤0.010 % 0.005/0.010
        G_MSs(ii)=interp1([0.01 0.005],[interp1(P10,G10,MS(ii,4),...
            'linear','extrap') interp1(P05,G05,MS(ii,4),...
            'linear','extrap')],Z,'linear','extrap');
        BP_MSs(ii)=interp1([0.01 0.005],[interp1(P10,BP10,MS(ii,4),...
            'linear','extrap') interp1(P05,BP05,MS(ii,4),...
            'linear','extrap')],Z,'linear','extrap');
        RP_MSs(ii)=interp1([0.01 0.005],[interp1(P10,RP10,MS(ii,4),...
            'linear','extrap') interp1(P05,RP05,MS(ii,4),...
            'linear','extrap')],Z,'linear','extrap');
    elseif Z>0.010 && Z≤0.015 % 0.010/0.015
        G_MSs(ii)=interp1([0.01 0.015],[interp1(P10,G10,MS(ii,4),...
            'linear','extrap') interp1(P15,G15,MS(ii,4),...
            'linear','extrap')],Z,'linear','extrap');
        BP_MSs(ii)=interp1([0.01 0.015],[interp1(P10,BP10,MS(ii,4),...
            'linear','extrap') interp1(P15,BP15,MS(ii,4),...
            'linear','extrap')],Z,'linear','extrap');
        RP_MSs(ii)=interp1([0.01 0.015],[interp1(P10,RP10,MS(ii,4),...
            'linear','extrap') interp1(P15,RP15,MS(ii,4),...
            'linear','extrap')],Z,'linear','extrap');
    elseif Z>0.015 && Z≤0.020 % 0.015/0.020
        G_MSs(ii)=interp1([0.02 0.015],[interp1(P20,G20,MS(ii,4),...
            'linear','extrap') interp1(P15,G15,MS(ii,4),...
            'linear','extrap')],Z,'linear','extrap');
        BP_MSs(ii)=interp1([0.02 0.015],[interp1(P20,BP20,MS(ii,4),...
            'linear','extrap') interp1(P15,BP15,MS(ii,4),...
            'linear','extrap')],Z,'linear','extrap');
        RP_MSs(ii)=interp1([0.02 0.015],[interp1(P20,RP20,MS(ii,4),...
            'linear','extrap') interp1(P15,RP15,MS(ii,4),...
            'linear','extrap')],Z,'linear','extrap');
    elseif Z>0.020 && Z≤0.025 % 0.020/0.025
        G_MSs(ii)=interp1([0.02 0.025],[interp1(P20,G20,MS(ii,4),...
            'linear','extrap') interp1(P25,G25,MS(ii,4),...
            'linear','extrap')],Z,'linear','extrap');
        BP_MSs(ii)=interp1([0.02 0.025],[interp1(P20,BP20,MS(ii,4),...
            'linear','extrap') interp1(P25,BP25,MS(ii,4),...
            'linear','extrap')],Z,'linear','extrap');
        RP_MSs(ii)=interp1([0.02 0.025],[interp1(P20,RP20,MS(ii,4),...
            'linear','extrap') interp1(P25,RP25,MS(ii,4),...
            'linear','extrap')],Z,'linear','extrap');
    elseif Z>0.025 && Z≤0.030 % 0.025/0.030
        G_MSs(ii)=interp1([0.03 0.025],[interp1(P30,G30,MS(ii,4),...
            'linear','extrap') interp1(P35,G35,MS(ii,4),...
            'linear','extrap')],Z,'linear','extrap');
        BP_MSs(ii)=interp1([0.03 0.025],[interp1(P30,BP30,MS(ii,4),...
            'linear','extrap') interp1(P35,BP35,MS(ii,4),...
            'linear','extrap')],Z,'linear','extrap');
        RP_MSs(ii)=interp1([0.03 0.025],[interp1(P30,RP30,MS(ii,4),...
            'linear','extrap') interp1(P35,RP35,MS(ii,4),...
            'linear','extrap')],Z,'linear','extrap');
    elseif Z>0.030 && Z≤0.035 % 0.030/0.035
        G_MSs(ii)=interp1([0.03 0.035],[interp1(P30,G30,MS(ii,4),...
            'linear','extrap') interp1(P35,G35,MS(ii,4),...
            'linear','extrap')],Z,'linear','extrap');
        BP_MSs(ii)=interp1([0.03 0.035],[interp1(P30,BP30,MS(ii,4),...
            'linear','extrap') interp1(P35,BP35,MS(ii,4),...
            'linear','extrap')],Z,'linear','extrap');
        RP_MSs(ii)=interp1([0.03 0.035],[interp1(P30,RP30,MS(ii,4),...
            'linear','extrap') interp1(P35,RP35,MS(ii,4),...
            'linear','extrap')],Z,'linear','extrap');
    elseif Z>0.035 && Z≤0.040 % 0.035/0.040
        G_MSs(ii)=interp1([0.04 0.035],[interp1(P40,G40,MS(ii,4),...
            'linear','extrap') interp1(P35,G35,MS(ii,4),...
            'linear','extrap')],Z,'linear','extrap');
        BP_MSs(ii)=interp1([0.04 0.035],[interp1(P40,BP40,MS(ii,4),...
            'linear','extrap') interp1(P35,BP35,MS(ii,4),...
            'linear','extrap')],Z,'linear','extrap');
        RP_MSs(ii)=interp1([0.04 0.035],[interp1(P40,RP40,MS(ii,4),...
            'linear','extrap') interp1(P35,RP35,MS(ii,4),...
            'linear','extrap')],Z,'linear','extrap');
    elseif Z>0.040             % 0.040/0.045
        G_MSs(ii)=interp1([0.04 0.045],[interp1(P40,G40,MS(ii,4),...
            'linear','extrap') interp1(P45,G45,MS(ii,4),...
            'linear','extrap')],Z,'linear','extrap');
        BP_MSs(ii)=interp1([0.04 0.045],[interp1(P40,BP40,MS(ii,4),...
            'linear','extrap') interp1(P45,BP45,MS(ii,4),...
            'linear','extrap')],Z,'linear','extrap');
        RP_MSs(ii)=interp1([0.04 0.045],[interp1(P40,RP40,MS(ii,4),...
            'linear','extrap') interp1(P45,RP45,MS(ii,4),...
            'linear','extrap')],Z,'linear','extrap');
    end
end

% Rename variables
G_abs_MSs=G_MSs';
BP_abs_MSs=BP_MSs';
RP_abs_MSs=RP_MSs';

% Save variables
```

```
Teff_MSs=10.^MS(:,2); logg_MSs=loggMSs;
M_MSs=MS(:,1);        index_MSs=MS(:,6); par_MSs=MS(:,5);
L_MSs=MS(:,4);        M_MSs=MS(:,1);



%% Add error

% See functions below

[MG_WDm_e, MBP_WDm_e, MRP_WDm_e, par_WDm_e,id_WDm_e] = ...
    p_error(par_WDm', G_WDm,     BP_WDm,     RP_WDm);
[MG_MSWD_e,MBP_MSWD_e,MRP_MSWD_e, par_MSWD_e, id_MSWD_e]= ...
    p_error(par_MSWD, G_MSWD,    BP_MSWD,    RP_MSWD);
[MG_WDWD_e,MBP_WDWD_e,MRP_WDWD_e, par_WDWD_e, id_WDWD_e]= ...
    p_error(par_WDWD, G_WDWD,    BP_WDWD,    RP_WDWD);
[MG_MSs_e, MBP_MSs_e, MRP_MSs_e, par_MSs_e, id_MSs_e] = ...
    p_error(par_MSs,  G_abs_MSs, BP_abs_MSs, RP_abs_MSs);

[MG_MS_e,MBP_MS_e,MRP_MS_e,MG_WD_e,MBP_WD_e,MRP_WD_e,par_MS_e,par_WD_e,...
    id_MS_WD_e]=p_error2(par_MS,par_WD,G_MS,BP_MS,RP_MS,G_WD,BP_WD,RP_WD);
[MG_WD1_e,MBP_WD1_e,MRP_WD1_e,MG_WD2_e,MBP_WD2_e,MRP_WD2_e,par_WD1_e,...
    par_WD2_e,id_WD12_e]=p_error2(par_WD1,par_WD2,G_WD1,BP_WD1,RP_WD1,...
    G_WD2,BP_WD2,RP_WD2);



%% Eliminate stars with G>20.7

Teff_WDm_e=Teff_WDm;        Teff_WDm_e(id_WDm_e)=[];
logg_WDm_e=logg_WDm;        logg_WDm_e(id_WDm_e)=[];
index_WDm_e=index_WDm;      index_WDm_e(id_WDm_e)=[];
L_WDm_e=L_WDm;              L_WDm_e(id_WDm_e)=[];
M_WDm_e=M_WDm;              M_WDm_e(id_WDm_e)=[];
Teff_MSWD_e=Teff_MSWD;      Teff_MSWD_e(id_MSWD_e,:)=[];
logg_MSWD_e=logg_MSWD;      logg_MSWD_e(id_MSWD_e,:)=[];
index_MSWD_e=index_MSWD;    index_MSWD_e(id_MSWD_e)=[];
L_MSWD_e=L_MSWD;            L_MSWD_e(id_MSWD_e,:)=[];
M_MSWD_e=M_MSWD;            M_MSWD_e(id_MSWD_e,:)=[];
Teff_WDWD_e=Teff_WDWD;      Teff_WDWD_e(id_WDWD_e,:)=[];
logg_WDWD_e=logg_WDWD;      logg_WDWD_e(id_WDWD_e,:)=[];
index_WDWD_e=index_WDWD;    index_WDWD_e(id_WDWD_e)=[];
L_WDWD_e=L_WDWD;            L_WDWD_e(id_WDWD_e,:)=[];
M_WDWD_e=M_WDWD;            M_WDWD_e(id_WDWD_e,:)=[];
Teff_MSs_e=Teff_MSs;        Teff_MSs_e(id_MSs_e)=[];
logg_MSs_e=logg_MSs;        logg_MSs_e(id_MSs_e)=[];
index_MSs_e=index_MSs;      index_MSs_e(id_MSs_e)=[];
L_MSs_e=L_MSs;              L_MSs_e(id_MSs_e)=[];
M_MSs_e=M_MSs;              M_MSs_e(id_MSs_e)=[];
Teff_MS_e=Teff_MS;          Teff_MS_e(id_MS_WD_e)=[];
logg_MS_e=logg_MS;          logg_MS_e(id_MS_WD_e)=[];
index_MS_e=index_MS;        index_MS_e(id_MS_WD_e)=[];
L_MS_e=L_MS;                L_MS_e(id_MS_WD_e)=[];
M_MS_e=M_MS;                M_MS_e(id_MS_WD_e)=[];
Teff_WD_e=Teff_WD;          Teff_WD_e(id_MS_WD_e)=[];
logg_WD_e=logg_WD;          logg_WD_e(id_MS_WD_e)=[];
index_WD_e=index_WD;        index_WD_e(id_MS_WD_e)=[];
L_WD_e=L_WD;                L_WD_e(id_MS_WD_e)=[];
M_WD_e=M_WD;                M_WD_e(id_MS_WD_e)=[];
Teff_WD1_e=Teff_WD1;        Teff_WD1_e(id_WD12_e)=[];
logg_WD1_e=logg_WD1;        logg_WD1_e(id_WD12_e)=[];
index_WD1_e=index_WD1;      index_WD1_e(id_WD12_e)=[];
L_WD1_e=L_WD1;              L_WD1_e(id_WD12_e)=[];
M_WD1_e=M_WD1;              M_WD1_e(id_WD12_e)=[];
Teff_WD2_e=Teff_WD2;        Teff_WD2_e(id_WD12_e)=[];
logg_WD2_e=logg_WD2;        logg_WD2_e(id_WD12_e)=[];
index_WD2_e=index_WD2;      index_WD2_e(id_WD12_e)=[];
L_WD2_e=L_WD2;              L_WD2_e(id_WD12_e)=[];
M_WD2_e=M_WD2;              M_WD2_e(id_WD12_e)=[];


%% Add error to Teff, logg and M

```

```
Teff_WDs_ee=Teff_WDs+normrnd(0,0.05*Teff_WDs);
logg_WDs_ee=logg_WDs+normrnd(0,0.05*logg_WDs);
M_WDs_ee=M_WDs+normrnd(0,0.05*M_WDs);
Teff_WDm_ee=Teff_WDm_e+normrnd(0,0.05*Teff_WDm_e);
logg_WDm_ee=logg_WDm_e+normrnd(0,0.05*logg_WDm_e);
M_WDm_ee=M_WDm_e+normrnd(0,0.05*M_WDm_e);
Teff_MSWD_ee=Teff_MSWD_e+normrnd(0,0.05*Teff_MSWD_e);
logg_MSWD_ee=logg_MSWD_e+normrnd(0,0.05*logg_MSWD_e);
M_MSWD_ee=M_MSWD_e+normrnd(0,0.05*M_MSWD_e);
Teff_WDWD_ee=Teff_WDWD_e+normrnd(0,0.05*Teff_WDWD_e);
logg_WDWD_ee=logg_WDWD_e+normrnd(0,0.05*logg_WDWD_e);
M_WDWD_ee=M_WDWD_e+normrnd(0,0.05*M_WDWD_e);
Teff_MSs_ee=Teff_MSs_e+normrnd(0,0.05*Teff_MSs_e);
logg_MSs_ee=logg_MSs_e+normrnd(0,0.05*logg_MSs_e);
M_MSs_ee=M_MSs_e+normrnd(0,0.05*M_MSs_e);
Teff_MS_ee=Teff_MS_e+normrnd(0,0.05*Teff_MS_e);
logg_MS_ee=logg_MS_e+normrnd(0,0.05*logg_MS_e);
M_MS_ee=M_MS_e+normrnd(0,0.05*M_MS_e);
Teff_WD_ee=Teff_WD_e+normrnd(0,0.05*Teff_WD_e);
logg_WD_ee=logg_WD_e+normrnd(0,0.05*logg_WD_e);
M_WD_ee=M_WD_e+normrnd(0,0.05*M_WD_e);
Teff_WD1_ee=Teff_WD1_e+normrnd(0,0.05*Teff_WD1_e);
logg_WD1_ee=logg_WD1_e+normrnd(0,0.05*logg_WD1_e);
M_WD1_ee=M_WD1_e+normrnd(0,0.05*M_WD1_e);
Teff_WD2_ee=Teff_WD2_e+normrnd(0,0.05*Teff_WD2_e);
logg_WD2_ee=logg_WD2_e+normrnd(0,0.05*logg_WD2_e);
M_WD2_ee=M_WD2_e+normrnd(0,0.05*M_WD2_e);

% Calculate error and take the systems that fulfill G<20.7 of the
% unresolved MSWD systems components

G_MSWD_MS_WD_a=G_MSWD_MS_WD; BP_MSWD_MS_WD_a=BP_MSWD_MS_WD;
RP_MSWD_MS_WD_a=RP_MSWD_MS_WD; par_MSWD_a=par_MSWD;
G_MSWD_MS_WD_a(id_MSWD_e,:)=[]; BP_MSWD_MS_WD_a(id_MSWD_e,:)=[];
RP_MSWD_MS_WD_a(id_MSWD_e,:)=[]; par_MSWD_a(id_MSWD_e)=[];

[MG_MSWD_MS_e, MBP_MSWD_MS_e, MRP_MSWD_MS_e] =p_error3(par_MSWD_a, ...
G_MSWD_MS_WD_a(:,1)', BP_MSWD_MS_WD_a(:,1)', RP_MSWD_MS_WD_a(:,1)');
[MG_MSWD_WD_e, MBP_MSWD_WD_e, MRP_MSWD_WD_e] =p_error3(par_MSWD_a, ...
G_MSWD_MS_WD_a(:,2)', BP_MSWD_MS_WD_a(:,2)', RP_MSWD_MS_WD_a(:,2)');
\end{verbatim}


%% Position in the HR diagram

% Decide if the stars is placed in the MS zone, the WD+MS area or the WD
% area
% i=0 => MSWD area
% i=1 => MS area
% i=2 => WD area

iWDs=zeros(length(G_abs_WDs),1);
iWDm=zeros(length(MG_WDm_e),1);
iMSs=zeros(length(MG_MSs_e),1);
iMS=zeros(length(MG_MS_e),1);
iWD=zeros(length(MG_WD_e),1);
iWD1=zeros(length(MG_WD1_e),1);
iWD2=zeros(length(MG_WD2_e),1);
iMSWD=zeros(length(MG_MSWD_e),1);
iWDWD=zeros(length(MG_WDWD_e),1);

% Check if the star is in the area, if not, then decide if it is in the MS
% area or WD area
for ii=1:length(G_abs_WDs)
    res=det_res(G_abs_WDs(ii),BP_abs_WDs(ii),RP_abs_WDs(ii));
    if res==1
        iWDs(ii)=0;
    elseif res==0
        if G_abs_WDs(ii)<2*(BP_abs_WDs(ii)-RP_abs_WDs(ii))+9
            iWDs(ii)=1;
        else
            iWDs(ii)=2;
        end
    end
end
for ii=1:length(MG_WDm_e)
    res=det_res(MG_WDm_e(ii),MBP_WDm_e(ii),MRP_WDm_e(ii));
    if res==1
        iWDm(ii)=0;
    elseif res==0
        if MG_WDm_e(ii)<2*(MBP_WDm_e(ii)-MRP_WDm_e(ii))+9
            iWDm(ii)=1;
        else
            iWDm(ii)=2;
        end
    end
end
for ii=1:length(MG_MSs_e)
    res=det_res(MG_MSs_e(ii),MBP_MSs_e(ii),MRP_MSs_e(ii));
    if res==1
        iMSs(ii)=0;
    elseif res==0
        if MG_MSs_e(ii)<2*(MBP_MSs_e(ii)-MRP_MSs_e(ii))+9
            iMSs(ii)=1;
        else
            iMSs(ii)=2;
        end
    end
end
for ii=1:length(MG_MS_e)
    res=det_res(MG_MS_e(ii),MBP_MS_e(ii),MRP_MS_e(ii));
    if res==1
        iMS(ii)=0;
    elseif res==0
        if MG_MS_e(ii)<2*(MBP_MS_e(ii)-MRP_MS_e(ii))+9
            iMS(ii)=1;
        else
            iMS(ii)=2;
        end
    end
end
for ii=1:length(MG_WD_e)
    res=det_res(MG_WD_e(ii),MBP_WD_e(ii),MRP_WD_e(ii));
    if res==1
        iWD(ii)=0;
    elseif res==0
        if MG_WD_e(ii)<2*(MBP_WD_e(ii)-MRP_WD_e(ii))+9
            iWD(ii)=1;
        else
            iWD(ii)=2;
        end
    end
end
for ii=1:length(MG_WD1_e)
    res=det_res(MG_WD1_e(ii),MBP_WD1_e(ii),MRP_WD1_e(ii));
    if res==1
        iWD1(ii)=0;
    elseif res==0
        if MG_WD1_e(ii)<2*(MBP_WD1_e(ii)-MRP_WD1_e(ii))+9
            iWD1(ii)=1;
        else
            iWD1(ii)=2;
        end
    end
end
for ii=1:length(MG_WD2_e)
    res=det_res(MG_WD2_e(ii),MBP_WD2_e(ii),MRP_WD2_e(ii));
    if res==1
        iWD2(ii)=0;
    elseif res==0
        if MG_WD2_e(ii)<2*(MBP_WD2_e(ii)-MRP_WD2_e(ii))+9
            iWD2(ii)=1;
        else
            iWD2(ii)=2;
        end
    end
```

```
end
for ii=1:length(MG_MSWD_e)
    res=det_res(MG_MSWD_e(ii),MBP_MSWD_e(ii),MRP_MSWD_e(ii));
    if res==1
        iMSWD(ii)=0;
    elseif res==0
        if MG_MSWD_e(ii)<2*(MBP_MSWD_e(ii)-MRP_MSWD_e(ii))+9
            iMSWD(ii)=1;
        else
            iMSWD(ii)=2;
        end
    end
end
for ii=1:length(MG_WDWD_e)
    res=det_res(MG_WDWD_e(ii),MBP_WDWD_e(ii),MRP_WDWD_e(ii));
    if res==1
        iWDWD(ii)=0;
    elseif res==0
        if MG_WDWD_e(ii)<2*(MBP_WDWD_e(ii)-MRP_WDWD_e(ii))+9
            iWDWD(ii)=1;
        else
            iWDWD(ii)=2;
        end
    end
end


%% Reorganize the results

% MG MBP MRP Teff_e logg_e log(L) M_e par location index

WDs_e= [G_abs_WDs  BP_abs_WDs  RP_abs_WDs  Teff_WDs_ee  logg_WDs_ee ...
    L_WDs    M_WDs_ee  par_WDs     iWDs  index_WDs];
WDm_e= [MG_WDm_e'  MBP_WDm_e'  MRP_WDm_e'  Teff_WDm_ee  logg_WDm_ee ...
    L_WDm_e  M_WDm_ee  par_WDm_e'  iWDm  index_WDm_e];
MSs_e= [MG_MSs_e   MBP_MSs_e   MRP_MSs_e   Teff_MSs_ee  logg_MSs_ee ...
    L_MSs_e  M_MSs_ee  par_MSs_e   iMSs  index_MSs_e];
MS_e=  [MG_MS_e'   MBP_MS_e'   MRP_MS_e'   Teff_MS_ee'  logg_MS_ee' ...
    L_MS_e'  M_MS_ee'  par_MS_e'   iMS   index_MS_e'];
WD_e=  [MG_WD_e'   MBP_WD_e'   MRP_WD_e'   Teff_WD_ee'  logg_WD_ee' ...
    L_WD_e'  M_WD_ee'  par_WD_e'   iWD   index_WD_e'];
WD1_e= [MG_WD1_e'  MBP_WD1_e'  MRP_WD1_e'  Teff_WD1_ee' logg_WD1_ee'...
    L_WD1_e' M_WD1_ee' par_WD1_e'  iWD1  index_WD1_e'];
WD2_e= [MG_WD2_e'  MBP_WD2_e'  MRP_WD2_e'  Teff_WD2_ee' logg_WD2_ee'...
    L_WD2_e' M_WD2_ee' par_WD2_e'  iWD2  index_WD2_e'];

% MG MBP MRP Teff_MS_e Teff_WD_e logg_MS_e logg_WD_e
% logL_MS logL_W M_MS_e M_WD_e par location index

MSWD_e=[MG_MSWD_e' MBP_MSWD_e' MRP_MSWD_e' Teff_MSWD_ee logg_MSWD_ee...
    L_MSWD_e M_MSWD_ee par_MSWD_e' iMSWD index_MSWD_e'];

% MG MBP MRP Teff_WD1_e Teff_WD2_e logg_WD1_e logg_WD2_e logL_WD1
% logL_WD2 M_WD1_e M_WD1_e par location index

WDWD_e=[MG_WDWD_e' MBP_WDWD_e' MRP_WDWD_e' Teff_WDWD_ee logg_WDWD_ee...
    L_WDWD_e M_WDWD_ee par_WDWD_e' iWDWD index_WDWD_e'];

% MG MBP MRP
MSWD_MS=[MG_MSWD_MS_e' MBP_MSWD_MS_e' MRP_MSWD_MS_e'];
MSWD_WD=[MG_MSWD_WD_e' MBP_MSWD_WD_e' MRP_MSWD_WD_e'];



%% Save the results

% Save the results in a .txt file

save('WDs_e_fb40.txt','WDs_e','-ASCII');
save('WDm_e_fb40.txt','WDm_e', '-ASCII');
save('MSs_e_fb40.txt','MSs_e', '-ASCII');
save('MS_e_fb40.txt','MS_e', '-ASCII');
save('WD_e_fb40.txt','WD_e', '-ASCII');
save('MSWD_e_fb40.txt','MSWD_e', '-ASCII');
save('WD1_e_fb40.txt','WD1_e', '-ASCII');
save('WD2_e_fb40.txt','WD2_e', '-ASCII');
save('WDWD_e_fb40.txt','WDWD_e', '-ASCII');
save('MSWD_MS_e_fb40.txt','MSWD_MS', '-ASCII');
save('MSWD_WD_e_fb40.txt','MSWD_WD', '-ASCII');




%% Functions

%Returns a 1 if the star is in the MSWD area and a 0 if not

function [res]=det_res(Gabs,bp,rp)
if ((Gabs>21.22759-118.35415*(bp-rp)+305.43033*(bp-rp)^2-393.13909*...
        (bp-rp)^3+294.22156*(bp-rp)^4-135.40159*(bp-rp)^5+38.90143*...
        (bp-rp)^6-6.80771*(bp-rp)^7+0.66434*(bp-rp)^8-0.02773*(bp-rp)^9 ...
        && bp-rp ≥ 0.58 && bp-rp < 4.5)|| (Gabs > -0.4+6.16332-1.00990*...
        (bp-rp)-0.78017*(bp-rp)^2+4.21489*(bp-rp)^3+1.71673*(bp-rp)^4-...
        25.49710*(bp-rp)^5-13.68387*(bp-rp)^6+24.16992*(bp-rp)^7 &&...
        bp-rp > -0.65 && bp-rp < 0.58))&& ((Gabs < 11.13367+3.51045*...
        (bp-rp)-1.13132*(bp-rp)^2+0.30628*(bp-rp)^3 && bp-rp > -0.65 ...
        && bp-rp < 1.7)|| (Gabs < 14.587+0.45098*(bp-rp) && bp-rp ≥...
        1.7 && bp-rp < 4.25) || (Gabs < 40.3-5.6*(bp-rp) && bp-rp ≥...
        4.25 && bp-rp < 4.5))&& bp-rp > -0.65 && bp-rp < 4.5
    res=1;
else
    res=0;
end

end


% Calculate error for single stars or unresolved systems

function [MG_e, MBP_e, MRP_e,par_e,id]=p_error(paral, G, BP, RP)
par=paral/1000;

aG=G-5-5*log10(par); aBP=BP-5-5*log10(par); aRP=RP-5-5*log10(par);
% eliminate G>20.7
id=0;id=find(aG>20.7);
aG(id)=[]; aBP(id)=[]; aRP(id)=[]; par(id)=[];

% parallax error
%sigm=T_factor*(40+800z+30z^2)^1/2
%T_factor=1    z=max(10^(0.4(13-15)),10^(0.4(G-15)));
Tf=1; mu=0;
z3=max(10^(0.4*(13-15)),10.^(0.4*(aG-15)));
sigmp=Tf*(40+800.*z3+30.*z3.^2).^0.5*1e-6;
par_e=par+normrnd(mu,sigmp);

% G error
% sigm=1.2*10^-3(0.04895z^2+1.8633z+0.0001985)^1/2
% z=max(10^(0.4(12-15)),10^(0.4(G-15)))
z2=max(10^(0.4*(12-15)),10.^(0.4*(aG-15)));
sigmg=1.2e-3*(0.0001985+1.8633.*z2+0.04895.*z2.^2).^0.5;
aG_e=aG+normrnd(mu,sigmg);

% BP and RP error
% sigm_bp/rp=1e-3(10^az^2+10^bz+10^c)^1/2
% z=max(10^(0.4(11-15)),10^(0.4(G-15)))
[aaBP,bbBP,ccBP,aaRP,bbRP,ccRP]=abc(aBP,aRP);
z1=max(10^(0.4*(11-15)),10.^(0.4*(aG-15)));
sigmb=1e-3*(10.^ccBP+10.^bbBP.*z1+aaBP.*z1.^2).^0.5;
sigmr=1e-3*(10.^ccRP+10.^bbRP.*z1+aaRP.*z1.^2).^0.5;
aBP_e=aBP+normrnd(mu,sigmb);
aRP_e=aRP+normrnd(mu,sigmr);

[MG_e,MBP_e,MRP_e]=Me(aG_e,aBP_e,aRP_e,par_e);

end
```

```

% Calculate error for resolved systems, so that when eliminating the stars
% with G>20.7 it eliminates the system

function [MG_e1, MBP_e1, MRP_e1,MG_e2, MBP_e2, MRP_e2,par_e1,par_e2,id]=...
    p_error2(paral1, paral2, G1, BP1, RP1, G2, BP2, RP2)
par1=paral1/1000; par2=paral2/1000;

aG1=G1-5-5*log10(par1); aBP1=BP1-5-5*log10(par1); aRP1=RP1-5-5*log10(par1);
aG2=G2-5-5*log10(par2); aBP2=BP2-5-5*log10(par2); aRP2=RP2-5-5*log10(par2);

% eliminate G>20.7
id1=0;id1=find(aG1>20.7); id2=0;id2=find(aG2>20.7);
id=unique([id1 id2]);
aG1(id)=[]; aBP1(id)=[]; aRP1(id)=[]; par1(id)=[];
aG2(id)=[]; aBP2(id)=[]; aRP2(id)=[]; par2(id)=[];

% parallax error
%sigm=T_factor*(40+800z+30z^2)^1/2
%T_factor=1    z=max(10^(0.4(13-15)),10^(0.4(G-15)));
Tf=1; mu=0;

z31=max(10^(0.4*(13-15)),10.^(0.4*(aG1-15)));
sigmp1=Tf*(40+800.*z31+30.*z31.^2).^0.5*1e-6;
par_e1=par1+normrnd(mu,sigmp1);

z32=max(10^(0.4*(13-15)),10.^(0.4*(aG2-15)));
sigmp2=Tf*(40+800.*z32+30.*z32.^2).^0.5*1e-6;
par_e2=par2+normrnd(mu,sigmp2);

% G error
%sigm=1.2*10^-3(0.04895z^2+1.8633z+0.0001985)^1/2
% z=max(10^(0.4(12-15)),10^(0.4(G-15)))

z21=max(10^(0.4*(12-15)),10.^(0.4*(aG1-15)));
sigmg1=1.2e-3*(0.0001985+1.8633.*z21+0.04895.*z21.^2).^0.5;
aG_e1=aG1+normrnd(mu,sigmg1);

z22=max(10^(0.4*(12-15)),10.^(0.4*(aG2-15)));
sigmg2=1.2e-3*(0.0001985+1.8633.*z22+0.04895.*z22.^2).^0.5;
aG_e2=aG2+normrnd(mu,sigmg2);

% BP and RP error
% sigm_bp/rp=1e-3(10^az^2+10^bz+10^c)^1/2
% z=max(10^(0.4(11-15)),10^(0.4(G-15)))

[aaBP1,bbBP1,ccBP1,aaRP1,bbRP1,ccRP1]=abc(aBP1,aRP1);
z11=max(10^(0.4*(11-15)),10.^(0.4*(aG1-15)));
sigmb1=1e-3*(10.^ccBP1+10.^bbBP1.*z11+aaBP1.*z11.^2).^0.5;
sigmr1=1e-3*(10.^ccRP1+10.^bbRP1.*z11+aaRP1.*z11.^2).^0.5;

[aaBP2,bbBP2,ccBP2,aaRP2,bbRP2,ccRP2]=abc(aBP2,aRP2);
z12=max(10^(0.4*(11-15)),10.^(0.4*(aG2-15)));
sigmb2=1e-3*(10.^ccBP2+10.^bbBP2.*z12+aaBP2.*z12.^2).^0.5;
sigmr2=1e-3*(10.^ccRP2+10.^bbRP2.*z12+aaRP2.*z12.^2).^0.5;

aBP_e1=aBP1+normrnd(mu,sigmb1);
aRP_e1=aRP1+normrnd(mu,sigmr1);

aBP_e2=aBP2+normrnd(mu,sigmb2);
aRP_e2=aRP2+normrnd(mu,sigmr2);

[MG_e1,MBP_e1,MRP_e1]=Me(aG_e1,aBP_e1,aRP_e1,par_e1);
[MG_e2,MBP_e2,MRP_e2]=Me(aG_e2,aBP_e2,aRP_e2,par_e2);

end


% Calculate error for the components of an unresolved system, where the
% systems with G>20.7 have already been chosen

function [MG_e, MBP_e, MRP_e]=p_error3(paral, G, BP, RP)
par=paral/1000;
```

```matlab

aG=G-5-5*log10(par); aBP=BP-5-5*log10(par); aRP=RP-5-5*log10(par);

% parallax error
%sigm=T_factor*(40+800z+30z^2)^1/2
%T_factor=1    z=max(10^(0.4(13-15)),10^(0.4(G-15)));
Tf=1; mu=0;
z3=max(10^(0.4*(13-15)),10.^(0.4*(aG-15)));
sigmp=Tf*(40+800.*z3+30.*z3.^2).^0.5*1e-6;
par_e=par+normrnd(mu,sigmp);
% G error
%sigm=1.2*10^-3(0.04895z^2+1.8633z+0.0001985)^1/2
% z=max(10^(0.4(12-15)),10^(0.4(G-15)))

z2=max(10^(0.4*(12-15)),10.^(0.4*(aG-15)));

sigmg=1.2e-3*(0.0001985+1.8633.*z2+0.04895.*z2.^2).^0.5;

aG_e=aG+normrnd(mu,sigmg);

% BP and RP error
% sigm_bp/rp=1e-3(10^az^2+10^bz+10^c)^1/2
% z=max(10^(0.4(11-15)),10^(0.4(G-15)))

[aaBP,bbBP,ccBP,aaRP,bbRP,ccRP]=abc(aBP,aRP);

z1=max(10^(0.4*(11-15)),10.^(0.4*(aG-15)));

sigmb=1e-3*(10.^ccBP+10.^bbBP.*z1+aaBP.*z1.^2).^0.5;
sigmr=1e-3*(10.^ccRP+10.^bbRP.*z1+aaRP.*z1.^2).^0.5;

aBP_e=aBP+normrnd(mu,sigmb);
aRP_e=aRP+normrnd(mu,sigmr);

[MG_e,MBP_e,MRP_e]=Me(aG_e,aBP_e,aRP_e,par_e);

end

% Calculate the absolute magnitudes
function [MG,MBP,MRP]=Me(G,BP,RP,par)
MG=real(G+5+5*log10(par));
MBP=real(BP+5+5*log10(par));
MRP=real(RP+5+5*log10(par));
end


% Calculate the auxiliary variables a,b,c for BP and RP

function [aBP,bBP,cBP,aRP,bRP,cRP]=abc(BP,RP)
aBP=-0.000562*(BP-RP).^3 + 0.044390*(BP-RP).^2 ...
    + 0.355123*(BP-RP) + 1.043270;
bBP=-0.000400*(BP-RP).^3 + 0.018878*(BP-RP).^2 ...
    + 0.195768*(BP-RP) + 1.465592;
cBP=+0.000262*(BP-RP).^3 + 0.060769*(BP-RP).^2 ...
    - 0.205807*(BP-RP) - 1.866968;
aRP=-0.007597*(BP-RP).^3 + 0.114126*(BP-RP).^2 ...
    - 0.636628*(BP-RP) + 1.615927;
bRP=-0.003803*(BP-RP).^3 + 0.057112*(BP-RP).^2 ...
    - 0.318499*(BP-RP) + 1.783906;
cRP=-0.001923*(BP-RP).^3 + 0.027352*(BP-RP).^2 ...
    - 0.091569*(BP-RP) - 3.042268;
end
```