



UNIVERSITÀ DI PISA

Dipartimento di Fisica

Tesi di Laurea Magistrale

Anno Accademico 2013/14

Systematic Study of Mass Loss in the Evolution of Massive Stars

Candidate:
Mathieu Renzo

Advisors:
Prof. Steven N. Shore
Prof. Christian D. Ott

Summary

Mass loss is of paramount importance for the lives of massive stars. It influences their evolution, and changes their final fate (successful supernova explosion of different types, or collapse), and remnant (neutron star or black hole). Two broad categories of mass loss mechanisms are expected for massive stars: (i) radiatively driven stellar winds; (ii) extreme events (e.g. eruptions, pulsational instabilities, wave driven mass loss). The latter could, in principle, strip away large fractions of a star's mass in a very short time. In a binary system, Roche Lobe Overflow is another mass loss mechanism from the point of view of the primary star.

However, because of its intrinsically dynamical nature, and because of the strong non-linearity of the responsible physical processes driving it, mass loss is one of the largest sources of uncertainty in the simulation of massive star evolution.

In most stellar evolution simulations, only wind mass loss is included through parametric algorithms obtained as combinations of different formulae for each phase of the evolution. These formulae express the mass loss rate as a function of an (arbitrarily) chosen set of stellar parameters, $\dot{M} \equiv \dot{M}(L, T_{\text{eff}}, Z, \dots)$, and have empirical or theoretical grounds. Moreover, it is common practice to use an efficiency factor η , whose (positive) value is not unique (and lacks a direct physical interpretation derived from first principles). η modifies the rate to account for possible biases in the mass loss rate determination (e.g. potential overestimation due to the assumption of homogeneity in the observed wind structures). Although it may be dominant in terms of the total mass shed, mass loss from eruptive events is commonly neglected. These events are particularly difficult to model both because of the uncertainties in the driving process(es) and because of their inherent short timescales and multidimensionality.

This thesis attempts to understand and constrain the uncertainties connected to mass loss in the evolution of massive stars in the (initial) mass range between $15M_{\odot}$ and $30M_{\odot}$. This is done by computing a grid of stellar models with the open-source stellar evolution code MESA, modified to include some mass loss algorithms, customized stopping criteria and stricter timestep controls. I carried out several simulations with varying initial masses and differing only in the wind algorithms and efficiencies, and compare the output. I then perform a simplified numerical experiment to simulate mass stripping in a $15M_{\odot}$ star, to study how its structure and evolution change with the removal of portions of its envelope at different moments in its evolution. In a simplified way, this mass stripping procedure mimics eruptive, violent events, or possibly Roche Lobe Overflow in a binary.

The use of different wind mass loss scheme produces significant disagreement among the simulated evolutionary tracks. I show that, although the mass loss algorithms compared are just, in principle, different parametrization of the same physical phenomenon, they are *not*

equivalent. The uncertainty on mass loss increases at higher initial masses and the η factor has a strong influence on the results.

The stripped models obtained by *artificially* removing different portions of the hydrogen-rich envelope of a star at different moments of its evolution all have the same core structure (i.e. the mass stripping at the moment chosen does not influence the core structure), but significant differences are found at intermediate masses. I also find significant variation in the density gradient within the hydrogen-rich envelope that remains at the onset of core collapse.

In Chapter 1, I review the importance of massive stars in the broad context of astrophysics, and outline the standard picture of their evolution. Then, I present the current challenges encountered in the numerical modelling of these stars with one-dimensional stellar evolution codes. Finally, I discuss the possible mass loss channels for massive stars and review the mass loss algorithms commonly used to model steady wind mass loss. Signatures of stellar winds which allow observational determinations of the mass loss rate are discussed too.

In Chapter 2, I review the basics of the MESA code, and present the customized routines that I implemented for this work. I also describe the setup of the simulations in detail, with the explicit aim of providing all the information needed to reproduce the results. Chapter 2 also contains the description of the simplified procedure to simulate a violent, short, eruptive mass loss event or an envelope stripping caused by a companion star, and the advantages and shortcomings of this method.

In Chapter 3, I present my grid of models to compare, in a systematic way, the various wind mass loss algorithms. I compare separately the algorithms for each evolutionary phase, and I discuss the different resulting evolutionary tracks and final characteristics of the stellar structures.

In Chapter 4, I discuss the results of the simplified model for an envelope shedding mass loss event in a $15M_{\odot}$ star, such as may arise from pulsational or wave driven mass loss events, or in the evolution of a binary system when the primary star increases its radius. I compare the outcome of the simplified procedure adopted to remove the envelope to the unstripped reference model, and discuss the pre-supernova structures resulting from these simulations.

Chapter 5, summarizes the main results and suggests possible observational probes and further research directions to better understand the uncertainties in massive stars mass loss.

Appendix A contains the routines implemented for this work, and the MESA parameter files used. Appendix B is a discussion of some of the computational issues encountered in this work.

Key words: massive stars – stellar evolution – mass loss – stellar winds – numerical simulations.

"The best things in life aren't things"
[Unknown on a wall somewhere]

Contents

1	Introduction	1
1.1	The Importance of Massive Stars	1
1.2	Evolution of a Single Massive Star	2
1.2.1	Main Sequence Evolution	3
1.2.2	The “Overall Contraction” and the Subgiant Branch	5
1.2.3	Post-SGB Evolution	6
1.2.4	Core Collapse	9
1.2.5	The Supernova Zoo	10
1.3	Current Challenges in the Numerical Simulation of Massive Stars	12
1.3.1	Nearly Super-Eddington Radiation Dominated Convective Envelopes	12
1.3.2	Silicon Burning	15
1.3.3	Lack of Systematic Studies to Quantify the Many Uncertainties	16
1.4	Massive Star Mass Loss	17
1.4.1	Outline of the Theory of Stellar Winds	18
1.4.2	<i>de Jager et al.</i>	22
1.4.3	<i>Nieuwenhuijzen et al.</i>	27
1.4.4	<i>van Loon et al.</i>	28
1.4.5	<i>Vink et al.</i>	28
1.4.6	<i>Kudritzki et al.</i>	30
1.4.7	<i>Nugis & Lamers</i>	32
1.4.8	<i>Hamann et al.</i>	33
1.4.9	Motivation to introduce non-wind Mass Loss Processes in Simulations of Massive Star Evolution	34
2	Methods	37
2.1	The MESA Code	37
2.1.1	Overview of the Numerical Implementation	37
2.1.2	Run time	43
2.1.3	MLT++	44
2.2	Modifications to the MESA Code	46
2.2.1	Implementation of Customized Timestep Controls	46

CONTENTS

2.2.2	Implementation of the Mass Loss Prescriptions	48
2.3	Systematic Comparison of Wind Mass Loss Algorithms	51
2.4	Simplified Simulation of Envelope Shedding Mass Loss Events	54
2.4.1	The Stripping Process	57
3	Results: Wind Algorithms Comparison	61
3.1	Overview	61
3.1.1	The Compactness at Oxygen Depletion	73
3.2	Hot Phase Mass Loss	75
3.2.1	Effects of the detailed Treatment of the Bistability Jump in the Vink <i>et al.</i> Algorithm	81
3.2.2	Effects of the Hot Phase Mass Loss on the Helium Core	82
3.3	Cool Phase Mass Loss	85
3.3.1	Comparison of the Jager <i>et al.</i> and Nieuwenhuijzen <i>et al.</i> Algorithms	87
3.3.2	The Van Loon <i>et al.</i> Algorithm	87
3.3.3	On the Morphology of the Blueward Evolution and Blue Loops	88
3.4	WR Mass Loss	90
4	Results: Simplified Envelope Shedding Mass Loss	97
4.1	Introduction	97
4.2	Stellar Structures after the Stripping Process	97
4.3	Stripped Models at the Onset of Core Collapse	103
5	Discussion and Conclusion	107
5.1	Context Summary	107
5.2	Wind Mass Loss	108
5.3	Envelope Shedding Events	110
5.4	Directions for Further Work	111
A	MESA parameters files and customized routines	113
A.1	The Inlist Parameters Files	113
A.1.1	inlists for the Systematic Comparison of Wind Mass Loss	113
A.1.2	inlist for the Simplified Envelope Shedding Mass Loss Events	117
A.2	Routines for the run_star_extras.f	122
A.2.1	Vink <i>et al.</i> , de Jager <i>et al.</i> , Nugis & Lamers – VdJNL	122
A.2.2	Vink <i>et al.</i> , de Jager <i>et al.</i> , Hamann <i>et al.</i> – VdJH	125
A.2.3	Vink <i>et al.</i> , Nieuwenhuijzen <i>et al.</i> , Nugis & Lamers – VNJNL	128
A.2.4	Vink <i>et al.</i> , Nieuwenhuijzen <i>et al.</i> , Hamann <i>et al.</i> – VNJH	131
A.2.5	Vink <i>et al.</i> , van Loon <i>et al.</i> , Hamann – VvLH	135
A.2.6	Vink <i>et al.</i> , van Loon <i>et al.</i> , Nugis & Lamers – VvLNL	138
A.2.7	Kudritzki <i>et al.</i> , de Jager <i>et al.</i> , Nugis & Lamers – KdJNL	141

CONTENTS

A.2.8	Kudritzki <i>et al.</i> , de Jager <i>et al.</i> , Hamann <i>et al.</i> – KdJH	143
A.2.9	Kudritzki <i>et al.</i> , Nieuwenhuijzen <i>et al.</i> , Nugis & Lamers – KNJNL	145
A.2.10	Kudritzki <i>et al.</i> , Nieuwenhuijzen <i>et al.</i> , Hamann <i>et al.</i> – KNJH	147
A.2.11	Kudritzki <i>et al.</i> , van Loon <i>et al.</i> , Hamann <i>et al.</i> – KvLH	149
A.2.12	Kudritzki <i>et al.</i> , van Loon <i>et al.</i> , Nugis & Lamers – KvLNL	151
A.2.13	Timestep Controls	152
A.2.14	run_star_extras.f for the Simplified Envelope Shedding Events	157
B	From a naive use of MESA toward physically sound Results	159
B.1	Warning to the Naive MESA User	159
B.2	Surface Oscillations	160
B.3	On the Resolution at high Temperatures	162
B.3.1	Introduction to the Problem	162
B.3.2	Need for a large Nuclear Reaction Network	163
B.3.3	Spatial Resolution of the Core	165

CONTENTS

CHAPTER 1

Introduction

[...] it is reasonable to hope that in the not too distant future we shall be competent to understand so simple a thing as a star.

[A. Eddington, *The Internal Constitution of Stars*, 1926]

1.1 The Importance of Massive Stars

According to the usual definition, “massive stars” are those with zero age main sequence (ZAMS, see §1.2 for the definition) mass M_{ZAMS} sufficiently large to form a degenerate oxygen/neon or iron core at the end of their globally hydrostatic evolution [1]. Typically, this means $8 - 10 \lesssim M_{\text{ZAMS}}/M_{\odot} \lesssim 150 - 200$. Both limits depend on the initial condition (especially the initial metallicity Z of the star, the values cited are for solar metallicity [1]), and they are still not well known because of the large uncertainties involved in modeling the evolution of such stars.

Because of their large mass, these stars are rather rare objects: they seldom form and are short lived. Nevertheless, their characteristics (e.g. high luminosity, large mass loss rate, complex nuclear burning, final fate, etc.) make them extremely important for many sub-fields of astrophysics. For example, because of their high luminosity, they are the only stars that can be observed in outer galaxies. They are paramount for the early universe re-ionization [2] and they create HII regions due to their ionizing radiation; the nuclear processes in their interiors are responsible for the production of most of the isotopes up to the iron group (e.g. [1, 3, 4], see also below), and their winds and/or final explosion as supernova (SN) release the matter processed by nuclear reactions, chemically enriching the interstellar medium (ISM, see e.g. [5]). Their winds also input momentum into the ISM, blowing giant bubbles and possibly triggering star formation [6, 7]. Their explosions too input momentum into the ISM, which can sweep away the surrounding matter and thus damp further star formation. In their final gasps of life, their cores become the environment for the interplay of a large number of fundamental physical phenomena: weak interaction enters because of neutrino cooling and electron captures [8], strong interaction and electromagnetic interaction are involved in the nuclear processes (e.g. [9–12]). Finally, when the central nuclear engine shuts down because of the lack of fuel, and nothing can sustain the core, gravitational collapse ensues, and gravity drives the dynamics (e.g. [13, 14]). The collapse can trigger one of the most energetic phenomena in the universe, a SN, and create a black hole (BH) or a neutron star (NS) [15, 16]. All these reasons make the study of massive stars an interesting and important topic.

The structure of the star at the onset of core-collapse strongly influences the final fate of the star, in ways that are a current research topic (see, e.g. [14–19]). But of course, the structure at the onset of collapse itself is the result of the previous evolution, during the globally hydrostatic and thermal equilibrium stages of core and shell nuclear burning.

1.2 Evolution of a Single Massive Star

Although virtually all massive stars are observed to be in binary (or multiple) systems, and nearly $\sim 70\%$ are expected to interact with their companion star [20], most modeling still relies on numerical simulations of single isolated stars. The nature of the interaction(s) between the binary companions and the precise moment it becomes relevant for the stars depend on the parameters of the binary system. The goal of this section is to give a qualitative outline of the evolution of a *single* massive star, according to current stellar evolution theory for non-rotating stars. The aim is to outline the various evolutionary stage for further reference. I do not discuss the formation of massive stars. The reader is referred to [21–24] or any other stellar astrophysics textbook for more details.

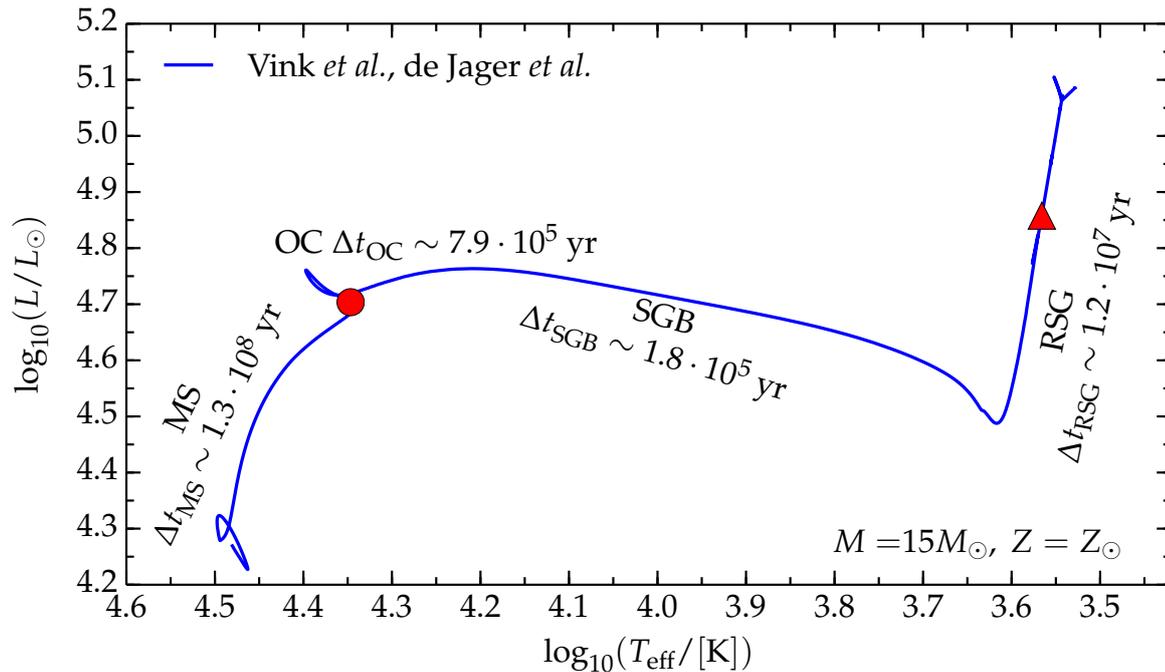


Figure 1.1: Theoretical HertzsprungRussell (HR) diagram (i.e. $(-T_{\text{eff}}, L)$ plane) for a $15 M_{\odot}$, $Z = Z_{\odot} \equiv 0.019$ [25] stellar model computed with MESA (see §2.1). The red dot indicates the terminal age main sequence, where the abundance of hydrogen in the core is $X_c < 0.01$ (TAMS), the red triangle indicates He exhaustion in the core (central abundance of helium $Y_c < 0.01$). The labels MS, OC, SGB, RSG stand for Main Sequence, Overall Contraction, SubGiant Branch and Red SuperGiant, respectively. The duration of each stage is indicated. The legend indicates the wind mass loss scheme employed, see §1.4.

As for all stars, massive stars are globally hydrostatic and at thermal equilibrium, self-gravitating, gaseous structures. Their evolution is governed by the consumption of nuclear fuel in the inner regions, which balances the energy loss from the surface with the energy

released from nuclear reactions. For any reasonable initial composition, the first nuclear fuel is hydrogen, and, usually, the evolution of the star is simulated from the so-called zero age main sequence (ZAMS). This is the moment when the star is supported by core hydrogen fusion *and* the abundances of the catalytic nuclei¹ have reached constant values. However, common variations of this definition may be adopted for computational purposes (e.g. defining the ZAMS as the point of the evolution where 99% of the luminosity comes from hydrogen burning).

Table 1.1: Indicative duration of several core burning phases for different initial mass M_{ZAMS} . I consider each phase to begin at the end of the previous one (i.e. the shell burning/inert core phase duration is included in the next burning phase) and to end when the abundance of all the isotopes of the burning specie drops below 0.01. The data come from MESA (see §2.1 and references therein) simulations using a 21-isotope nuclear network (approx21.net). See also §1.3.2, and §B for more details.

Core Burning $M_{\text{ZAMS}} [M_{\odot}]$	duration [yrs]			
	15	20	25	30
H	$\sim 1.29 \times 10^7$	$\sim 8.93 \times 10^6$	$\sim 7.05 \times 10^6$	$\sim 5.98 \times 10^6$
He	$\sim 1.18 \times 10^6$	$\sim 8.63 \times 10^5$	$\sim 7.00 \times 10^5$	$\sim 6.06 \times 10^5$
C	$\sim 4.04 \times 10^4$	$\sim 2.58 \times 10^4$	$\sim 2.13 \times 10^4$	$\sim 1.73 \times 10^4$
Ne	$\sim 1.76 \times 10^2$	$\sim 2.89 \times 10^1$	$\sim 1.14 \times 10^1$	$\sim 3.96 \times 10^0$
O	$\sim 1.25 \times 10^0$	$\sim 4.36 \times 10^{-1}$	$\sim 5.56 \times 10^{-3}$	$\sim 3.14 \times 10^{-1}$
Si	$\sim 1.39 \times 10^{-1}$	$\sim 2.66 \times 10^{-2}$	$\sim 3.61 \times 10^{-2}$	$\sim 1.03 \times 10^{-1}$
Total	$\sim 1.41 \times 10^7$	$\sim 9.82 \times 10^6$	$\sim 7.78 \times 10^6$	$\sim 6.60 \times 10^6$

1.2.1 Main Sequence Evolution

The main sequence is the longest evolutionary stage of any star, during which hydrogen is processed in the core, and there are no other nuclear reaction chains. In massive stars, because of the high central temperature, hydrogen is consumed via the CNO cycle [14]. The virial theorem² in the case of hydrostatic equilibrium ($\vec{\mathcal{I}} = 0$, where \mathcal{I} is the momentum of inertia) states that

$$2\mathcal{K} + \mathcal{U} = 0, \quad (1.1)$$

where \mathcal{K} is the total kinetic energy, and \mathcal{U} the potential energy. The latter can be expressed as (except for a dimensionless order unity constant depending on the details of the mass distribution)

$$\mathcal{U} \sim -\frac{GM^2}{R} \sim -\frac{GM^2 \langle \rho \rangle^{1/3}}{M^{1/3}}, \quad (1.2)$$

where I introduce the mean density $\langle \rho \rangle = M/(4\pi R^3/3)$ in order to eliminate the radius. The kinetic energy \mathcal{K} can be expressed (apart from constants of order unity depending on

¹The catalytic nuclei are those which are both produced and destroyed in the nuclear reaction chains through which the matter is processed.

²The virial theorem can be applied to stars since these are globally in thermal equilibrium.

the details of the chemical composition) as

$$\mathcal{K} \sim k_b \langle T \rangle N \sim k_b \langle T \rangle \frac{M}{\langle \mu \rangle m_p} , \quad (1.3)$$

where k_b is the Boltzmann constant, m_p is the proton mass, $\langle \rangle$ indicate quantities averaged over the entire star, N is the total number of particles, expressed as the total mass M divided by the mass of a “mean particle” and μ is the mean molecular weight, which, for a completely ionized gas, corresponds to

$$\mu = \sum_i \frac{1}{X_i \frac{Z_i+1}{A_i}} , \quad (1.4)$$

where X_i is the abundance of the i – *th* isotope, Z_i is number of electrons (equal to protons), and A_i the total number of nucleons. Thus, μ can be interpreted as the inverse of the total number of particles (Z_i electrons plus one nucleus), per unit of baryonic mass. Substituting Eq. 1.2 and Eq. 1.3 into Eq. 1.1, it is easily found that the average temperature $\langle T \rangle$ is proportional to

$$\langle T \rangle \propto M^{2/3} \langle \rho \rangle^{1/3} \langle \mu \rangle . \quad (1.5)$$

Thus, the more massive the star, the higher its average temperature $\langle T \rangle$ and its central temperature T_c . Since the energy generation rate per unit mass of the CNO cycle (ε_{CNO}) is a stiffer function of the temperature than the energy generation rate per unit mass of the other hydrogen burning cycle (the PP chain) [22],

$$\varepsilon_{\text{CNO}} \propto \rho^2 T^{18} \quad \text{cf.} \quad \varepsilon_{\text{PP}} \propto \rho^2 T^4 , \quad (1.6)$$

Eq. 1.5 explains why the CNO cycle is the dominant channel for hydrogen burning in massive stars³. The caveat is that C, N and O (which are among the most abundant metals in the universe, [3, 4]) must be present in the initial composition [24]. In other words, the CNO cycle has a peculiar dependence on the metallicity Z , while the PP-chain does not depend on the presence of nuclei heavier than helium.

Because of the strong dependence of ε_{CNO} on T , during core hydrogen burning, the nuclear processing happens only in the hottest central region, and its extent is quite small compared to the pressure scale height. Therefore, in the central regions the temperature gradient is large and the core is convective, while the outer envelope is radiative, cf. Fig. 1.2. Convective mixing flattens the chemical abundances in the entire convective region (which is larger than the burning region), and it provides hydrogen fuel to the burning core from a region much larger than where T is enough for nuclear burning to occur.

During its main sequence evolution, the star climbs the HR diagram, increasing its luminosity and consuming the hydrogen in the entire convective core. Since the cross section for photon interactions with (ionized) helium is smaller than the cross section for interactions with hydrogen (i.e. free protons and electrons in a completely ionized medium such as the stellar core), the consumption of hydrogen and the corresponding production of helium cause a decrease in the opacity κ in the entire convective region. The lower opacity decreases the radiative gradient, and this in turn partially stabilizes the outer convective region. Furthermore, the decreased amount of hydrogen fuel requires an higher density of the core to provide the same energy generation rate. Therefore, the core tend to contract and increase its temperature and luminosity L , and the envelope responds expanding slightly.

³Note that the exact dependence of $\varepsilon \equiv \varepsilon(T)$ is not a power law, and the approximate exponent depends on the density and temperature range.

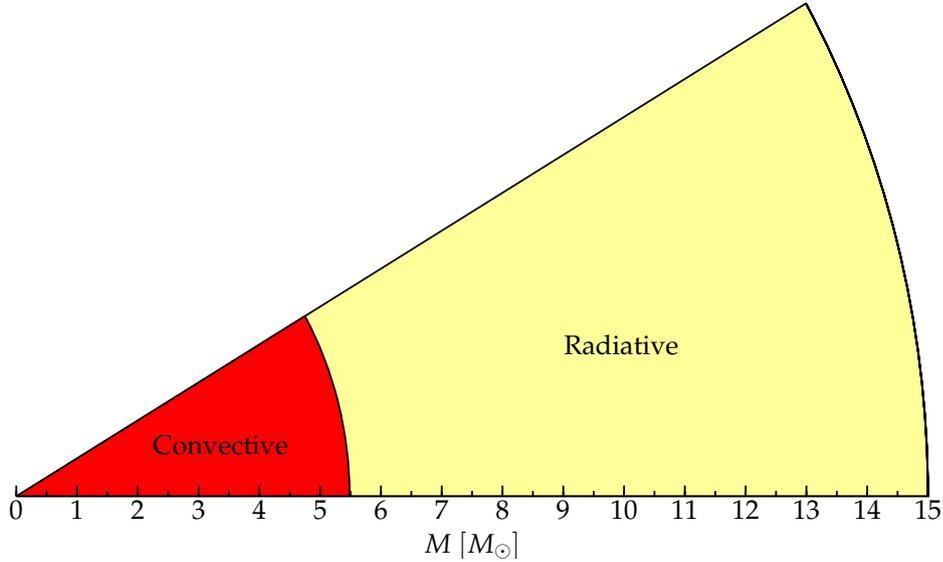


Figure 1.2: Schematic structure of the star of Fig. 1.1 during the main sequence (at $t \sim 1900$ years after ZAMS). The radius of the wedge is proportional to the total mass, the red region is convective, while the yellow region is radiative.

1.2.2 The “Overall Contraction” and the Subgiant Branch

When the star reaches the red dot in Fig. 1.1, a helium-rich core is formed, and hydrogen is depleted in the center. This condition can be taken as the formal definition of the end of the main sequence. The locus of the HR diagram points corresponding to this condition for different sets of initial parameters (M_{ZAMS} , Z , etc..) is usually referred to as the “Terminal Age Main Sequence” (TAMS).

Because of the convective mixing in the core during main sequence evolution, hydrogen is depleted in a region much larger than the region where T is high enough to trigger nuclear reactions. Therefore, when the star reaches the red dot in Fig. 1.1, it lacks fuel not only in its center, but in a region large enough that the ignition of further nuclear reactions cannot be smooth and immediate. There is a short phase of homologous “overall contraction” (OC), during which the star shrinks in radius and increases its temperature until it is able to ignite hydrogen burning in a shell at the helium core’s edge (e.g. [22]).

The hydrogen shell is an off-center nuclear power source. When it turns on, the envelope above the shell starts inflating and cooling. Thus, the star moves across the HR diagram roughly at constant luminosity. The corresponding horizontal track on the HR diagram is often called the “Sub-Giant Branch” (SGB), and it is identified, from the observational point of view, as the so-called Hertzsprung gap. The time spent in this expansion phase is much shorter ($\lesssim 10^5$ years) than both the main sequence duration and the subsequent red supergiant (RSG) phase (see Fig. 1.1). This is the reason for the small number of stars observed in this phase. During the SGB, the star inflates and cools so much that the temperature gradient becomes steeper and steeper in order to connect the high temperature (for the $15M_{\odot}$ star of Fig. 1.1, $T_{\text{H shell}} \sim \text{few} \times 10^7$ K) of the shell with the low temperature at the surface of the envelope ($T_{\text{eff}} \sim \text{few} \times 10^3$ K). The low temperature at the outer boundary of the envelope causes the onset of convection.

As the extent of the convective envelope grows, the stellar envelope becomes similar to a

homogeneous convective structure, whose place on the HR diagram would be the Hayashi track (e.g. [22]). Thus the stellar track on the HR diagram bends and the star enters the RSG phase.

1.2.3 Post-SGB Evolution

During the SGB (and at least also during the beginning of the RSG) evolutionary stage, the envelope is supported by hydrogen shell burning, and the inert core below it, which is made mainly of helium, contracts and grows in mass because of the ashes of the shell. Above the shell, there is a relatively small radiative region separating the burning shell from the convective portion of the envelope. Its extent depends on the total mass of the star, and the details of the treatment of convection. The radiative layer exists because convection in the envelope is driven by the low temperature at the outer boundary, not by the extreme energy output of the shell itself. Thus, in the region just above the shell, radiative processes are sufficient to carry the energy flux out. In massive stars, the helium core is never supported by electron degeneracy pressure because of its high temperature, and the ignition of helium below the hydrogen shell happens relatively quickly during (or even before) the RSG phase, depending on the total mass.

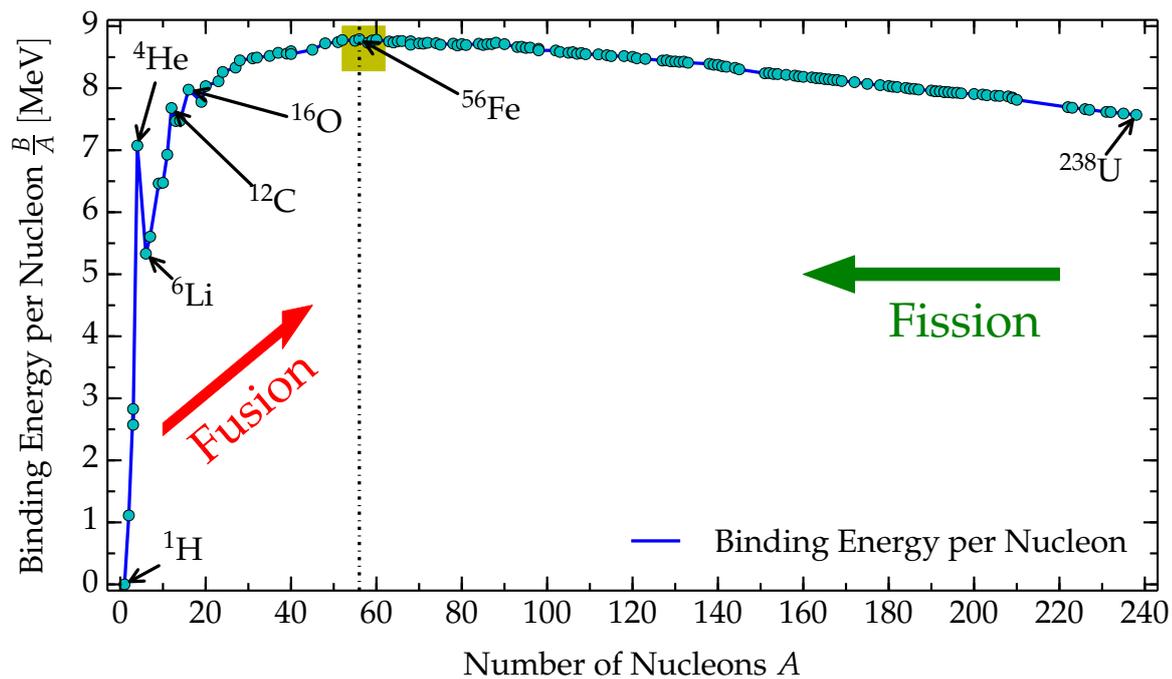


Figure 1.3: Average binding energy per nucleon as a function of the atomic mass number A . The red arrow indicates the direction in which nuclear fusion proceeds, producing heavier and more tightly bound nuclei. The green arrow indicates the direction in which nuclear fission proceeds, breaking heavy nuclei into more tightly bound pieces. The yellow box around ^{56}Fe indicates the so-called “iron group” nuclei, with $52 \leq A \leq 62$. The data are from http://www.einstein-online.info/spotlights/binding_energy-data_file/index.txt/view.

Helium burning takes place in the inner core, below the hydrogen shell, and is much faster than the previous core hydrogen burning. In general, the higher the mass of the ele-

ment being burned, the lower the energy release per nucleon. Hydrogen burning releases $\sim 7 \text{ MeV nucleon}^{-1}$, [26], while silicon burning (which is the last burning stage) releases only $\sim 0.1 \text{ MeV nucleon}^{-1}$, [9]. This can be easily understood by looking at Fig. 1.3, which shows the average nuclear binding energy per nucleon B/A : the energy released by nuclear fusion is roughly⁴ the difference between the binding energy of the final products minus that of the initial nuclei. As the number of nucleons A approaches $A_{\text{max}} \simeq 56$, this difference becomes progressively smaller and the energy release per nucleon diminishes. Thus, during advanced burning stages, the core temperature needs to increase, to increase the nuclear reaction rates, and sustain the star through nuclear fusion. Moreover, at the high temperatures occurring in late burning stages, thermal neutrinos (from pair production, bremsstrahlung, electron-positron annihilation and/or any other non-nuclear process, [27]) add another energy loss term that must be balanced by the nuclear reaction energy release, further speeding up the evolution. Thus, heavier elements burn and are depleted faster, as Tab. 1.1 illustrates.

While the evolutionary track before helium ignition is (at least qualitatively) well established, what happens subsequently is harder to follow on the HR diagram. The evolutionary track can depart significantly from the cool RSG track depending on the physical processes included in the simulation (rotation, mass loss rate, mixing, etc.). In the example shown in Fig. 1.1, the star remains a RSG until the onset of core collapse, but other models (see §3 and §4) may evolve toward higher temperatures (the so called blue loop), and end their life as yellow or blue supergiants (YSG or BSG, respectively), or they can return to the cool side of the HR diagram after a blue loop. This blueward evolution can be triggered by many different physical processes, such as mass loss removing the cold extended envelope and revealing the hotter inner regions, chemical mixing bringing heavier nuclei toward the surface, etc. (see also [28] and references therein).

A better understanding of the physics triggering the blueward evolution is required for the solution of the so-called “Red Supergiant problem”, [29]. This concerns the unknown fate of massive stars in the mass range $16M_{\odot} \lesssim M_{\text{ZAMS}} \lesssim 30M_{\odot}$. For stars in this mass range, evolutionary models (computed with standard assumptions, especially for \dot{M}) predict evolution to RSG with extended hydrogen-rich envelopes (e.g. [14, 29]). These are expected to die as Type IIP core-collapse SNe (see §1.2.4 and §1.2.5). However, the upper limit for the progenitor mass for observed SN of this type is only of $\sim 16M_{\odot}$ [29], which raises the question of what the final fate is for stars of $16M_{\odot} \lesssim M_{\text{ZAMS}} \lesssim 30M_{\odot}$ that are not massive enough to shed their hydrogen envelope (within the standard set of assumptions of stellar astrophysics), [30], but that do not end their life as predicted by theory.

While the surface properties of massive stars in late evolutionary stages are still uncertain (see also §2.1.3), the qualitative behavior of their cores is more established. After helium depletion, the core is made mainly of carbon and oxygen. These become the next nuclear fuel, with carbon igniting first. Above the carbon-oxygen core, two shell sources exist burning helium and hydrogen, respectively. During (late) carbon burning, a large fraction of the energy of the core is carried out by thermal neutrinos produced because of the high temperature reached [27]. This cooling via neutrinos has a complex and yet not well understood relation to convective instability, which also depends significantly on the initial mass of the star. This strongly influences the final structure of the core [8]. Moreover, neutrinos leave the star unimpeded (because of the inherently small weak-interaction cross sections), which even further accelerates the evolution. This neutrino cooling becomes the dominant energy

⁴Corrections are needed in order to account, e.g., for the energy that goes into neutrinos, which leave the star immediately.

loss process in the late evolutionary stages.

After core carbon depletion, the star contracts and heats again, until it ignites neon (through photodisintegrations), and then oxygen, and finally silicon. Each fuel type is made of the ashes of the previous burning stages. For every new element processed in the core, a shell of the old type of fuel ignites above it, leading to the characteristic pre-SN onion-skin structure, see Fig. 1.4.

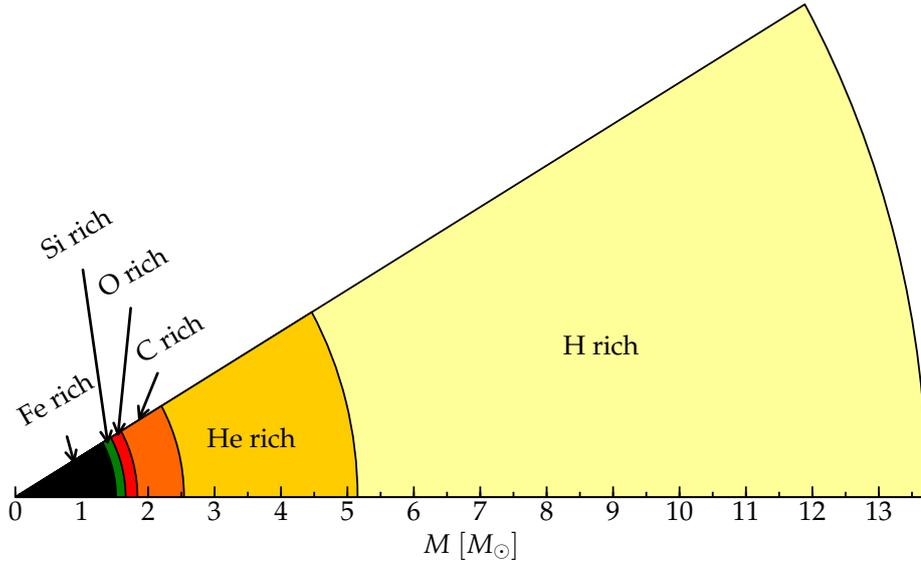


Figure 1.4: Schematic structure of the model of Fig. 1.1 at the onset of core-collapse (see Eq. 1.9). The radius of each wedge is proportional to its mass. Note that the final mass is lower than $M_{\text{ZAMS}} (= 15M_{\odot})$ because of the (wind) mass loss. At the interface between each shell there is a nuclear burning region using the material of the overlying region as fuel.

At the end of silicon burning, the star is left with a core composed of nuclei of the iron group (mainly iron isotopes), that is too massive to be sustained by electron degeneracy pressure. This means, [8],

$$M_{\text{Fe}} \geq M_{\text{Ch}}^{\text{eff}} \sim (5.83M_{\odot})Y_e^2 \left[1 + \left(\frac{s_e}{\pi Y_e} \right)^2 \right] \quad (1.7)$$

where M_{Fe} is the mass of the iron core and $M_{\text{Ch}}^{\text{eff}}$ is the effective Chandrasekhar mass. In Eq. 1.7, s_e is the electronic entropy per baryon in units of the Boltzmann constant k_b and

$$Y_e \stackrel{\text{def}}{=} \sum_i X_i \frac{Z_i}{A_i}, \quad (1.8)$$

is the number of electrons per baryons. The sum runs over all isotopes and the notation is the same as in Eq. 1.4. Eq. 1.7 yields the usual value $M_{\text{Ch}} \sim 1.44M_{\odot}$ for $Y_e \sim 0.5$ and $s_e \sim 0$. The second term in brackets includes the thermal corrections.

1.2.4 Core Collapse

Since the nuclei of the iron group are the most tightly bound (see also Fig. 1.3), the fusion of two of them would require energy input greater than the energy released. Therefore, inside the iron core, fusion reactions cannot compensate the energy loss of the star, and the core is doomed to collapse. The conventional definition for the onset of collapse [31] is

$$\max\{|v|\} \geq 10^3 \text{ [km s}^{-1}\text{]} , \quad (1.9)$$

where v is the radial infall velocity. The arbitrary threshold set by Eq. 1.9 is motivated by the fact that, at this point, the star is a few tenths of seconds (roughly a dynamical timescale) away from “core bounce” (see below). The central density is so high ($\rho \gtrsim 10^{10} \text{ g cm}^{-3}$) that stellar evolution codes usually cannot properly simulate the physics needed (e.g. the high density regions require a different equation of state - EOS, hydrostatic equilibrium does not hold any longer, neutrinos start to be trapped because of the higher density and neutrino opacity). However, this is a purely technical threshold, while in nature the evolution of such a star is continuous during collapse. Fig. 1.5 illustrates the velocity profile of a $15 M_{\odot}$ star at the onset of core collapse.

During collapse, electron capture reactions, e.g.,

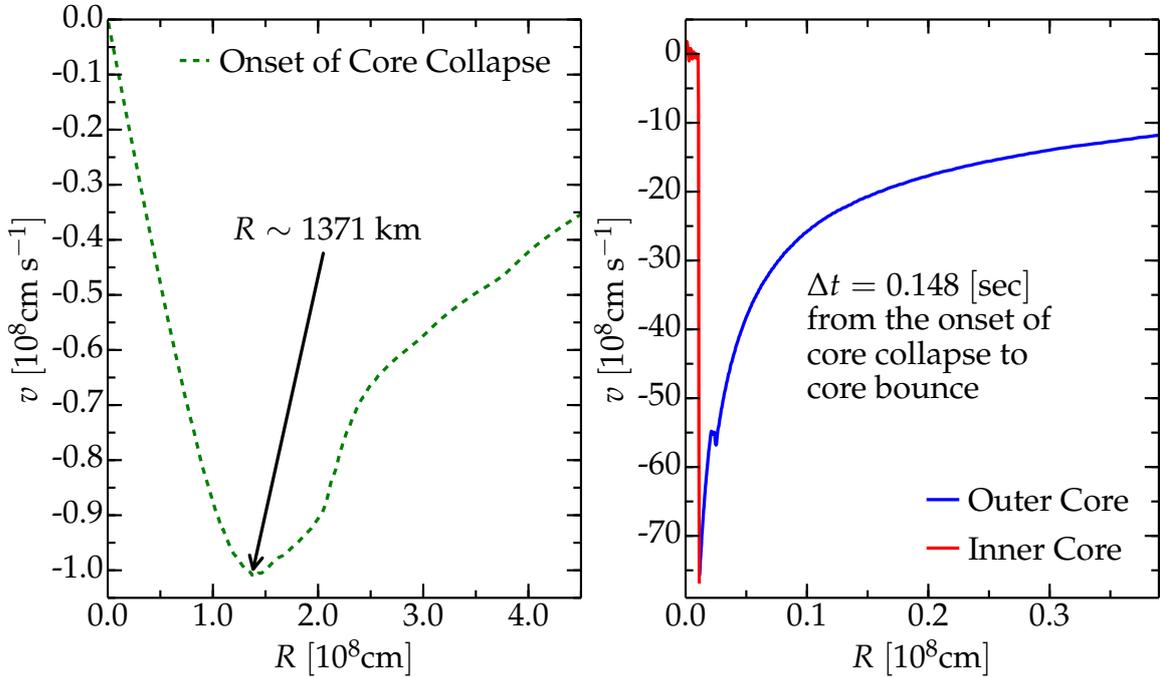
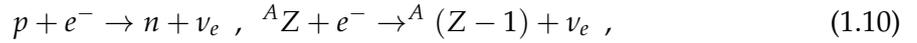


Figure 1.5: Velocity profile for the star in Fig. 1.1 at the onset of core-collapse (left panel) and at core bounce (right panel). Note the linear behavior of the infall velocity in the inner core on the left panel. Note also the different scales on the two panels: at the onset of core collapse, the infall velocity is still subsonic and directed inward ($v < 0$) everywhere. The data in the right panel are obtained using the open-source code GR1D, [32], with the data at the onset of core collapse as input.

decrease Y_e , and diminishing $M_{\text{Ch}}^{\text{eff}}$ (see Eq. 1.7), and accelerating the collapse further. Together with positron capture reactions, electron capture reactions form the so-called URCA processes, responsible for the lion's share of the cooling (provided by neutrinos) during the collapse phase. As the infall velocity progressively increases, the core divides into two separate parts [14]:

- **Inner Core:** in sonic contact and collapsing self-similarly (i.e. the infall velocity $|v| \propto r$). Its mass is given by:

$$M_{\text{i.c.}} = \int_{|v(r)| \leq c_s(r)} 4\pi\rho(r)r^2 dr , \quad (1.11)$$

where $c_s \equiv c_s(r)$ is the local sound speed, and the integral can be evaluated analytically⁵. The value of $M_{\text{i.c.}}$ at core bounce is almost independent of the stellar progenitor [33, 34].

- **Outer Core:** in supersonic collapse, because at lower density the sound speed c_s decreases, so no information about the inner core can reach into the outer core.

The collapse goes on until the central density is so high ($\rho_c \sim 10^{14} \text{ g cm}^{-3}$) that the repulsive core of the nuclear force becomes relevant. This repulsive contribution causes a sudden stiffening of the EOS, and triggers the so-called core bounce, which is conventionally defined by an arbitrary threshold on the specific entropy at the edge of the inner core: $s = 3$ (in units of the Boltzmann constant k_b). The physical picture of the core bounce is the following. The inner core overshoots the equilibrium density of the stiffened EOS, stops collapsing and reverses its radial velocity. This launches a shock wave at the edge of the inner core. It is thought that this shock wave (at least in some cases) disrupts the star, producing a SN, but the explosion mechanism is still unknown. In fact, as the shock wave propagates in the outer core, it loses energy by heating and photodisintegrating the infalling material. Moreover, neutrino losses from behind the shock diminish its energy. The energy loss through these mechanisms leads to a stalled shock in *all numerical simulations* available to date, e.g. [3, 4, 13, 35, 36]. Therefore, some uncertain "shock revival mechanism" must act to revive the shock and allow it to unbind the stellar envelope and produce a SN explosion [14].

1.2.5 The Supernova Zoo

As described in §1.2.4, the presumable final fate of a massive star is a core-collapse SN. The observational classification of a core-collapse SN depends on the spectrum and light curve it produces (e.g. [37] and references therein). The connection between the stellar progenitor and the resulting SN (if there is a successful explosion) is a topic of active research (e.g. [19, 30, 36]). For the sake of completeness, I report here the schematic classifications of all SN types, including also those that do not involve core-collapse, see Fig. 1.6.

It is worth underlining that the SN classification is to a great extent historic and does not always match the physics of the progenitor star and of the explosion. The first distinction between SN-types is based on the presence of hydrogen lines in the spectrum: the SNe showing no hydrogen are classified as type I SNe, while those with hydrogen are classified as type II SNe.

⁵The dominant pressure term at high density ($\rho_c \gtrsim \text{few} \times 10^9 \text{ [g cm}^{-3}\text{]}$) is due to relativistic degenerated electrons, so we can take a polytropic EOS $P \propto \rho^{4/3}$ to evaluate c_s .

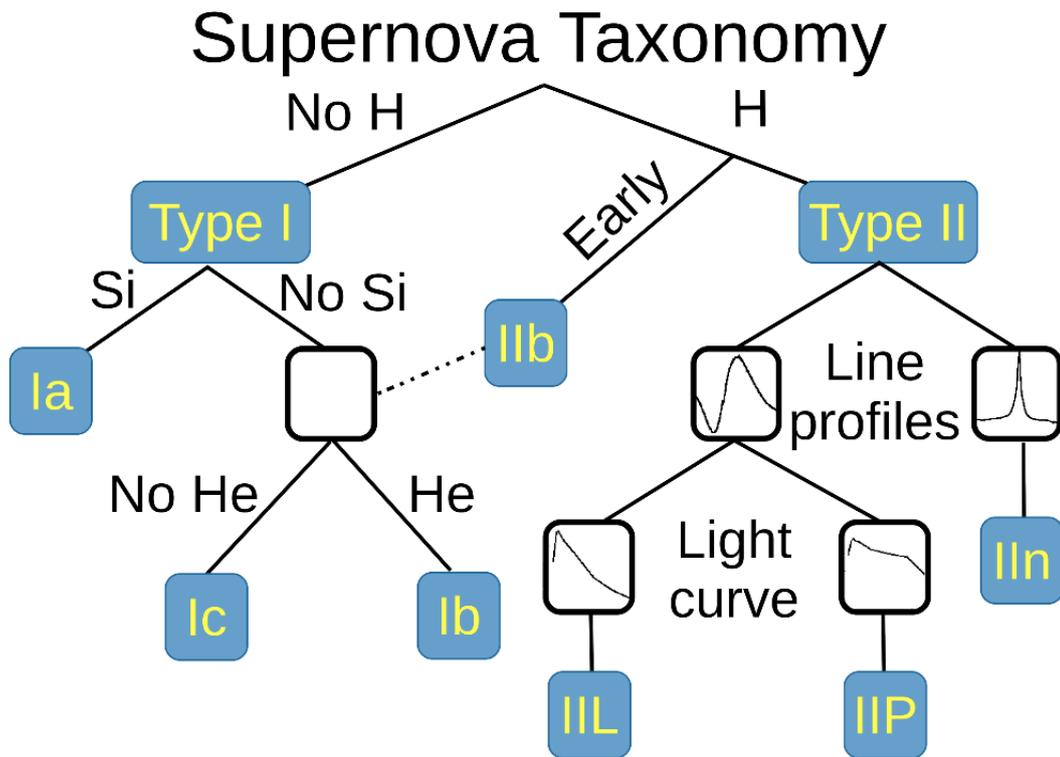


Figure 1.6: Schematic representation of the SN taxonomy, based on spectral features and light curve shape. This figure is inspired by Fig. 2 in [37]. The dot-dashed line indicates the possible connection between SN events detected in late stages and classified as type Ib/Ic and SNe of type IIb.

Type I SNe are further subdivided based on the presence of other elemental lines. Those showing silicon lines are type Ia SNe, and the proposed progenitor is *not* a massive star, but instead a white dwarf (WD) that experiences a thermonuclear explosion triggered by merger or mass accretion. In this scenario, the explosion happens only when a certain threshold condition (the mass becomes greater than the Chandrasekhar mass) is met, explaining the homogeneity in the luminosity decay and spectral features of the objects in this class (however, see [37] and references therein for further discussion). Type I SNe without silicon lines are further divided into those showing helium lines (type Ib), and those without helium lines (type Ic). These type Ib/Ic are thought to be the outcome of the collapse of a massive star that has lost all or most of its envelope.

The subdivision of type II SNe is instead based on the shape of the emission lines: the presence of narrow emission lines classifies the SN event as a type II_n (where “n” stands for narrow). These are usually very bright SNe, and the theorized progenitor is a massive star producing a core-collapse explosion shock wave running into a dense circumstellar material (CSM). If the spectrum does not show narrow lines, then the sub-classification is based on the light curves, in particular the behavior of the luminosity decay. If the magnitude decays linearly in time (i.e. exponential decay of the luminosity), then the SN is classified as a type II_L. Instead, if there is a phase of constant luminosity (i.e. a “plateau”), the SN is classified as a type II_P. Finally, if the spectrum shows hydrogen lines only in the very early stages and then they disappear, the SN is classified as a type II_b (because of the analogy with a type Ib/Ic SN).

Except type Ia SN, all other types are thought to be the outcome of a core collapse event (e.g. [37] and references therein). The differences among core-collapse SNe are strongly correlated with the structure and (outer) composition of the progenitor, and the presence of a dense CSM. Mass loss has an important role in shaping both the CSM and the outer portion of the SN-progenitor structures and compositions, see §1.4. The main motivation of this work is to understand how mass loss changes the stellar structure, and in perspective, how this can influence the SN outcome.

1.3 Current Challenges in the Numerical Simulation of Massive Stars

From the point of view of numerical simulations, massive stars are especially difficult. In fact, on top of the problems common to the evolution of any star (e.g. uncertainties in the physics of mixing, limitations due to the assumptions of spherical symmetry), [38], massive stars pose also severe numerical challenges because they experience dynamically unstable phases during their evolution, and because of complexity and very short timescale of the very last evolutionary stages (namely, neon, oxygen and silicon burning). In the following sections, I summarize the most severe problems that can be encountered in the simulation of massive stars.

1.3.1 Nearly Super-Eddington Radiation Dominated Convective Envelopes

Stars with $M_{\text{ZAMS}} \gtrsim 20M_{\odot}$ develop extended convective envelopes during their evolution, which are radiation dominated,

$$\beta \stackrel{\text{def}}{=} \frac{P_{\text{gas}}}{P_{\text{tot}}} \lesssim 0.5 \Leftrightarrow P_{\text{gas}} \lesssim P_{\text{rad}} , \quad (1.12)$$

where $P_{\text{tot}} \stackrel{\text{def}}{=} P_{\text{gas}} + P_{\text{rad}}$ is the total pressure, given by the sum of the gas pressure P_{gas} and the radiation pressure P_{rad} . The exact moment when radiation pressure starts to dominate in the stellar envelope depends on the initial mass of the star: for $M_{\text{ZAMS}} \sim 20M_{\odot}$ this happens during the RSG phase, but stars of higher initial mass (e.g. $M_{\text{ZAMS}} \gtrsim 70M_{\odot}$) can have radiation-pressure dominated envelopes starting at ZAMS and may develop the instability discussed below early in their evolution [39].

Moreover, the luminosity in the envelope can approach (and exceed) the local “modified” Eddington luminosity⁶, say

$$L(r) \gtrsim L_{\text{Edd}} , \quad (1.13)$$

where

$$L_{\text{Edd}} \equiv L_{\text{Edd}}(r) \stackrel{\text{def}}{=} \frac{4\pi GM(r)c}{\kappa(r)} . \quad (1.14)$$

This can happen, for example, if the local value of the opacity $\kappa(r)$ increases, lowering the local modified Eddington luminosity. This is common in regions where $\log_{10}(T/[K]) \sim 5.3$

⁶The L_{Edd} defined in Eq. 1.14 is “modified” since it depends on the local opacity $\kappa(r)$. The Eddington luminosity is usually defined using the Thomson scattering opacity κ_e , which is a lower limit on the total opacity, since line processes are resonant. Therefore, when defined using κ_e , the Eddington luminosity provides a global upper-limit to the luminosity a star can have while maintaining hydrostatic equilibrium.

or $\log_{10}(T/[K]) \sim 6.2$, where the recombination of iron causes a local increase of κ (the so-called “iron opacity bump”), see Fig. 1.7.

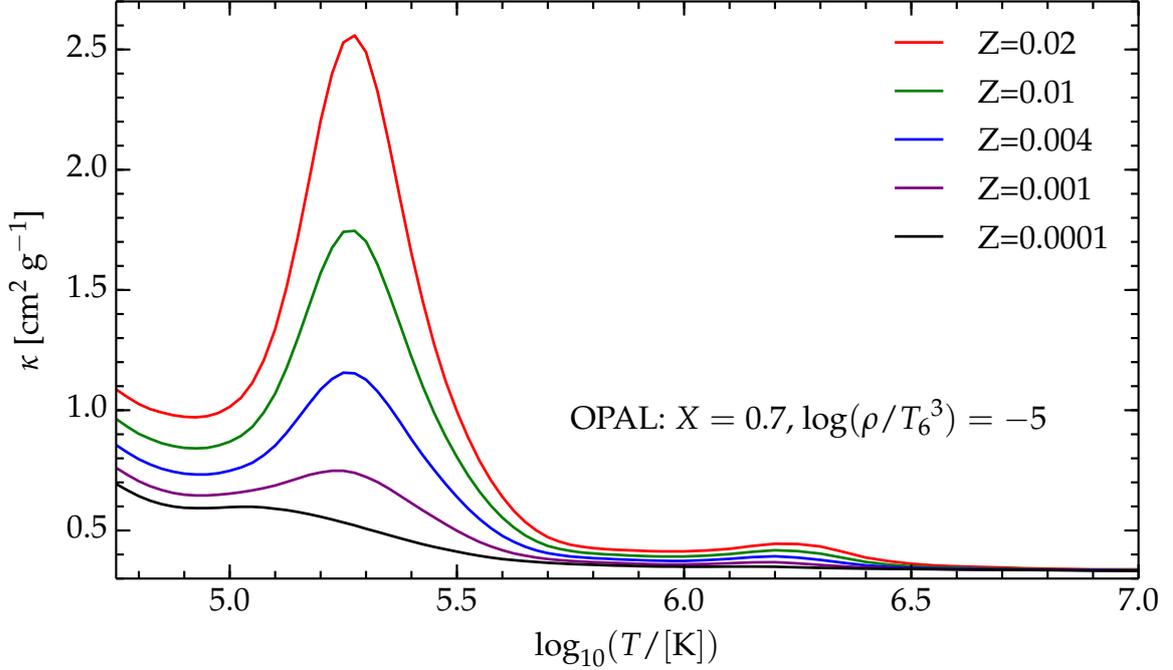


Figure 1.7: Iron opacity bump for different values of metallicity. The data are from the OPAL tables, for hydrogen abundance $X = 0.7$. T_6 is the temperature in units of 10^6 K, and ρ the density. This figure is inspired by Fig. 38 of [39].

When both the conditions of Eq. 1.12 and Eq. 1.13 are fulfilled, the envelope becomes numerically and dynamically unstable (a least in one-dimensional – 1D – numerical simulations) [39]. In fact, combining the equation for hydrostatic equilibrium,

$$\frac{dP_{\text{tot}}}{dr} = -\frac{GM(r)\rho(r)}{r^2}, \quad (1.15)$$

with the equation for the radiative pressure gradient,

$$\frac{dP_{\text{rad}}}{dr} = -\frac{\kappa\rho}{c} \frac{L_{\text{rad}}}{4\pi r^2}, \quad (1.16)$$

where L_{rad} is the radiative luminosity, it is easily seen (using also the definition of the Eddington luminosity, Eq. 1.14) that

$$\frac{dP_{\text{tot}}}{dP_{\text{rad}}} = \frac{L_{\text{Edd}}}{L_{\text{rad}}}. \quad (1.17)$$

Thus, using $P_{\text{gas}} = P_{\text{tot}} - P_{\text{rad}}$,

$$\frac{dP_{\text{gas}}}{dr} = \frac{dP_{\text{rad}}}{dr} \left[\frac{L_{\text{Edd}}}{L_{\text{rad}}} - 1 \right], \quad (1.18)$$

from which it is clear that, when $L_{\text{rad}} \rightarrow L_{\text{Edd}}$, a gas pressure inversion occurs [39, 40]. However, in non-degenerate environments, the total luminosity is the sum of the radiative plus the convective luminosity $L \equiv L_{\text{tot}} = L_{\text{rad}} + L_{\text{conv}}$, therefore $L_{\text{tot}} \geq L_{\text{Edd}}$ does not imply

$L_{\text{rad}} > L_{\text{Edd}}$. A condition for a gas density inversion can be derived writing the EOS in the form $P_{\text{gas}} \equiv P_{\text{gas}}(\rho, P_{\text{rad}})$. Therefore the density gradient can be written as:

$$\frac{d\rho}{dr} = \frac{\frac{dP_{\text{gas}}}{dr} - \frac{\partial P_{\text{gas}}}{\partial P_{\text{rad}}} \frac{dP_{\text{rad}}}{dr}}{\frac{\partial P_{\text{gas}}}{\partial \rho}} = \frac{dP_{\text{rad}}}{dr} \left[\frac{L_{\text{Edd}}}{L_{\text{rad}}} - 1 - \frac{\partial P_{\text{g}}}{\partial P_{\text{rad}}} \right], \quad (1.19)$$

where I used Eq. 1.18 in the last step. Therefore, the sign of $d\rho/dr$ is determined by the term in square brackets in Eq. 1.19 and it is positive (gas density inversion) when

$$L_{\text{rad}} \geq L_{\text{inv}} \stackrel{\text{def}}{=} L_{\text{Edd}} \left[1 + \frac{\partial P_{\text{gas}}}{\partial P_{\text{rad}}} \right]^{-1}, \quad (1.20)$$

see [39, 40]. Note that $L_{\text{inv}} < L_{\text{Edd}}$. Since T decreases outwards, if ρ is constant or even increases, there must be a strong superadiabaticity⁷, therefore, the region is necessarily convective. Convection must set in before the condition of Eq. 1.20 is reached, and, if it is efficient, it carries a large fraction of the energy flux, preventing $L_{\text{rad}} \rightarrow L_{\text{Edd}}$ [40]. However, if convection is inefficient density and pressure inversion occurs, see Fig. 1.8.

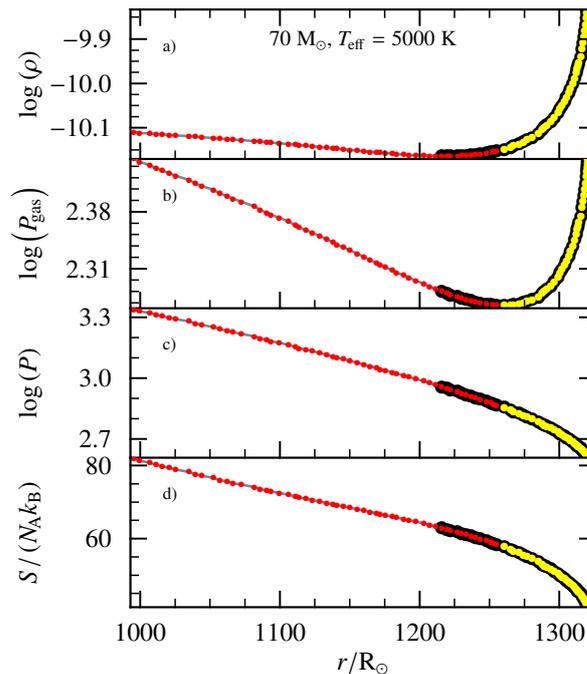


Figure 1.8: From top to bottom: density, gas pressure, total pressure ($P \equiv P_{\text{tot}}$) and specific entropy profiles in the outer envelope of a $M_{\text{ZAMS}} = 70M_{\odot}$ MESA (§2.1) model at the first crossing of the Hertzsprung gap (i.e. $T_{\text{eff}} = 5000$ K). The gas pressure and density inversion is clearly visible in the two top panels, small red dots indicate convective regions with no inversion, large red dots with black borders indicate a predicted density inversion but no gas pressure inversion, yellow dots indicate gas pressure and density inversion. This figure is Fig. 40 of [39].

⁷The superadiabaticity is defined as the difference between the local temperature gradient $d \log(T)/d \log(P)$ and the adiabatic temperature gradient $\partial \log(T)/\partial \log(P)|_s$. A strong superadiabaticity means the local temperature gradient is steeper than the adiabatic gradient.

The situation described by Eq. 1.18 (with $L_{\text{rad}} \rightarrow L_{\text{Edd}}$), Eq. 1.20 and Fig. 1.8 is unstable both numerically and physically. From the numerical point of view, 1D hydrostatic stellar evolution codes will try to take very small timesteps to try to follow what is more likely a dynamical phase of evolution. From the physical point of view, this situation is clearly dynamically unstable because of the density inversion, but the physical mechanism possibly preventing its onset, or the nature of the instability that may develop are not yet understood. It is possible that a multidimensional treatment of convection could prevent such an instability: if the convective flux is not limited to

$$F_{\text{conv}} \lesssim \rho c_s^3 , \quad (1.21)$$

as it is in the one-dimensional paradigm of Mixing Length Theory for subsonic convection (e.g. [23] and references therein), it may be able to prevent the formation of super-Eddington layers in the star by supporting a larger flux than in 1D simulations⁸. Another possibility is that a mechanism other than convection carries the energy away. For example, “photon bubbles” have been proposed in the literature [41, 42] as a way to bypass the Eddington limit. Photon bubbles could create preferential pathways for the photons [41, 42] and provide the missing flux. The mechanism of photon bubbles is thought in analogy with what happens when a fluid is forced through a bed of solid particles. The fluid can (under sufficient pressure and with the appropriate fluid-to-bed-particles density ratio) push the solid particles, force itself in between and create bubbles in the solid bed, through which it is easier to flow. In principle, if the radiation pressure is high, photons could do the same (acting as the fluid), pushing the stellar gas into over-dense and opaque clumps that absorbs photons (possibly resulting in a radiatively-driven acceleration above the escape velocity, and therefore mass loss), and creating voids through which (most of the) photons can stream away almost freely (see [42]). It is also possible that when nearly super-Eddington, radiation dominated envelopes form, they trigger mass loss either in eruptive events or as very dense (continuum-driven) wind outflows [7, 43].

1.3.2 Silicon Burning

Another challenge for evolutionary codes is the simulation of stages of advanced nuclear processing, especially silicon burning. This nuclear burning process produces as ashes all the elements of the so-called “iron group” (see Fig. 1.3), and happens with central temperature and density of [9, 12]

$$T \sim (3 - 5) \times 10^9 \text{ [K]} , \quad \rho \sim 10^7 - 10^{10} \text{ [g cm}^{-3}\text{]} . \quad (1.22)$$

This stage duration is very short (cf. Tab. 1.1), because the energy yield of silicon burning is only of order $0.1 \text{ MeV nucleon}^{-1}$ [9]. Therefore the rates of the thermonuclear reactions must be very high, and the fuel is exhausted rapidly. At this stage, the stellar core is composed mainly of silicon nuclei, which start photo-disintegrate

$$\gamma + (A, Z) \rightarrow (A', Z') + \{p, n, \alpha\} . \quad (1.23)$$

This process produces lighter particles (i.e. protons, neutrons, α s, etc...), which are then captured by the remaining nuclei to build heavier (and unstable) nuclei of the iron group

$$\{p, n, \alpha\} + \{(A, Z), (A', Z')\} \rightarrow \{\text{Fe group nuclei}\} + \dots . \quad (1.24)$$

⁸Keeping $L_{\text{rad}} = F_{\text{rad}} 4\pi r^2$ lower than the thresholds in Eq. 1.20 and Eq. 1.18.

Note that the iron group nuclei produced can be photo-disintegrated too,

$$\gamma + \{\text{Fe group nuclei}\} \rightarrow \{(A, Z), (A', Z')\} + \{p, n, \alpha\} , \quad (1.25)$$

and Eq. 1.23 and Eq. 1.25 nearly balance each other. Many (A', Z') nuclei produced by photodisintegration and particle captures are extremely neutron or proton rich, therefore a large number of weak reaction such as β^\pm -decays and electron captures⁹ play an important role. Moreover, these weak reactions are paramount in the determination of the value of Y_e in the core, which enters quadratically into the Chandrasekhar mass, Eq. 1.7. Another complication arises because of non-nuclear weak reactions, i.e. the important production of thermal neutrinos that carry away energy and entropy. This neutrino cooling interacts in a complicate way with convection [8], tailoring the extent of convective shells and thus the final structure of the inner portion of the star, [19].

Despite the challenges posed by silicon burning, because of its importance in the determination of the structure and composition of the iron core, it needs to be followed in great detail in order to produce pre-SN structures as reliable initial conditions for the further evolution in core-collapse SN.

Because of the large number ($\gtrsim 200$) of isotopes involved in the processes outlined by Eq. 1.23–1.25, and because of the extremely high rates of the reactions¹⁰, silicon burning is very difficult from the computational point of view. Often, physical approximations are required (e.g. Quasi Statistical Equilibrium – QSE – see for example [3, 12]), or the numerical simulations are stopped at earlier stages, before silicon ignition (e.g. [44, 45]).

1.3.3 Lack of Systematic Studies to Quantify the Many Uncertainties

Because of the problems mentioned in §1.3.1 and §1.3.2, the numerical simulation of a massive star from ZAMS to the onset of core-collapse is a time-consuming task, even with 1D codes. This directly translates to a lack of systematic studies that quantify the uncertainties associated with the large number of free or poorly constrained parameters (e.g. mixing length, some nuclear reaction rates, mass loss algorithm and efficiency, etc...).

Most of the stellar astrophysics community dealing with the evolution of massive stars seems to trust fiducial sets of parameters adopted. Often, the many parameters necessary to carry out a simulation are tweaked in order to find an overall agreement with observations (e.g. the mass loss rate, see [7]). The lack of a quantified systematic error in stellar simulations significantly complicates the comparison with observed phenomena [7, 38].

Moreover, it is necessary to stress that many physical processes involved in the evolution of stars are not completely understood (e.g. mass loss, see §1.4) or they are simulated with simplified parametric algorithms (e.g. Mixing Length Theory for convection). The many free parameters present in stellar evolutionary codes can, in principle, permit the reproduction of a large variety of observed phenomena without an accurate reproduction of the underlying physics.

This calls for studies aiming to quantify the systematic errors associated with the many assumptions commonly made in numerical simulations of massive stars. The present study aims to understand and constrain the systematic uncertainty associated with mass loss from

⁹Positron captures are always negligible for stars with $M_{\text{ZAMS}} \leq 40M_\odot$ [9].

¹⁰Near equilibrium, nearly balancing forward and backward reactions with extremely high rates create numerical round-off error problems.

massive stars, by comparing the effects of various mass loss algorithms commonly used in massive star evolutionary calculations (see §1.4). Moreover, I attempt a numerical experiment to explore the effects of violent and short-lasting mass loss events (regardless of the mechanism triggering it) on the stellar structure (see §1.4.9 and §2.4).

1.4 Massive Star Mass Loss

Mass loss is one of the most important phenomena in massive star evolution. It is also a channel by which massive stars affect their environment. For the star, mass loss reduces its total mass and alters the star's evolutionary timescales (e.g. [46]), especially the time spent on the RSG branch of the HR diagram. Moreover, mass loss is necessary to explain the variety of core-collapse SN event (see §1.2.5 and e.g. [7, 45, 47, 48]).

Despite its centrality, mass loss is not fully understood. It is presently one of principal sources of uncertainty in massive star evolution [7]. There are three main modes of mass loss:

- Radiatively driven stellar winds;
- Episodic and/or eruptive mass loss (e.g. wave driven, pulsational instabilities or giant eruptions from Luminous Blue Variables - LBVs, [6, 7, 49, 50]);
- Roche lobe overflow (RLOF) in binary systems.

Which of these processes determines the total amount of mass shed has been a matter of intense debate in the literature (see [7]), but it seems well established that for the more massive stars (e.g., the ones that become LBVs), the bulk of the mass must be lost in violent and brief events rather than in long-lasting steady winds[7]. Moreover, because of the large fraction of massive stars in close¹¹ binary systems [20], RLOF must be involved in extracting mass from these stars. I describe the simplified simulation of a violent and short mass loss event in §2.4, and the corresponding result in §4.

Since stellar evolution codes are usually one-dimensional and hydrostatic, i.e. they assume spherical symmetry and do not solve time dependent equations of motion for the matter, it is hard to include mass loss in a physical and self-consistent way. Observed mass outflows are non-spherical and their asphericity could play a role in the mass loss dynamics. For these technical reasons, the intrinsically dynamical and fast eruptive, explosive and/or episodic mass loss is completely neglected in single star evolutionary models¹²: stellar evolution codes cannot compute the response of the structure to the mass removal if it happens on a dynamical timescale.

Stellar winds are also dynamical (see §1.4.1), however, they are characterized by a much smaller mass loss rate ($\dot{M} \sim 10^{-9} - 10^{-4} M_{\odot} \text{ yr}^{-1}$, largely depending on the evolutionary stage of the star) than the eruptive and/or RLOF events ($\dot{M} \gtrsim 10^{-4} - 10^{-1} M_{\odot} \text{ yr}^{-1}$ [7]), and their duration is of course much longer than that of the episodic events. Moreover, the characteristic timescale is shorter than the thermal timescale of the star. Thus, it is easier to average wind mass loss in time: the assumption of steadiness permits including it in stellar evolutionary codes with a parametric algorithm that provides a value of \dot{M} . This, multiplied

¹¹That is, the star interacts with its companion before the end of the globally hydrostatic evolution.

¹²Some authors, e.g. [46], include it by artificially enhancing the mass loss rate in order to reproduce the total amount of mass expected to be shed by the star in episodic events.

by the timestep, gives the total amount of mass to be removed from the (outer) envelope at each time integration step. The amount of mass to be removed is determined at the point where the gas reaches the speed of sound (see below), therefore, regardless of the dynamics happening in the outermost layers, mass can be removed from the outer portion of the star. Note that the entire stellar structure re-adjusts to mass loss at each time step, because of the changes in the boundary conditions.

Underlying the available parametric algorithms there is also the assumption of spherical symmetry, which necessarily breaks when rotation and/or magnetic fields are required (although some codes include also parametric enhancements of mass loss because of the centrifugal force lowering the effective gravity g_{eff} , see [39]).

Note that even stellar evolution codes including the time-dependent hydrodynamical equations need parametric algorithms for the winds, since they do not compute the radiative acceleration, (see below), and neglect the episodic mass loss. This is partly because the physical mechanism triggering these events is yet unknown, and partly because stellar evolution codes use large timesteps that cannot properly resolve short eruptions.

Note also that none of the algorithms available to date self-consistently compute the velocity structure $v(r)$ in the wind, the radiative acceleration and the mass loss rate. These quantities are inter-dependent (see below), and $v(r)$, together with the ionic populations, enters into the observational diagnostics of the mass loss rate. The self-consistent computation of the wind velocity structure, the radiative acceleration and the mass loss rate is a very challenging task because of the high non-linearity of the problem.

1.4.1 Outline of the Theory of Stellar Winds

Because an algorithm that gives a time averaged \dot{M} over a given numerical timestep is needed, the community has focused on steady line-driven stellar winds [7]. In line-driven winds, momentum is resonantly transferred from photons to the gas via absorption and line scattering (i.e. bound-bound transitions). The basic idea is that each incoming photon has a well defined direction and excites an atom or ion which then de-excites via isotropic emission. This results in the transfer of momentum,

$$\Delta p = \frac{h}{c}(v_i \cos(\theta_i) - v_f \cos(\theta_f)) , \quad (1.26)$$

in the radial direction to the atom/ion, where the subscript i indicates the quantities of the incoming photon and the subscript f those of the outgoing photon produced by the de-excitation [49]. When integrating over all directions, the effects of all the de-excitation photons average out, and the net result is a radially directed acceleration. This is possible because the radiation field is not isotropic in stellar atmospheres¹³ (otherwise the effects of the incoming photons would also average to zero). Atoms with a large number of lines, i.e. metals, are paramount for the momentum transfer, since they provide high opacities κ despite their low abundance. This is why line-driven winds should be metallicity dependent. The collisional (Coulomb) coupling [49] redistributes the momentum that metals acquire from photons among all the species. This requires a density high enough to yield a collisional rate higher than the absorption rate.

Note also that in this picture the wind is expected to be inhomogeneous [7, 49]. In fact, the acceleration of a gas parcel at radial distance r from the center of the star, caused by a

¹³The anisotropy of the radiation field can be considered as the definition of “stellar atmosphere”.

single line-absorption, is the amount of momentum lost by the radiation field per unit time [51, Ch. 8],

$$g_{\text{rad},\nu}^{\text{line}} = \left(\frac{L_\nu}{4\pi r^2} \right) \left(\frac{\kappa_\nu \rho}{\tau_\nu} \right) (1 - e^{-\tau_\nu}) \frac{1}{c\rho} , \quad (1.27)$$

where the first term in brackets is the monochromatic flux from the star (approximated as a point-like source), the second term corresponds in the Sobolev approximation (see below) to

$$\frac{\kappa_\nu \rho}{\tau_\nu} = \frac{\nu}{c} \frac{dv}{dr} , \quad (1.28)$$

where τ_ν is the (specific) optical depth at frequency ν (see definition below, Eq. 1.31). Eq. 1.28 defines the width of the absorption band. The third term in brackets of Eq. 1.27 is the probability for a photon to reach the point r in the wind without being absorbed in the innermost region. Thus, the product of the three terms in brackets on the right-hand side of Eq. 1.27 is the energy absorbed per unit time and per unit volume by the gas at distance r from the center of the star. Dividing it by c , it becomes the momentum per unit time and per unit volume, and dividing again by ρ , it becomes the acceleration per unit volume due to the line absorption. The only subtle point here is the approximation needed to write down Eq. 1.28, i.e. the so-called Sobolev approximation. This corresponds to the assumption of a Dirac δ profile for the intrinsic profile of the absorbing line, (or, in other words, it corresponds to neglecting the intrinsic width of the line with respect to the Doppler shifts, see below)

$$\phi(\nu - \nu_0) = \delta(\nu - \nu_0) , \quad (1.29)$$

so that the radiative acceleration depends only on the incoming flux and local quantities at the point where it is absorbed. This assumption is reasonable whenever all quantities can be considered constant over the so-called Sobolev length,

$$L_s \stackrel{\text{def}}{=} \frac{v_{\text{th}}}{\frac{dv}{dr}} , \quad (1.30)$$

where the thermal velocity v_{th} enters because thermal line-broadening causes departures from the δ -function profile since the lines are Doppler shifted in random directions because of the thermal agitation of the atoms, and the velocity gradient appears because of the Doppler shift (see below). In order to derive Eq. 1.28, consider the (specific) optical depth definition [51, Ch. 8]

$$\tau_\nu(r) \stackrel{\text{def}}{=} \int_r^\infty \kappa_{\nu'} \rho(r') dr' \equiv \frac{\pi e^2}{m_e c} f_l \int_r^\infty n_l(r') \left(1 - \frac{n_u(r') g_u}{n_l(r') g_l} \right) \phi(\nu' - \nu_0) dr' , \quad (1.31)$$

where the specific absorption coefficient κ_ν is expressed using the oscillator strength f_l , the population of the upper (lower) level of the considered transition $n_{u(l)}$, and its degeneracy $g_{u(l)}$, and $\nu' \equiv \nu'(r')$ is the frequency ν as seen from the point r' . Using Eq. 1.29 and the Doppler shift relation,

$$\nu' - \nu_0 = \nu \left(1 - \frac{v(r')}{c} \right) - \nu_0 , \quad (1.32)$$

where ν is the frequency emitted at the photosphere, ν_0 is the central frequency of the line and v is the velocity in the radial direction, we can carry out the integration in Eq. 1.31 and get the so-called Sobolev optical depth

$$\tau_\nu = \kappa_\nu \rho \left(\frac{dv}{dr} \right)^{-1} \Big|_r \frac{c}{\nu} , \quad (1.33)$$

where the velocity gradient appears when integrating the Dirac δ with the right hand side of Eq. 1.32 as the argument. Eq. 1.33 is equivalent to Eq. 1.28. Note that the Doppler shift is paramount to obtain Eq. 1.33, and, more in general, for the line driven radiation acceleration. First of all, it allows to photons that do not interact with the inner layer of the atmosphere, to give a contribution in the outer portion of it. Secondly, it allows for the de-saturation of spectral line. And finally, when the atomic/ionic species approach the terminal velocity of the wind, the Doppler effect shifts the spectral lines out of the frequency range in which the driving is most effective.

By the substitution of Eq. 1.28 into Eq. 1.27 it is easily seen that the single line-absorption radiative acceleration is proportional to the velocity gradient,

$$g_{\text{rad},\nu}^{\text{line}} = \left(\frac{L_\nu}{4\pi r^2 c^2 \rho} \right) \nu (1 - e^{-\tau_\nu}) \frac{dv}{dr} \Rightarrow g_{\text{rad},\nu}^{\text{line}} \propto \frac{dv}{dr}, \quad (1.34)$$

therefore, line driven acceleration tends to separate the gas into “clumps”, because the acceleration of a parcel increases the velocity gradient dv/dr , causing an increase of the acceleration of the parcel itself. This mechanism creates a differential acceleration among initially adjacent gas parcels and can produce shocks and therefore inhomogeneities in the wind. It has been proposed as a cause of the observed over-dense regions in stellar winds¹⁴ (the so-called “clumps”, see [7, 49] and references therein for a review of the observational signature of the presence of overdense clumps).

Note that Eq. 1.34 is valid for a differential band-width around the central frequency ν . Stellar evolution codes do not compute the spectra of the simulated stars, therefore the dependence of the radiative acceleration (integrated over all frequencies) on the outgoing radiation is parametrized in terms of the total photon flux, expressed as a function of the effective temperature T_{eff} and/or the bolometric luminosity L .

Most mass loss algorithms currently available are not derived from first principles, but are rather semi-empirical formulae based on observations. Therefore many do not include an explicit metallicity dependence, or just assume a smooth scaling with Z ¹⁵,

$$\dot{M} \propto \left(\frac{Z}{Z_\odot} \right)^x, \quad (1.35)$$

with $x \sim 0.5$ [14, 25], which arises from the combination of the metallicity dependence of the number of photon interactions driving the wind, and the metallicity dependence of the wind terminal velocity (see also §1.4.5). Note that deviations from Eq. 1.35 are expected at very high Z (once the metal lines driving the wind saturate, i.e. absorb all the relevant photons, there cannot be further acceleration) and at very low Z (because of the lack of metals and thus lines) [25]. Deviations can occur also whenever the chemical composition does not scale with the solar metallicity (i.e. when there are differential variations among the abundances of different elements). This could happen in stars born in an environment very different from the pre-solar nebula, or whenever mixing processes (e.g. rotational mixing [52, 53]) lift large amounts of metals to the stellar surface.

Winds are theoretically expected, and also observed, to be variable both in space (e.g. because of rotation or clumpiness) and time [49], but all the available algorithms assume

¹⁴Note, however, that this may not be the only mechanism driving the formation of clumps, and other unknown physical processes may also result in inhomogeneities in the wind structure.

¹⁵However, few exceptions exists, such as [25], §1.4.5.

steady state and homogeneity [7]. On the one hand, this is the reason why they are used in stellar evolution codes, but on the other hand this introduces poorly controllable systematic errors¹⁶. In fact, recent work (e.g. [7, 49]) suggests that the current predictions for mass loss rates are overestimated by a factor between 2 to 10, and suggests 3 as the most realistic over-estimation factor [7]. This over-estimation can arise whenever the mass loss rate is inferred from the continuity equation,

$$\dot{M} = 4\pi r^2 \rho v(r) , \quad (1.36)$$

assuming that $\rho \equiv \rho(r)$ is constant in each spherical shell of given r . Instead, theoretical and observational evidences of small-scale inhomogeneities (at constant r) are available (e.g. [7, 49] and references therein). Many observed spectral features are dominated by the over-dense “clumps” of matter. Therefore, the density inferred from observation is the density of the “clumps”, which is higher than the mean density of the wind, and if this “observed” density is used in Eq. 1.36, the resulting mass loss rate obtained is higher than it should be. Note, however, that if the radiative acceleration of clumps is extremely efficient, inhomogeneities could also, in principle, increase the mass loss rate. To quantify how much the mass loss rates are over-estimated, it is standard practice to define a “clumping factor” [7, 49],

$$f_{cl} \stackrel{\text{def}}{=} \frac{\langle \rho^2 \rangle}{\langle \rho \rangle^2} , \quad (1.37)$$

where $\langle \rangle$ indicates the average over the wind structure. The numerator can be inferred from observational diagnostics which depend on ρ^2 , i.e. recombination emission such as $H\alpha$, radio or infrared excess. The denominator instead can be inferred from diagnostics which are linearly proportional to ρ , such as resonant P Cygni line profiles. See §1.4.2 for a short review of the observational methods, or [51, Ch. 2] and [49] and references therein. If f_{cl} is known, then the efficiency of the wind rates inferred from the first kind of observational diagnostics can be reduced by a factor $\eta = \sqrt{f_{cl}}$, [7].

Figure 1.9 illustrates the mass loss history of a $15M_{\odot}$ solar metallicity MESA (see §2.1) model, as an example of the current results from numerical simulations. The wind algorithm used is a combination of the Vink *et al.* (§1.4.5), de Jager *et al.* (§1.4.2) and Nugis & Lamers (§1.4.7) algorithms¹⁷. In the stellar model shown in Fig. 1.9, the mass loss rate has been reduced by a factor $\eta = 0.33$, to account for wind clumpiness. Note, however, that this picture may still be far from the mass loss history of real stars.

As Fig. 1.9 illustrates, a typical massive star will lose only a small fraction of its mass during the main sequence, the mass lost until core H depletion for a $M_{\text{ZAMS}} = 15M_{\odot}$ is $\lesssim 0.2M_{\odot}$, even if the main sequence is the longest phase of a star’s lifetime. The wind becomes stronger during the supergiant phase, when the stellar radius increases dramatically, the atmosphere cools down and dust grains may form through nucleation. During this evolutionary phase, the wind could be dust-driven instead of line-driven, i.e. the radiation field may transfer momentum primarily to dust grains instead of metal atoms/ions (see also §1.4.4). Collisional coupling between dust and the surrounding gas redistributes momentum and drives mass loss, as in the line-driven case. The qualitative behavior of the $M(t)$ function is similar for all massive stars. However, for higher initial masses, the luminosity will be higher, the radius larger and the surface cooler, so the wind mass loss rate will be

¹⁶Note that the semi-empirical nature of these algorithms is largely responsible for this. In fact, also the empirical determination of the mass loss rate of observed stars is largely uncertain, see below, and, e.g., [7].

¹⁷The last algorithm is for to Wolf-Rayet (WR, see §2.2.2 for the definition) stars and is not used for the $15M_{\odot}$ star shown in Fig. 1.9.

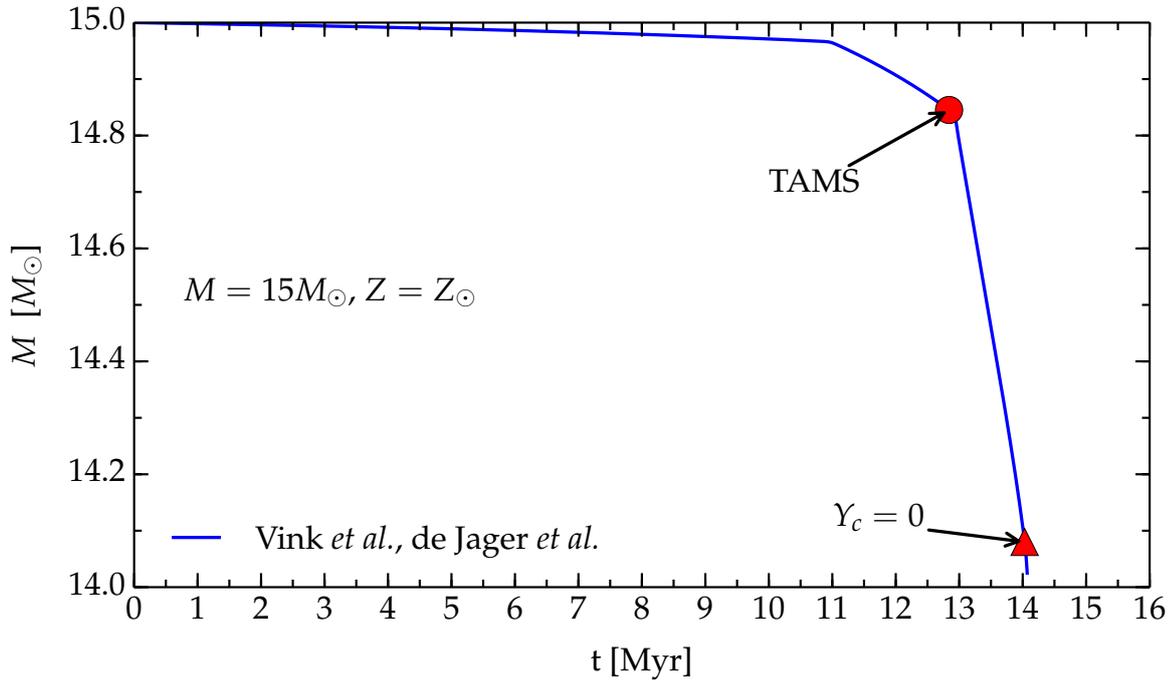


Figure 1.9: Mass as a function of time for a $M_{\text{ZAMS}} = 15M_{\odot}$, Z_{\odot} star, computed with MESA (§2.1, release version 6794) using the wind mass loss algorithm “Dutch” (see legend and text) with an efficiency $\eta = 0.33$. The model is evolved from the zero age main sequence (ZAMS) to the onset of core-collapse. The red circle indicates the TAMS, i.e. central H depletion ($X_c < 0.01$), and the red triangle indicates the end of core He burning, i.e. $Y_c < 0.01$.

much higher. Thus, more massive stars lose a larger fraction of their mass during their globally hydrostatic evolution, possibly reaching the onset of collapse with a lower total mass. Note that in Fig. 1.9 only wind mass loss is considered (and even with reduced efficiency).

1.4.2 de Jager *et al.*

The de Jager wind scheme [54] is an empirical relationship of the form $\dot{M} \equiv \dot{M}(T_{\text{eff}}, L)$. This choice of variables has the advantage of using only observable quantities, making it easy to track the mass loss rate while the star climbs on the HR diagram. The drawback is that this approach does not incorporate direct information about the physical mechanism driving the wind.

Note that T_{eff} does not have a direct physical meaning. It is defined by the black body relation,

$$L = 4\pi R^2 \sigma T_{\text{eff}}^4, \quad (1.38)$$

where σ is the Stefan-Boltzmann constant. Therefore, for any L , higher T_{eff} corresponds to smaller radii and higher surface gravity: the mass loss rate is expected to be lower at higher T_{eff} , which may not seem intuitive at first sight.

To obtain a reliable mass loss algorithm, de Jager *et al.* [54] proceed as follows. They collect from the literature the observed mass loss rates of 271 stars, nearly all of population I and spectral types from O to M. These mass loss rates are determined using several different methods based on the analysis of different portions of the stellar spectrum. For the sake of completeness, I include here a brief description of these methods (the interested reader is

referred to [51, Ch. 2] and [49, 55] for more details).

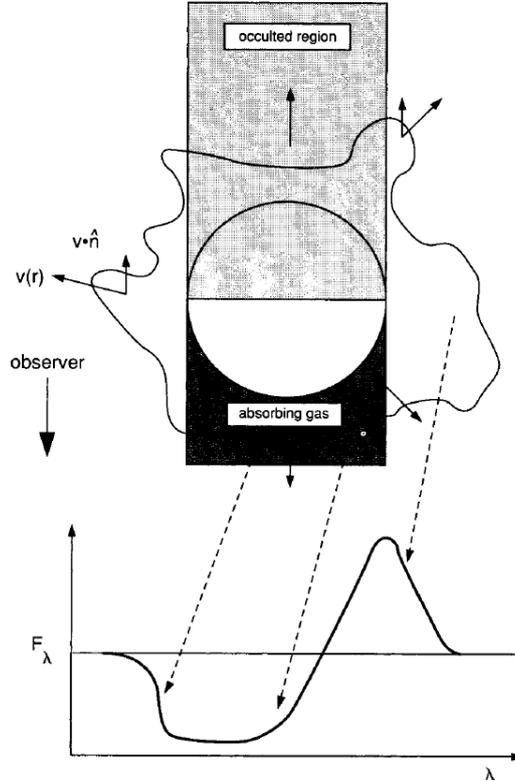


Figure 1.10: Schematic representation of the formation of a P Cygni profile in a stellar wind. The dark region is moving toward the observer, causing the bulk of the blue-shifted absorption, as the dashed arrows indicate. The gas moving in all other directions contributes to the red-shifted emission. The gray region is occulted by the mass-losing star, and thus the gas in that region is not accessible to observations. This figure is Fig. 10.4 of [56].

- **Ultraviolet (UV) spectra, mainly focusing on UV resonance line profiles - the so-called P Cygni profiles¹⁸.** These profiles (see Fig. 1.10) are characterized by a red-shifted emission component and a blue-shifted absorption component. The red-shifted component is caused by emission from outflowing gas moving away from the observer [51, Ch. 8], while the blue shifted component is caused by the absorption in the gas moving toward the observer along the line of sight. Scattering of photons into the line of sight from matter moving in other directions contributes to broaden both components, adding contributions from the gas not moving along the line of sight. To deduce the mass loss rate one needs to fit the observed profiles varying the spatial distribution of scatterers $n_i(r)$ in the wind, which in turn can be translated in a density distribution $\rho(r)$. Then \dot{M} can be evaluated using the continuity equation, Eq. 1.36, and an assumption, or an independent measurement of v . The disadvantage of this method is that it requires assuming the chemical composition and degree of ionization throughout the wind. But it has the advantage of being only weakly influenced by the wind clumpiness (compared to other methods that depend on higher powers of

¹⁸P Cygni profiles do not need to be in the UV range. However, the most relevant P Cygni profiles for the mass loss determinations are in the UV spectra of stars.

the density) since the physical process for line formation is scattering¹⁹, for which the line strength is only linear in the density ρ [7, 55]. Thus, the fit of P Cygni profile lines provides a good estimate of the mass loss rate even if the wind is clumpy;

- **Spectral lines in the optical and near UV, such as H α .** These are mostly emission lines due to recombination processes [51, Ch. 2], therefore their emissivity is proportional to ρ^2 and they are strongly affected by clumpiness. Since the recombination rate is also a function of the temperature, the emissivity is $j_\nu = \rho^2 f(T)$, with f some known function of the temperature. Usually, the wind is optically thin to photons emitted via recombination because of the Doppler shift: the outermost matter is faster and will not absorb the photons produced in the inner regions. In the optically thin limit, if the distance to the star is known, the mass loss rate can be inferred from the total line luminosity L_l , i.e. the energy emitted by the star per unit time in that line.

$$L_l = \int_R^\infty j_\nu 4\pi r^2 W(r) dr = \int_R^\infty \rho^2 f(T) 4\pi r^2 W(r) dr = \frac{\dot{M}^2}{4\pi} \int_R^\infty f(T) \frac{W(r)}{r^2 v^2(r)} dr, \quad (1.39)$$

where $W(r)$ is the so-called dilution factor that is used to exclude the photons emitted toward the star (which cannot reach the observer), and in the last step I use Eq. 1.36. The drawback of this method is that it requires detailed knowledge of the velocity structure $v(r)$ and the distance of the star. Analytic approximations exist for $v(r)$ (see Eq. 1.58). Moreover, because of the ρ^2 dependence, the mass loss rate inferred with this method is biased toward higher values for clumpy winds (if the data are interpreted assuming homogeneity);

- **Broadband photometric data in the infrared (IR) and radio continuum.** The presence of an ionized wind surrounding the star causes an excess in the IR and Radio continuum (see below), because of bremsstrahlung (free-free) emission in the wind [51, Ch. 8]. Thus, this excess is proportional to the density squared and if the velocity and temperature structures of the wind are known, the mass loss can be inferred from the continuity equation, Eq. 1.36. To use this method at radio wavelength, it is necessary to exclude stars whose emission is dominated by synchrotron radiation. Again, the ρ^2 dependence introduces the risk of over-estimating \dot{M} for clumpy winds if the assumption of homogeneity is used;
- **IR spectrum of molecules containing C, especially CO** [51, Ch. 8]. This method is used to determine the mass loss rate of cool stars with very extended winds. The CO molecule is so stable that it forms very deep in the wind (i.e. very close to the photosphere), and it is found everywhere in the wind. Therefore CO is a good tracer of the wind density, from which \dot{M} can be inferred using Eq. 1.36. The interesting molecular transitions are between levels of the vibrational band (excited by IR photons) or rotational band (excited by collisions with H_2 molecules) [51, Ch. 8]. Note that collisions will push the level populations toward thermal equilibrium, while photon excitations will cause a departure from it (and in the extreme photon dominated case this will lead to masers, see below). To illustrate the mass loss determination using molecular lines, we can consider a rotational transition from $J_u \rightarrow J_l$, where the subscripts stand for “upper” and “lower” respectively. The population of the lower level n_l is proportional to ρ and can be determined using the formalism of partition functions, while the

¹⁹Note however that the ionization fraction varies as ρ^2 .

population of the upper level can be expressed using a Boltzmann-like relation,

$$n_u = \frac{g_u}{g_l} n_l e^{-\Delta E/k_b T^*} , \quad (1.40)$$

where g_u (g_l) is the degeneracy of the upper (lower) level, $\Delta E = E_u - E_l$ is their energy separation, and T^* is an effective (non-physical) temperature, which will correspond to the gas temperature only if photon excitation can be neglected, and which can even

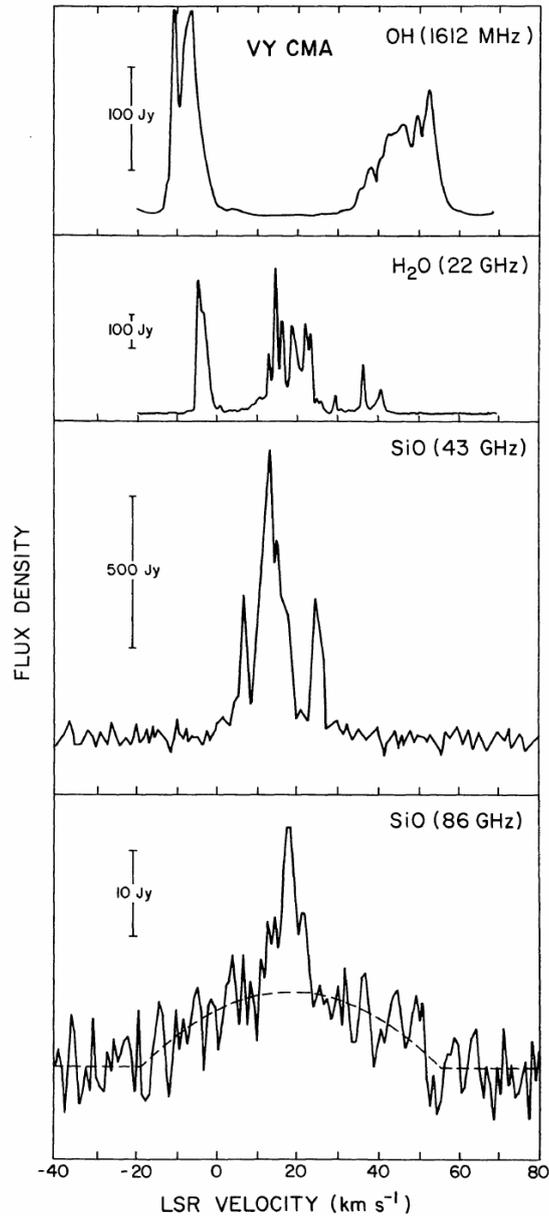


Figure 1.11: Examples of maser line profiles from VY Canis Majoris. The x axis is the “local standard of rest” (LSR) velocity. In the last panel, the dashed line indicates the thermal profile of SiO emission, whose centroid is probably the stellar velocity, and a weak maser feature appears near $\sim 18 \text{ km s}^{-1}$. This figure is Fig. 7 of [57].

become negative if a population inversion occurs. The emissivity of the line is $j_\nu = n_u A_{ul} h\nu$, where A_{ul} is the Einstein coefficient for spontaneous emission, and the line luminosity is related to j_ν (see Eq. 1.39), so, if T^* and $v(r)$ are known from independent measures, it is possible to determine ρ and thus \dot{M} from L_l . This works only if CO is optically thin, i.e. photons from these transition can reach the observer;

- **Maser lines.** This method provides \dot{M} only for low density winds, otherwise collisions will prevent the population inversion which causes the maser. To have a maser line in a spectrum, a population inversion is needed, so that the non-local thermal equilibrium (non-LTE) correction factor for the absorption coefficient $\chi = [1 - n_u g_l / n_l g_u]$ (cf. Eq. 1.40) becomes negative. This is usually caused by a “pump”, provided in stellar atmospheres by photons Doppler-shifted to the right frequency. Therefore the maser lines will have a double-peaked profile (Fig. 1.11), one peak corresponding to $-v_D$ (wind moving toward the observer) and the other to $+v_D$ (wind moving away from the observer), where v_D is the velocity needed to get the maximum pump efficiency. This means that the velocity of the wind can be inferred from the separation between the two peaks [51, Ch. 2]. Then, the density can be inferred from the amplification factor e^{τ_ν} , and therefore the mass loss rate can be calculated via Eq. 1.36.

Note that these are not the only methods available to measure the mass loss rate [51, Ch. 2]. de Jager *et al.* determine the accuracy of each mass loss estimate by comparing the results of different methods for the same star²⁰. For each star having the mass loss rate determined with at least 3 different methods, de Jager *et al.* derive a mean value of the logarithm of the mass loss rate ($\langle \log_{10}(-\dot{M}) \rangle$). Together with $\langle \log_{10}(-\dot{M}) \rangle$, they also derive the deviation $\Delta = \log_{10}(\dot{M}) - \langle \log_{10}(-\dot{M}) \rangle$ of each determination from the mean value. The authors then derive the one sigma value of the distribution of Δ , $\sigma^{\text{det}} \simeq 0.37$, and use it as the “average intrinsic error per determination”. They assign a weight to each mass loss determination based on the number of methods used to find it. Then, they determine the bolometric luminosity and effective temperature for each star, using both a weighted average between the observed value(s) and the theoretical relationship of spectral type and luminosity class as functions of T_{eff} and L .

de Jager *et al.* obtain the functional dependence $\dot{M} \equiv \dot{M}(T_{\text{eff}}, L)$ by writing $\log_{10}(-\dot{M})$ as a sum of Chebychev polynomials of the first kind $T_n(x) \stackrel{\text{def}}{=} \cos(n \arccos(x))$:

$$\log_{10}(-\dot{M}) = \sum_{n=0}^N \sum_{\substack{i=0 \\ j=n-i}}^{i=n} a_{ij} T_i(\log_{10}(T_{\text{eff}})) \cdot T_j(\log_{10}(L)) , \quad (1.41)$$

where the coefficients a_{ij} are found by fitting the available data with this formula.

What is commonly used in stellar evolution codes is the first order approximation of Eq. 1.41,

$$\log_{10}(-\dot{M}) = 1.769 \log_{10}(L/L_\odot) - 1.676 \log_{10}(T_{\text{eff}}/[K]) - 8.158 . \quad (1.42)$$

The algorithm of Eq. 1.42 is presented by de Jager *et al.* as a valid formula for the “averaged statistical behavior” of stellar winds across the entire HR diagram within an error comparable to algorithms valid only for smaller portions of the diagram. This interpretation of their result is natural, since the mass loss rate is determined as an average over a large set of stars and not from the observation of a single object. The standard deviation of

²⁰It was not their goal to determine which observational method is more reliable.

the $\log_{10}(\dot{M})$ predicted by Eq. 1.42 is $\sigma \simeq 0.45 - 0.50$, slightly larger than σ^{det} . This indicates that, even if T_{eff} and L are the most important quantities to characterize the mass loss rate, also other quantities should play a minor role. It is important to note that WR and Be stars were intentionally left out of the data set used to determine Eq. 1.42.

1.4.3 Nieuwenhuijzen *et al.*

The algorithm described in Nieuwenhuijzen *et al.* [58] is a refinement of Eq. 1.42. It is derived from the same data set, and includes the dependence of the mass loss rate on the total mass. It also translates the temperature dependence into a radius dependence. The inclusion of the total mass in the quantities determining the mass loss rate allows the authors to find a smaller standard deviation of their predicted $\log_{10}(-\dot{M})$, matching the value for the distribution of the observed mass loss rates.

However, since the mass is not a directly observable quantity for single stars, the mass determination is based on numerical simulations of stars. The theoretical models used are from Maeder & Meynet [59, 60]. An important caveat must be mentioned: the fact that different stellar evolution codes consider a large variety of physical processes (e.g. for mixing and mass loss), or just use different implementations of the same physical processes, causes a spread in the stellar masses found at the same point of each model evolution. Therefore, the Nieuwenhuijzen *et al.* mass loss algorithm depends on the validity of the set of models used. This drawback applies to all mass loss algorithms involving a functional dependence of the form $\dot{M} \equiv \dot{M}(M)$ derived from stellar evolution models. However, the mass dependence of the mass loss rate found by Nieuwenhuijzen *et al.* is rather weak (see Eq. 1.45). Some authors (e.g. [61]) suggest avoiding the model dependence issue by substituting the total mass M with a fixed value, and find that this does not significantly change their results.

Stars with different masses pass through the same point on the HR diagram at different stages of their evolution. Nieuwenhuijzen *et al.* [58] use this to build an “average expected mass” M of a star at a given (T_{eff}, L) point. This value is derived as follows. The authors define a *dwelt time* representing the time for a star to travel over a unit length track on the HR diagram, i.e. ,

$$t^{(d)} \stackrel{\text{def}}{=} \frac{\Delta t}{\sqrt{[\Delta \log_{10}(T_{\text{eff}}/[K])]^2 + [\Delta \log_{10}(L/L_{\odot})]^2}}, \quad (1.43)$$

where Δt is the time spent to travel over the $(\Delta \log_{10}(T_{\text{eff}}/[K]), \Delta \log_{10}(L/L_{\odot}))$ distance.

For every point on the HR diagram, there are leftward or rightward subtracks of the stellar evolutionary tracks crossing it. Let $t_n^{(d)}$ be the dwell time for the n -th subtrack. The average expected mass M is obtained as:

$$M = \frac{\sum_{n=1}^N \Psi(M_n) \frac{dM_n}{d \log_{10}(L)} t_n^{(d)} M_n}{\sum_{n=1}^N \Psi(M_n) \frac{dM_n}{d \log_{10}(L)} t_n^{(d)}}, \quad (1.44)$$

where Ψ is the initial mass function for stars on the subtrack considered, N is the total number of crossing, and $dM_n/d \log_{10}(L)$ is the density of tracks over a unit $\log_{10}(L)$ interval.

The authors perform a fit of the data set²¹ used in [54], adding the value of M from

²¹Which on purpose does not include WR and Be stars.

Eq. 1.44 to the set, and find the interpolation formula

$$\log_{10}(-\dot{M}) = -14.02 + 1.24 \log_{10}(L/L_{\odot}) + 0.16 \log_{10}(M/M_{\odot}) + 0.81 \log_{10}(R/R_{\odot}) . \quad (1.45)$$

The uncertainties on this mass loss rate is determined as in [54], i.e. using the standard deviation of their fit. They find $\sigma = 0.36$, which is equal to the intrinsic deviation of the individual mass loss determinations, σ^{det} (see §1.4.2).

1.4.4 van Loon *et al.*

The mass loss algorithm in van Loon *et al.* [62] is an empirical formula derived from the observation of a sample of oxygen-rich asymptotic giant branch (AGB) and red supergiant (RSG) stars in the Large Magellanic Cloud (LMC). Their analysis is based on a dust-driven wind model: AGB and RSG stars have very extended, cool and dusty envelopes.

To derive their mass loss formula, the authors first derive the spectroscopic classification of the stars in their sample. Then, they fit the IR spectrum of the observed stars with the dust radiative transfer model DUSTY [63] to deduce T_{eff} and L . Their model assumes a very simple structure of the dust grains (all identical) and dust-to-gas mass ratio obtained by scaling the value observed in solar metallicity environment to the LMC metallicity. Once T_{eff} and L are known, they fit their data to obtain the relation

$$\log_{10}(-\dot{M}) = -5.65(15) + 1.05(14) \log_{10}(L/10^4 L_{\odot}) - 6.3(1.2) \log_{10}(T_{\text{eff}}/3500 \text{ K}) . \quad (1.46)$$

In Eq. 1.46 the numbers in parenthesis indicate the estimate of the error on the last digits as reported in [62].

Although many authors use this algorithm for RSG stars (e.g. [46]), it is semi-empirically derived from a particular set of data for low metallicity AGB and post-AGB stars, so it may not be suitable to simulate more massive stars mass loss in the early RSG phase.

1.4.5 Vink *et al.*

Vink *et al.* [25, 64] provide a metallicity dependent algorithm from a theoretical model of the wind. They carry out a Monte Carlo simulation of the photon transport through the stellar atmosphere, allowing for multiple scattering and overlapping of the wings of adjacent spectral lines. They evaluate the momentum acquired by the gas as the momentum lost by photons.

The structure of the atmosphere, needed as an input to evaluate the radiative acceleration, is based on a gray LTE approximation (i.e. radiative equilibrium is not strictly fulfilled, otherwise it would not be an atmospheric model), and non-LTE populations of the levels of several species (H, He, C, N, O, Si) are computed. Iron group elements are treated in the so-called “generalized nebular approximation” [65], i.e. assuming that all ionization/recombination processes happen from/to the ground state only.

The authors explore the dependence of \dot{M} on various stellar parameters (M , the ratio of the escape velocity v_{esc} to the terminal velocity v_{∞} of the wind, Z , T_{eff} , L) by computing a grid of models, evaluating the wind and finding a fitting formula for their synthetic data.

The metallicity dependence of the wind mass loss rate is extracted by running simulations with varying Z in the range between $0.01Z_{\odot}$ and $10Z_{\odot}$, [25]. The scaling of \dot{M} with Z is smooth, such as in Eq. 1.35, in the range $1/30 \leq Z/Z_{\odot} \leq 3$, with an exponent $x \simeq 0.85$. By

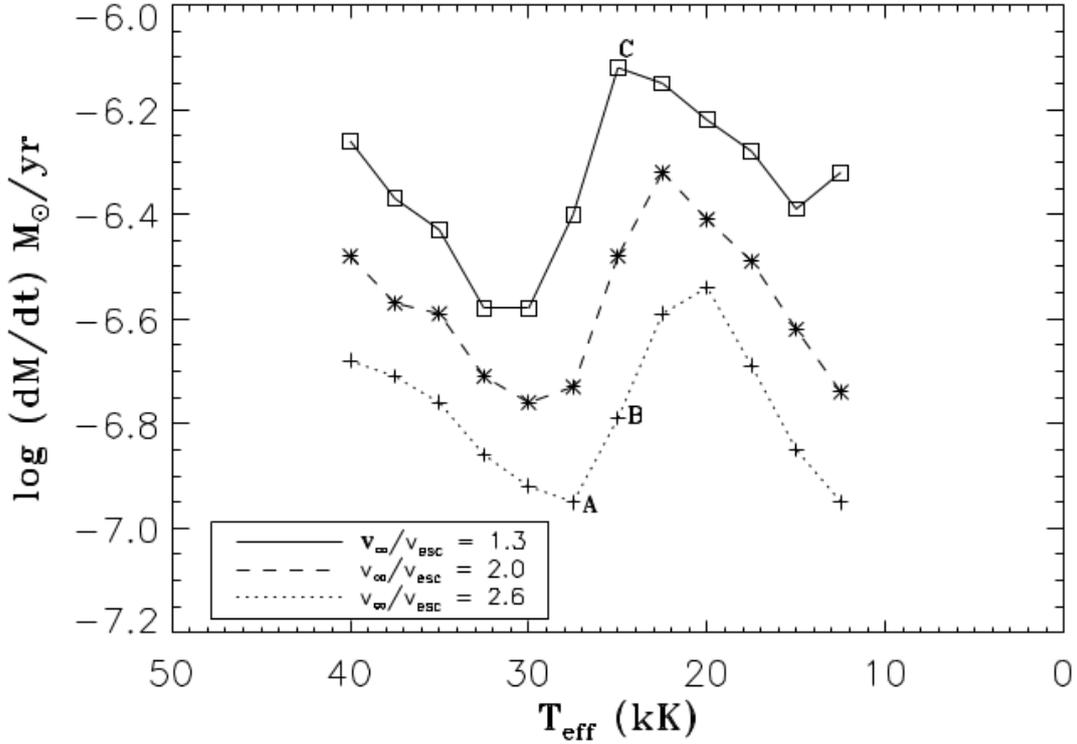


Figure 1.12: Predicted bistability jump in \dot{M} at $T_{\text{eff}} \sim 25000$ K. The stellar parameters used are $L = 5.0L_{\odot}$, $M = 20M_{\odot}$, $Z = Z_{\odot}$. This figure is Fig. (3) of [66].

carrying out simulations with different composition of the wind (only iron group elements, only lighter metals and then a third considering all of them), the authors find two different kinds of Z dependence of the mass loss rate: the interplay of these gives the smooth scaling. The first is the dependence of the mass loss rate on Z through the total amount of mass lost ($\dot{M} \equiv \dot{M}(M, \dots)$ and for each timestep t_n we have $M(t_n) = M(t_{n-1}) - |\dot{M}(t_n)| \cdot (t_n - t_{n-1})$, and the last term is a function of Z). The second is the dependence on Z through the wind final velocity ($\dot{M} \equiv \dot{M}(v_{\infty}, \dots)$ and $v_{\infty} \equiv v_{\infty}(Z)$). The total amount of mass lost, $M_{\text{lost}} = \dot{M}(t_n) \cdot (t_n - t_{n-1})$ is determined mainly by the abundance of iron group elements in the inner (subsonic) portion of the wind, because the great number of spectral lines of iron group elements causes the lion's share of line-driven acceleration in the subsonic part of the wind. The final velocity v_{∞} depends mainly on the abundances of lighter metals (C, N, O, etc...) in the outer (supersonic) part of the wind. These are irrelevant in the inner (supersonic) portion because they have many fewer spectral lines than iron group elements, but in the Vink *et al.* simulations they are found to dominate the outer subsonic portion of the wind [66]. Note that once the wind crosses the sonic point the amount of mass lost $M_{\text{lost}}(t_n)$ is set and only the final velocity is yet to be determined [66].

In the temperature range they explore, bi-stability jumps of \dot{M} are expected, see Fig. 1.12. These bi-stability jumps are regions in the T_{eff} domain where the behavior of \dot{M} is reversed, or in other words, regions where $d\dot{M}/dT_{\text{eff}}$ changes sign. For example, if $27500 \text{ K} \lesssim T_{\text{eff}} \lesssim 50000 \text{ K}$ the mass loss rate is lower for lower T_{eff} (i.e. $d\dot{M}/dT_{\text{eff}} > 0$), because the peak of the spectral distribution shifts to longer wavelengths where there are fewer spectral lines to drive the wind. The same happens for $12000 \text{ K} \lesssim T_{\text{eff}} \lesssim 22500 \text{ K}$, while around $T_{\text{eff}} \sim$

25000 K the opposite behavior is found: \dot{M} increases as T_{eff} decreases in this domain (i.e. $d\dot{M}/dT_{\text{eff}} < 0$). A similar behavior is found also in other temperature ranges.

These jumps happen because of recombination of certain ions when T_{eff} reaches the relevant values. For example, at $T_{\text{eff}} \sim 25000$ K recombination of Fe^{3+} to Fe^{2+} happens, and the corresponding bi-stability jump is due to the different cross sections for photon interactions of these different ions. To be more specific, Fe^{2+} has a higher cross section, therefore it is a better driver than Fe^{3+} .

The algorithm obtained by the authors gives two different relationships above and below the bi-stability jump at $T_{\text{eff}} \sim 25000$ K:

$$\begin{aligned} \log_{10}(-\dot{M}) = & -6.697(61) + 2.194(21) \log_{10} \left(\frac{L}{10^5 L_{\odot}} \right) - 1.313(46) \log_{10} \left(\frac{M}{30 M_{\odot}} \right) + \\ & -1.226(37) \log_{10} \left(\frac{v_{\infty}}{2v_{\text{esc}}} \right) + 0.933(64) \log_{10} \left(\frac{T_{\text{eff}}}{40000\text{K}} \right) + \\ & -10.92(90) \left[\log_{10} \left(\frac{T_{\text{eff}}}{40000\text{K}} \right) \right]^2 + 0.85(10) \log_{10} \left(\frac{Z}{Z_{\odot}} \right) , \end{aligned} \quad (1.47)$$

if $27500\text{K} < T_{\text{eff}} \leq 50000\text{K}$. For $12500\text{K} < T_{\text{eff}} \leq 22500\text{K}$ they give:

$$\begin{aligned} \log_{10}(-\dot{M}) = & -6.668(80) + 2.210(31) \log_{10} \left(\frac{L}{10^5 L_{\odot}} \right) - 1.339(68) \log_{10} \left(\frac{M}{30 M_{\odot}} \right) + \\ & -1.601(55) \log_{10} \left(\frac{v_{\infty}}{2v_{\text{esc}}} \right) + 1.07(10) \log_{10} \left(\frac{T_{\text{eff}}}{40000\text{K}} \right) + \\ & + 0.85(10) \log_{10} \left(\frac{Z}{Z_{\odot}} \right) . \end{aligned} \quad (1.48)$$

In Eq. 1.48 and Eq. 1.47, the numbers in parenthesis indicate the estimated error on the last digits of the coefficients, as it is reported in [25]. This error is usually neglected in stellar evolution calculations.

Note that this algorithm is valid only for stars in the temperature ranges indicated (even if several extrapolations have been suggested in the literature, e.g. [67]). It is not valid for stars on the cool (red) side of the HR diagram.

1.4.6 Kudritzki *et al.*

The algorithm proposed by Kudritzki *et al.* [68] differs conceptually from the those described so far. It is an analytic solution for the system of equations describing a stationary, isothermal, spherically-symmetric, ideal (neither viscosity nor heat conduction) gas flow with no magnetic fields and no rotation. This system consists of the Euler equation, the continuity equation and an (isothermal) equation of state (EOS). The Euler equation for a stationary ($\partial_t = 0$) spherical-symmetric ($\partial_{\theta} \equiv \partial_{\varphi} = 0$) ideal gas reads:

$$v \frac{dv}{dr} = -\frac{1}{\rho} \frac{dP}{dr} - \frac{GM}{r^2} + g_{\text{ph}} , \quad (1.49)$$

where g_{ph} is the radiative acceleration, a source term representing the effects of photon-gas interaction, i.e. Thomson scattering on free electrons and line scattering and absorption

effects. In the spherically symmetric case, the radiative acceleration can be parametrized following Castor *et al.* (CAK) [69], as

$$g_{\text{ph}} = g_{\text{rad}}^{\text{Th}} \left(1 + k \left(\frac{\kappa_{\text{Th}} \rho c_s}{dv/dr} \right)^{-\alpha} \left(\frac{2n_e}{\sqrt{1 - (1 - (R/r)^2)}} \right)^{\delta} \text{CF} \left(r, v, \frac{dv}{dr} \right) \right), \quad (1.50)$$

where $g_{\text{rad}}^{\text{Th}}$ is the radiative acceleration due to Thomson scattering, and the second term in the brackets is the so-called “force multiplier”, i.e. the line acceleration in units of $g_{\text{rad}}^{\text{Th}}$. It depends on the Thomson opacity κ_{Th} , the speed of sound c_s , the radius of the star R , the electron number density n_e , and three free parameters k , α , and δ , used to parametrize the line opacity. k can be interpreted roughly as the number of lines strong enough to have an effect, and α as the slope of the distribution of the number of lines as a function of their strength.

The parameter δ and the correction factor CF are used to include the “finite cone-angle effect”, i.e. the fact that incoming photons may not be in the radial directions [68]. CF is the ratio between the opacity as a function of the incoming angle and the opacity in the radial direction, and it can be expressed as a function of the radial coordinate $x \stackrel{\text{def}}{=} r/R$ and $h \stackrel{\text{def}}{=} d \log_{10}(x) / d \log_{10}(v)$, cf. Eq. (4) in [68], as

$$\text{CF} \left(r, v, \frac{dv}{dr}, \alpha \right) = \left(\frac{1}{\alpha + 1} \right) \left(\frac{x^2}{1 - h} \right) \left(1 - \left(1 - \frac{1}{x^2} + \frac{h}{x^2} \right)^{\alpha+1} \right). \quad (1.51)$$

If both δ and CF were equal to one, the parametrization would be valid only in the “radial streaming” limit [70, 71]. While this is a good approximation in the outer portion of the wind, where $r \gg R$ (R is the stellar radius, i.e. the inner boundary at the photosphere), it is quite poor in the inner portion, where the mass loss rate is set and photons traveling in non-radial direction have a relevant effect.

Note that α , k and δ parametrize not only the interaction between photospheric photons and metal lines, but also the feedback on the radiation field, whose energy density is increased by the contribution of scattered and re-emitted photons.

The reader is referred to [55, 69, 70] and references therein for more details on this parametrization.

The EOS is isothermal:

$$P = \rho c_s^2, \quad T = \text{const.} \Rightarrow c_s = \text{const.}, \quad (1.52)$$

Note that the thermal structure of the wind $T \equiv T(r)$ has a small influence on the mass loss rate, although it influences the ionization fraction. Pauldrach *et al.* [70] state that the most noticeable effect is a change in the position of the critical point (see below). Thus, Kudritzki *et al.* consider the isothermal EOS of Eq. 1.52 for the sake of simplicity.

Finally, the continuity equation completes the system:

$$\dot{M} = 4\pi r^2 \rho v. \quad (1.53)$$

Substituting Eq. 1.53 and Eq. 1.52 into Eq. 1.49, we obtain

$$v \frac{dv}{dr} = -c_s^2 \frac{4\pi r^2 v}{\dot{M}} \frac{d}{dr} \left(\frac{\dot{M}}{4\pi r^2 v} \right) - \frac{GM}{r^2} + g_{\text{ph}}, \quad (1.54)$$

which does not allow an analytic solution without further assumptions. The authors of [68], following CAK, rewrite Eq. 1.54 as

$$F(x, y, v) = 0 , \quad (1.55)$$

where $x = r/R$ is the dimensionless radius, and $y = r^2 v (dv/dr)$. In this form it is easier to see that there is a critical point where

$$\frac{\delta F}{\delta y} = 0 , \quad (1.56)$$

and the regularity of the solution requires:

$$\frac{\delta F}{\delta x} + \frac{\delta F}{\delta v} \frac{\delta v}{\delta x} = 0 . \quad (1.57)$$

To find a solution, the authors impose a “ β -law” velocity field on the outflowing matter composing the wind,

$$v(r) \simeq v_\infty \left(1 - \frac{R}{r}\right)^\beta , \quad (1.58)$$

beyond the critical point. β is a free parameter, whose value has little influence on the result, and v_∞ is the asymptotic velocity of the wind. The velocity field given by Eq. 1.58 is found to be close to the numerical solution, and it is a common assumption of many stellar wind models [5]. Note that this assumption means that the authors do not solve for the dynamics of the wind, but instead they just solve self-consistently for the density and force imposing a velocity field.

With this assumption, using $\beta = 1$, and assuming also $v \gg c_s$, which is likely in the outer part of the wind, they use Eq. 1.56 and Eq. 1.57 to find

$$\tilde{D}(\alpha, \delta, v(r = r_{\text{crit}}), v_\infty) \left(\frac{\sigma_{\text{Th}} k L}{4\pi c}\right)^{1/(\alpha-\delta)} \left(\frac{4\pi\alpha}{\sigma_{\text{Th}} v_{\text{th}}}\right)^{\alpha/(\alpha-\delta)} \left(\frac{1-\alpha}{GM(1-\Gamma_E)}\right)^{(1-\alpha)/(\alpha-\delta)} , \quad (1.59)$$

where σ_{Th} is the cross section for Thomson scattering, v_{th} is the thermal velocity of protons, k , α , and δ are free parameters, L is the luminosity, $\Gamma_E = L/L_{\text{Edd}}$ is the Eddington ratio and \tilde{D} is a known function of the free parameters and the velocity of the wind at the critical point and at infinity, [68].

The suggested values for the free parameters are $\alpha = 0.657$, $\beta = 1$, $\delta = 0.095$, and $k = 0.085$, following the algorithm of Wellstein & Langer [72]. These values were derived to reproduce the observed mass loss properties of ζ Puppis by Pauldrach *et al.* [73]. The main limitation of this algorithm is that the free parameters α , β and k are not constant, but rather depend on the optical depth. The values suggested should be interpreted as depth-independent mean values.

1.4.7 Nugis & Lamers

The mass loss algorithm derived by Nugis & Lamers [74] applies only to WR stars, i.e. hydrogen-poor stars with a very strong wind manifested by strong, broad emission and P

Cygni lines. The wind mass loss rate of these stars depends heavily on their chemical composition: the metallicity and also the helium mass fraction Y . This is because the amount of helium in the stellar atmosphere changes the mean molecular weight, influencing the temperature and therefore the ionization fraction and the level populations of all other atoms and ions.

Nugis & Lamers [74] derive a mass loss rate algorithm as a function of the luminosity and the chemical composition starting from a sample of observed galactic WR stars. Their sample consists of two subsets: those for which both mass and distance (i.e. luminosity) are known, thanks to binarity or membership association in open clusters; and those for which the luminosity is not known. They use stars from the first subset to derive an empirical bolometric correction (i.e. the difference between the bolometric luminosity and the observed luminosity, $BC \stackrel{\text{def}}{=} L - L_{\text{obs}}$). They then use a theoretical mass-luminosity relation to infer the luminosity of stars in the second subset, and correct it with the previously derived bolometric correction. Note that the mass-luminosity relation is derived using only the age of the star and its spectral type, therefore it can be used self consistently for the luminosity determination, which does not enter as an input [74].

The mass loss rate for the stars in the sample, determined from observations of the radio continuum, are then fitted as follows. The authors make two independent fits for stars of different composition and then merge them together in a single formula, valid for all WR stars:

$$\log_{10}(-\dot{M}) = -11.0 + 1.29(14) \log_{10}(L/L_{\odot}) + 1.73(42) \log_{10}(Y) + 0.47(09) \log_{10}(Z) . \quad (1.60)$$

In Eq. 1.60, the number in parenthesis indicate the estimated uncertainty in the last digits of the coefficients.

It is worth mentioning that the authors of [74] state that the mass loss algorithm for WR stars cannot be expressed as a function of T_{eff} and/or the radius of the star R . This because WR winds are so strong (i.e. dense) that they are optically thick and their velocity gradient is very steep. Therefore, the outward energy flux (or, equivalently, T_{eff}) is strongly dependent on the wavelength observed, or, in other words on the velocity and density structure of the wind. This prevents using T_{eff} to parametrize the wind mass loss rate self-consistently.

1.4.8 Hamann *et al.*

This wind scheme, which applies only to WR stars, is a combination of the algorithms from [75–77].

It is derived from a spherically symmetric, homogeneous and stationary (but not static, i.e. $\partial_t = 0$ but $v \neq 0$) expanding WR atmosphere model. The authors assume an *ad-hoc* velocity structure $v \equiv v(r)$: for the supersonic part of the wind they assume a β -law in the form of Eq. 1.58 with $\beta = 1$, while, for the subsonic part, $v(r)$ is chosen such that the density approaches the thermal velocity. Note that since the velocity field is imposed, the acceleration is not computed.

This allows the authors to adopt a very simple chemical composition, since they do not need to evaluate the line-driven acceleration and do not need to keep track of all atomic species and their level populations. The authors include only ions of H and He, and the radiation field is considered only to determine non-LTE populations of these species, which are necessary to produce synthetic spectra to compare with observations. The temperature

stratification is derived with the assumption of a grey LTE model, assuming a value of T_{eff} at the base of the atmosphere determined by the stellar luminosity L and the radius R via a black body relation (Eq. 1.38), in contrast with the suggestion of Nugis & Lamers [74].

A synthetic spectrum is derived from the simulations and a best fit to the observed line profiles for ions of H and He is obtained by variation of the stellar parameters (i.e. the radius of the inner boundary of the atmosphere and L , the surface hydrogen mass fraction X_s and \dot{M}) [76]. Once the stellar parameters are known from this fit, a mass loss formula is derived for high luminosity WR stars, i.e. $\log_{10}(L/L_{\odot}) > 4.5$. The algorithm for the low luminosity WR stars, i.e. $\log_{10}(L/L_{\odot}) \leq 4.5$, is derived with a similar technique [75], but the spectra fitted are from a small sample of Helium stars (i.e. stars during He shell burning with most of the mass in a C/O core, without H lines in their spectra).

The resulting mass loss algorithm is

$$\log_{10}(-\dot{M}) = \begin{cases} -12.25 + 1.5 \log_{10} \left(\frac{L}{L_{\odot}} \right) - 2.85 X_s & \text{if } \log_{10}(L/L_{\odot}) > 4.5 \text{ ,} \\ -35.8 + 6.8 \log_{10} \left(\frac{L}{L_{\odot}} \right) & \text{if } \log_{10}(L/L_{\odot}) \leq 4.5 \text{ .} \end{cases} \quad (1.61)$$

Variant models that include inhomogeneities in the wind have been proposed in [77]. Specifically, the authors suggest reducing the wind efficiency by a factor between 2 and 3 to account for the wind clumpiness. Note that this algorithm is derived considering only spectral lines for a few ions.

1.4.9 Motivation to introduce non-wind Mass Loss Processes in Simulations of Massive Star Evolution

Since recent studies suggest that the wind mass loss rates commonly used in stellar evolution calculations may be overestimated by a factor between 2 to 10 [e.g., 7, 49], e.g. because of wind clumpiness, the lower, revised rates imply that another mass loss channel(s) must exist to produce pre-SN stellar structures with little or no hydrogen-rich envelope (e.g. the progenitors of type IIb/Ib/Ic SNe). Such additional mass loss mechanism(s) may also be relevant for producing YSG and BSG pre-SN structures, and thus be important for the solution of the RSG problem (see §1.2.3 and [29]).

Several additional mechanisms have been proposed to remove part or all of a massive stars's hydrogen-rich envelope. Theoretical studies suggest that single stars may shed their envelopes through pulsational instabilities and/or wave driven mass loss [e.g., 50], and there are many observational signatures of episodic events, like luminous blue variable (LBV) eruptions²², or circumstellar shells and rings [e.g., 79, 80].

Other mass loss mechanisms, e.g. Roche Lobe Overflow (RLOF), depend on the presence of a companion star. By definition, the Roche Lobe is the volume around one of the stars of the binary within which matter is gravitationally bound to that star. Therefore, whenever a parcel of gas exits the Roche Lobe (e.g. because of the inflation of the star during its evolution), it is no more bound to the star. Mass loss mechanisms depending on the presence of a companion star may be especially relevant since virtually all massive stars are in multiple systems and up to $\sim 70\%$ of them interact with their companion before core collapse [20]. RLOF is likely to happen when a massive star expands after leaving the main sequence. The precise moment at which RLOF begins and whether it leads to a common envelope phase

²²However, it is possible that binarity plays a role in formation and/or evolution of LBVs, e.g. [78].

and/or merger depend on the properties of the binary (mass ratio, period, eccentricity, angular momenta, etc.). Further, the structure of the star at the onset of RLOF, along with the binary parameters, determine whether the star undergoes dynamically stable or unstable mass loss.

However, it is possible that (some of) these mechanisms cannot completely remove the hydrogen envelope, and that regular line or dust driven stellar winds take the burden of removing the remaining hydrogen-rich layer in the subsequent evolution. Another possibility to consider are “non-catastrophic” mass loss events, such as small eruptions, which are also neglected in stellar evolution calculations, as their timescale, and consequently the possible stellar response are not known.

All of the mechanisms mentioned above are intrinsically dynamical, and/or multidimensional, and contrary to winds, they cannot be simulated using common, one-dimensional, steady-state techniques. This is because they may happen on timescales that are shorter than the thermal timescale of the star, preventing a hydrostatic re-adjustment of the structure to the mass loss. Moreover, contrary to winds, they can have energy loss rates that are large fractions of the stellar luminosity. There is also considerable uncertainty in the processes that trigger and terminate these events. Therefore, the self-consistent simulation of such dynamical mass loss mechanisms is not possible with current stellar evolution codes. Thus, these events are usually neglected in stellar evolution calculations, although, if the wind mass loss rates are indeed overestimated, they are far from rare, since $\sim 30\%$ of the observed SNe lacks hydrogen lines and comes from progenitors without an extended hydrogen envelope, [29].

Pre-SN structures with little or no hydrogen envelope have been produced by increasing the wind mass loss rate [e.g. 46]. However, this is unrealistic: stellar winds are radiatively driven phenomena, and the corresponding mass loss rates depend on the physical conditions in the outer portion of the star, (e.g. luminosity, effective temperature, and metallicity), while the process that actually removes the hydrogen envelope may not be a radiative phenomenon with the same kind of dependencies. In fact, for most of the proposed mass loss mechanisms, the amount of mass lost could be determined by the inner structure (location of the burning shells, density profile, etc.) at a given moment of the evolution. Additionally, the amount of mass removed at each timestep, and the corresponding readjustment of the structure, will be very different in the case of a steady wind-like mass loss and in the case of a rapid, impulsive, mass loss event, that occurs all at once with a fixed initial structure.

The exploration of the effects of catastrophic mass loss events on the stellar structure is still in a very infant stage, because of the many technical difficulties that come with it. More work to understand these phenomena is needed, especially for their relevance in explaining the origin of the progenitors of (almost) hydrogen-free core-collapse SNe. Moreover, it is possible that such events may also play a role in the solution of the RSG problem.

CHAPTER 2

Methods

dove Alinardo sembra dare informazioni preziose e Guglielmo rivela il suo metodo per arrivare a una verità probabile attraverso una serie di sicuri errori
[U. Eco, Il Nome della Rosa, Quarto Giorno – Vespri]

2.1 The MESA Code

MESA (Modules for Experiments in Stellar Astrophysics) [39, 81] is an open-source one-dimensional (1D) stellar evolution code, organized as a set of independent threadsafe¹ modules. Each module deals with one aspect of the physics (e.g. EOS, opacities, the nuclear reaction network, etc.) or the numerics (i.e. matrix operations², interpolation of data sets, numerical error handling, etc.). The module that performs one step of the stellar evolution, by calling all other modules in the appropriate order, is the module `star` (see `$MESA_DIR/star` in a standard installation and [81]). The code is written in FORTRAN 90, and is designed to be easy to modify, thanks to the possibility of adding custom algorithms for many purposes (wind mass loss, meshing, stopping conditions, output settings, etc.) locally, without the need to modify and recompile the whole code (see §2.2). Simulations presented in this work are carried out with the release 6794 of MESA (July, 20, 2014).

Extensive descriptions of the code capabilities are available in [39, 81], even if its rapid development makes them outdated in some parts. It is not my goal to complete or update these descriptions, and in the following sections I give a brief general description of the numerical implementation of the code (§2.1.1), and then focus on the features I used (and modified) to perform the systematic study of massive star mass loss presented in this work (§2.1.3, §2.2.2, §2.2.1). In §2.3 and §2.4 I describe the setup used for the systematic study of wind mass loss, and for the effects of violent, eruptive mass loss events, respectively.

2.1.1 Overview of the Numerical Implementation

The equations describing stellar structures are a set of coupled non-linear differential equations, which cannot be solved analytically (see also below). Therefore, stellar astrophysics relies on numerical simulations, which requires a number of (numerical and physical) approximations (e.g. finite differences instead of derivatives) to get a solution.

¹Multicore processor capabilities can be exploited using OpenMP parallelization.

²Sparse matrix operations are handled with the “SPARSEKIT” solver, see [81].

MESA star is a Henyey-like [21] code, which uses the MESA modules to solve the equations of a stellar structure, with automatic mesh refinement, adaptive time-step control, and analytic jacobians. In a Henyey-like code, the solution for the stellar structure at a given time is found starting from a trial solution, and correcting it with (small) variations of all the parameters at all points simultaneously until the required degree of accuracy is reached. The iterations on the primitives is performed using a generalized Newton-Raphson scheme, see below. This method is particularly convenient for problems, such as stellar evolution, for which boundary conditions are given at both ends of the computational domain (i.e. at the center – e.g. $L_c = 0$ and $M_c = 0$ – and at the surface – e.g. T_{eff} and P_{out}). This is because the Henyey method (in contrast with the shooting method) varies the boundary condition, and finds more easily a solution within the degree of accuracy required [21].

Meshing

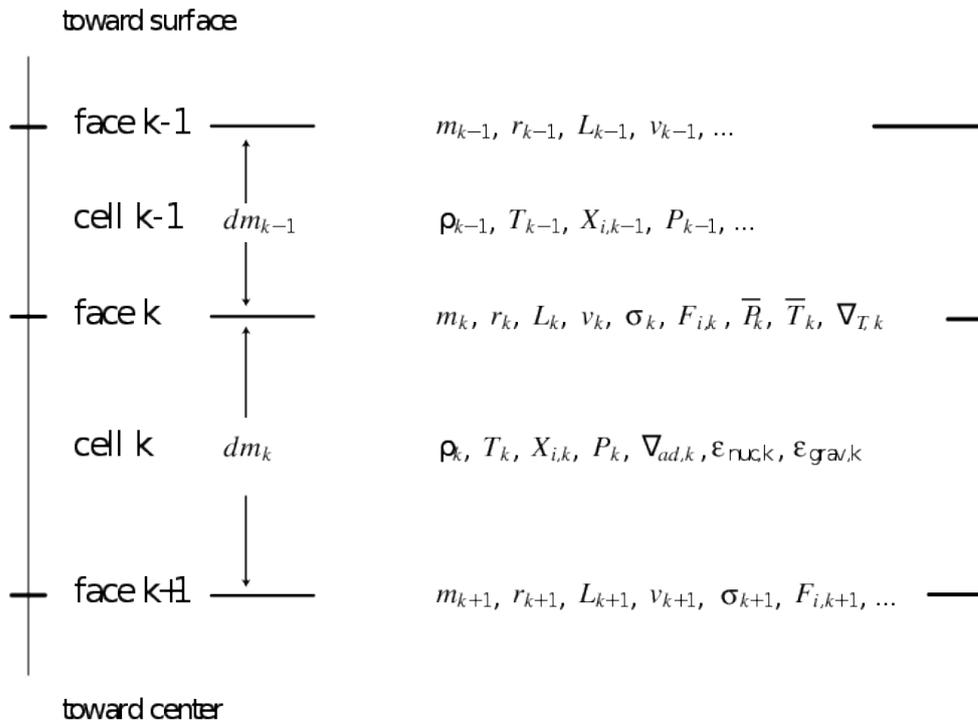


Figure 2.1: Schematic representation of the structure of the mesh in MESA. Note that cells are numbered from 1 on the surface, to the maximum value in the center. This figure is a modification of Fig. 9 in [81].

At the beginning of each step, MESA defines a grid of points (cell boundaries), and evaluates finite differences between the adjacent points instead of derivatives. The meshing algorithm is described extensively in §6.5 of [81]. The number of points is determined automatically by full-filling some user defined conditions. In a typical situation, the user controls the number of points – and thus the spatial resolution – using two parameters, `mesh_delta_coeff` and `mesh_delta_coeff_for_highT`. These multiply the maximum relative difference allowed between two adjacent cells for several quantities ($\log_{10}(P)$, $\log_{10}(T)$, $\log_{10}(R)$, $\log_{10}(Y)$, where Y is the local abundance of helium).

Note that a lower `mesh_delta_coeff` (`mesh_delta_coeff_for_highT`) means that smaller

variations are enforced, thus the code divides the star in a larger number of cells and higher spatial resolution is obtained (at the cost of computational time and memory). Moreover, MESA includes parameters to increase the resolution locally in critical regions of the star (e.g. where there is a large variation of the energy generation rate or of the abundance of certain isotopes).

After each evolutionary step (see also below), MESA tries to use the mesh from the previous solution for the new (evolved) structure. If one (or more) of the criteria determining the mesh is not met, MESA re-meshes the structure by splitting some cells into two parts, and/or merging adjacent cells. The interested reader should consult the MESA documentation (see `$MESA_DIR/star/defaults/controls.defaults` in a standard installation).

MESA is a Lagrangian code, i.e. it uses as independent coordinate a Lagrangian coordinate, namely the mass of each $k - th$ cell m_k . Note however that it is hydrostatic, i.e. time independent code. Fig. 2.1.1 shows a schematic representation of the mesh in MESA. In general, extensive quantities (r, L, m , etc.) are given at the boundary of each computational cell, while intensive quantities (T, ρ, P , etc.) are averaged in mass over the cell.

Time-step control

To follow the temporal evolution of the stellar structure, the time dimension must be discretized – together with the spatial dimension(s). For this purpose, a timestep is defined for each iteration. The timestep Δt is the (finite) amount of time used to compute time-derivatives instead of the (infinitesimal) dt . For example, in a forward finite differences scheme the time derivative for the $k - th$ step is

$$\frac{df}{dt} = \frac{f(t_{k+1}) - f(t_k)}{t_{k+1} - t_k} \rightarrow \Delta t_k = t_{k+1} - t_k, \quad (2.1)$$

where Δt_k is the timestep for the $k - th$ step. However, note that MESA is an implicit code. The timestep should be sufficiently small to resolve all time variations, but sufficiently large to allow a reasonable run-time³. This makes the selection of the timestep a complex process, since several timescales are involved in the evolution of a star (nuclear, free fall, Kelvin-Helmholtz, mass change, sound crossing time, etc.), and they are usually very different from each other.

MESA does the timestep selection in a two-stage process, described in §6.4 of [81]. First, a scheme based on “Digital Control Theory” is used to estimate a timestep. This means that Fourier transforms are used to evaluate the characteristic frequencies of the system, and a timestep smaller than all the corresponding periods is chosen. At this stage MESA only checks if the control variable v_c , i.e. the unweighted average of the relative variation of $\log_{10}(T)$, $\log_{10}(\rho)$, $\log_{10}(Y)$ and $\log_{10}(r)$ across all cells, is smaller than a user-defined value `varcontrol_target`, whose default is $v_t = 10^{-4}$. The timestep proposed during this stage is defined to be a function of the previous two timesteps, to improve numerical stability. If Δt_{k-1} and Δt_k are the timesteps for the $k - th$ and $(k - 1) - th$ steps, and $v_{c,k-1}$ and $v_{c,k}$ are the corresponding control variables, then the timestep proposed for the $(k + 1) - th$ step is

$$\Delta t_{k+1} = \Delta t_k g \left(\frac{g(v_t/v_{c,k})g(v_t/v_{c,k-1})}{g(\Delta t_k/\Delta t_{k-1})} \right)^{1/4}, \quad (2.2)$$

³Moreover, if the timesteps taken are too small, actual physical processes in nature may not have enough time to make a physical region as large as a cell of the mesh reach equilibrium.

where $g(x) \stackrel{\text{def}}{=} 1 + 2 \tan^{-1}(0.5(x - 1))$. Then, in the second stage, MESA does additional checks that can further reduce the timestep yielded by Eq. 2.2. These check that the variations of several quantities (maximum nuclear burning rate, maximum change in surface luminosity, etc.) are lower than the corresponding thresholds. It is possible to customize the thresholds and implement other checks to enforce smaller timesteps (see §2.2 and §2.2.1, and `$MESA_DIR/star/defaults/controls.defaults` for the default checks).

Reformulation of the Stellar Structure Equations for Numerical Stability

The stellar structure equations⁴ are reformulated (see §6.2 of [81]) to improve the numerical stability of the MESA code. To limit round-off errors, the continuity equation is written as:

$$\ln(r_k) = \frac{1}{3} \ln \left[r_{k+1}^3 + \frac{3}{4\pi} \frac{dm_k}{\rho_k} \right], \quad (2.3)$$

where ρ_k is the average density in the cell, and $dm_k \stackrel{\text{def}}{=} m_k - m_{k-1}$ is the mass of the computational cell (cf. Fig 2.1.1).

The hydrostatic balance equation is written as

$$\frac{P_{k-1} - P_k}{0.5(dm_{k-1} - dm_k)} = \left(\frac{dP}{dm} \Big|_k \right)_{\text{static}} + \left(\frac{dP}{dm} \Big|_k \right)_{\text{dynamic}} = -\frac{Gm_k}{4\pi r_k^4} - \frac{a_k}{4\pi r_k^2}, \quad (2.4)$$

where P_k is the (mass average) total pressure in the k -th cell, m_k is the total mass enclosed from the center to the outer boundary of the cell, $dm_k = m_k - m_{k+1}$ is the mass in the k -th cell, r_k its radius and

$$a_k \stackrel{\text{def}}{=} \frac{dv_k}{dt}, \quad v_k = r_k \frac{d \ln(r_k)}{dt}, \quad (2.5)$$

is the Lagrangian acceleration, with v_k the cell velocity computed as the variation of the radius between two successive timesteps. Eq. 2.4 includes both the hydrostatic and the so-called “dynamical pressure”⁵ term, which are the first and second term on the right hand side, respectively. The dynamical term applies only if the user specifies that velocities are to be computed explicitly, in which case v_k is divided by the local sound speed to improve numerical stability. Otherwise a_k is set to zero.

The energy transportation equation is

$$\frac{T_{k-1} - T_k}{0.5(dm_{k-1} - dm_k)} = -\nabla_{\text{loc},k} \left(\frac{dP}{dm} \Big|_k \right)_{\text{static}} \frac{\langle T_k \rangle}{\langle P_k \rangle}, \quad (2.6)$$

where T_k is the cell temperature, $\nabla_{\text{loc}} \stackrel{\text{def}}{=} d \log(T) / d \log(P)$ and

$$\langle f_k \rangle \stackrel{\text{def}}{=} \frac{f_{k-1} dm_k + f_k dm_{k-1}}{dm_{k-1} + dm_k} \equiv \frac{\frac{f_{k-1}}{dm_{k-1}} + \frac{f_k}{dm_k}}{\frac{1}{dm_{k-1}} + \frac{1}{dm_k}}, \quad (2.7)$$

is a weighted average value of the function $f(=T$ or $P)$ between the two adjacent cells. For numerical stability Eq. 2.4 is divided by $\langle P_k \rangle$ and Eq. 2.6 is divided by $\langle T_k \rangle$.

⁴See e.g. [23, Ch. 21] for the original formulation of these equations.

⁵The “dynamical pressure” is not really a pressure, but a correction to account for the movement of mass. It is usually a very small term.

Energy conservation equation reads:

$$L_k - L_{k+1} = dm_k \{ \varepsilon_{\text{nuc}} - \varepsilon_\nu + \varepsilon_{\text{grav}} \} , \quad (2.8)$$

where ε_i are the local energy generation rate per unit mass:

- ε_{nuc} is the total energy generation rate from all nuclear reactions, minus the energy going into neutrinos (which leave immediately the star). It comes from the JINA⁶ database, as tables in function of T and ρ . MESA does the interpolation between the available data automatically;
- ε_ν is the thermal neutrino loss rate, that is significant for late burning stages in massive stars. This term includes only thermal neutrinos because those from nucleosynthetic processes are already subtracted from the nuclear energy generation rate ε_{nuc} ;
- $\varepsilon_{\text{grav}} = -Tds/dt$ is the so-called “gravitational work”, which expresses the specific rate of change of gravitational energy because of the gas expansion or contraction. The time derivative of the specific entropy s is evaluated as a function of the thermodynamical quantities and the EOS, see Eq. 12 in [81].

The EOS uses T and ρ as independent variables, and it is put in tabular form. It also depends implicitly on the chemical composition of the gas, e.g. free electrons and ionic species available. EOS tables are loaded from different sources, depending on the temperature and density range. Computation shown in this work use the default option, which corresponds, for the relevant density ($-10 \lesssim \log_{10}(\rho/[g \text{ cm}^{-3}]) \lesssim 10$) and temperature ($3 \lesssim \log_{10}(T/[K]) \lesssim 10$) ranges to the OPAL [83] and HELM [84] databases.

Finally, the equation for the chemical abundance $X_{i,k}$ of the i -th species in the k -th cell reads

$$X_{i,k}(t + \Delta t) = X_{i,k}(t) + \left(\frac{dX_{i,k}}{dt} \right)_{\text{nuc}} \Delta t + \frac{\Delta t}{dm_k} (F_{i,k+1} - F_{i,k}) , \quad (2.9)$$

where

$$F_{i,k} \stackrel{\text{def}}{=} \frac{\sigma_k}{0.5(dm_{k-1} - dm_k)} (X_{i,k} - X_{i,k-1}) . \quad (2.10)$$

In Eq. 2.9, the second term on the left hand side is evaluated from the net module (see `$MESA_DIR/net`) and the third term includes all mixing processes (chemical diffusion, radiative levitation, overshooting, shear, rotational mixing, etc.). σ_k is an appropriate coefficient⁷, and Δt is the timestep.

The set of isotopes that MESA traces is defined in the nuclear reactions network. This is specified by choosing between optimized built-in options, or as a list of isotopes and reactions. MESA keeps track only of the isotopes listed in the network. The initial abundances are specified through hydrogen abundance, helium abundance and metallicity (X , Y , and Z , respectively), and by default a scaling with the solar value is assumed for each isotope. This assumption is typical in stellar evolution codes, but it is questionable. Since there is evidence that isotopes are synthesized in a large variety of astrophysical processes (stellar cores and shells, novae, supernovae, binary neutron star mergers, etc. See, e.g., [3, 4, 85] and references therein) and nothing guarantees that the scaling with solar abundances is correct, MESA also includes options to specify the initial composition as a list of isotopes with the corresponding abundances.

⁶An option to use the NACRE [82] database is provided.

⁷Note that, in principle, the instabilities could cancel out, instead of summing, even if usually one instability is dominant and develops faster than all the others.

Algorithm to Solve the Equations

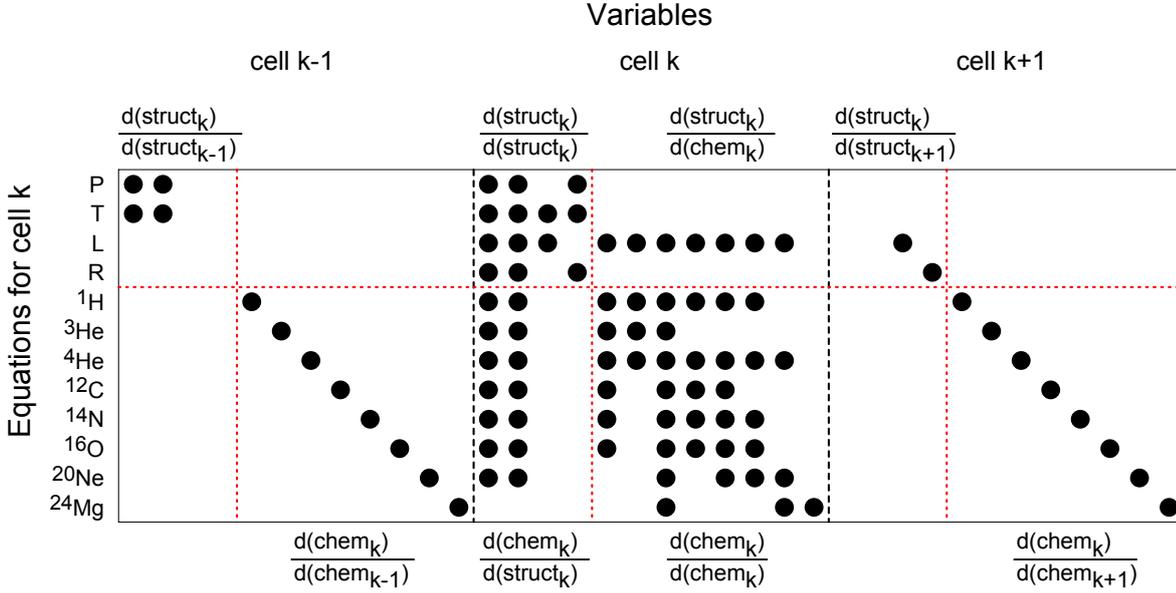


Figure 2.2: Schematic representation of one row of the matrix representing the algebraic system that MESA solves to find δy_i at each i -th iteration of the Newton-Raphson solver. Black dots indicate nonzero entries (note the diagonal terms in the lower right sub-block of the left and right blocks, representing the contribution of the chemical mixing between adjacent cells). This is Fig. 47 of [39].

The full set of coupled nonlinear differential (transformed into difference equation by the discretization) algebraic equations (Eq. 2.4, Eq. 2.3, Eq. 2.6, Eq. 2.8, Eq. 2.9 and the tabulated EOS) is simultaneously solved for the structure ($T, \rho, L, \text{etc.}$) and composition ($X_i \forall i$ in the nuclear reactions network). MESA uses a generalized Newton-Raphson solver⁸ to integrate the full set of equations (see §3 in [81] and the appendix in [39]). The equations are written in the form

$$0 = \mathbb{F}(y) = \mathbb{F}(y_i + \delta y_i) = \mathbb{F}(y_i) + \left[\frac{d\mathbb{F}(y)}{dy} \right]_i \delta y_i + \mathcal{O}((\delta y_i)^2), \quad (2.11)$$

where y_i is a trial solution (usually, the first iteration uses as its first guess the solution of the previous timestep, just re-meshed and with the mass change from winds or accretion applied), \mathbb{F} is the (vector valued) residual, i.e. the distance between the trial solution y_i and the (unknown) exact solution y , and the term in brackets is the Jacobian matrix, calculated analytically. For the calculation of the Jacobian matrix, each MESA module evaluates not only the output quantities, but also all the partial derivatives with respect any input variable.

Neglecting the term of $\mathcal{O}((\delta y_i)^2)$, Eq. 2.11 provides the correction δy_i needed to go from the trial solution y_i to an improved solution y_{i+1} ⁹. Iterating this process – usually more

⁸i.e. based on the algorithm developed to solve the equation $f(x) = 0$ with an iterative procedure:

$$x_{n+1} = \left(x_n - \frac{f(x_n)}{f'(x_n)} \right) \rightarrow x_\infty, \text{ with } f(x_\infty) = 0.$$

⁹The improved solution is not the exact solution, because of the approximation used to neglect terms of $\mathcal{O}((\delta y_i)^2)$.

than twice and less than ten times – a solution y_j with a “sufficiently small” residual $F(y_j)$ is found and accepted as a good approximation to the exact solution, $y_j \simeq y$. Note that the definition of what is “sufficiently small”, or, in other words, the maximum value of the term of order $\mathcal{O}((\delta y_i)^2)$ neglected is what limits the gridding. This procedure requires to solve the algebraic Eq. 2.11 for δy_i at each i -th iteration of the Newton-Raphson solver, which means solving a (very large) matrix. A row of such a matrix is depicted in Fig. 2.2.

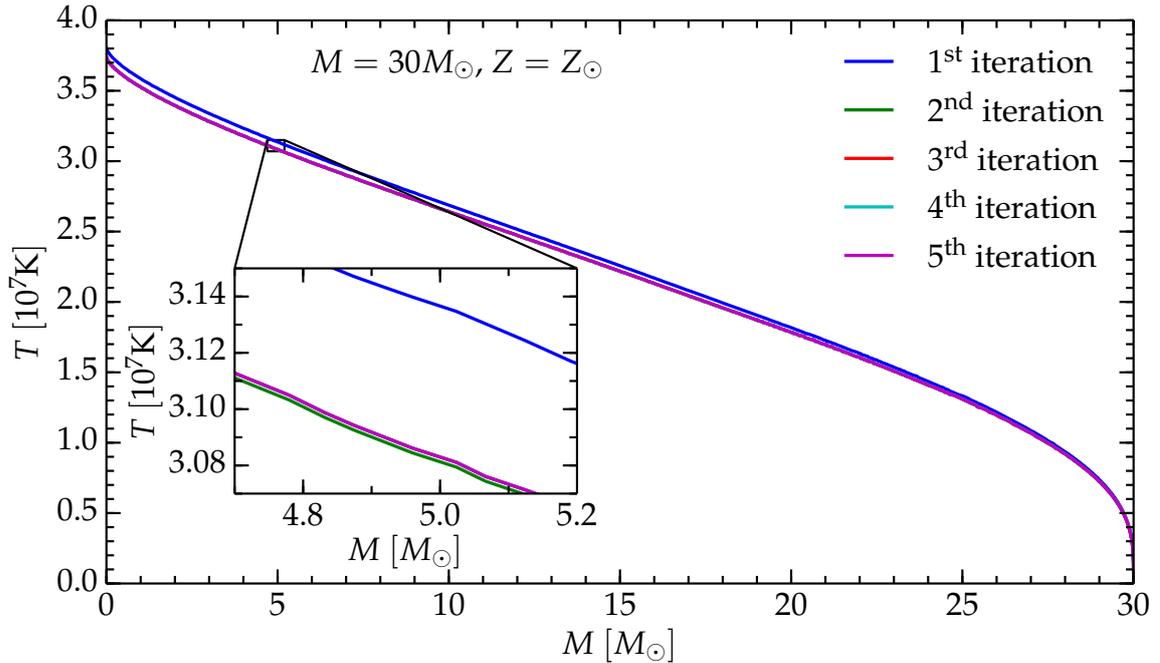


Figure 2.3: $T \equiv T(M)$ for $M_{\text{ZAMS}} = 30M_{\odot}$, $Z = Z_{\odot}$ model at ZAMS. Note the use of a linear scale on both axes, which (barely) allows to resolve the readjustments of the structure.

Fig. 2.3 shows the temperature profile at each iteration of the Newton-Raphson solver for a model at the beginning of the main sequence. Note the use of a linear scale on both axes: the relative adjustment of each quantity is small and not visible on a logarithmic scale.

When MESA starts from the ZAMS, it loads the initial model from a built-in set of models at solar metallicity [81]. The metallicity (and initial Y) can then be changed with an initial “pseudo-evolution” (i.e. evolution at fixed time). There are 32 initial models in the built-in set, with masses in the range $[0.8M_{\odot}, 100M_{\odot}]$, and MESA interpolates between those to create ZAMS stars of any requested mass (see the file `$MESA_DIR/data/star_data/zams_z2m2_y28.data`).

2.1.2 Run time

MESA is a very efficient tool for many stellar astrophysics problems not involving extreme regimes (such as the late stages of nuclear burning, the Helium flash in low mass stars, etc.). For example, Fig. 2.4 compares the run-times of three stars computed from ZAMS to TAMS. The run times are less than a minute in each case. The $M_{\text{ZAMS}} = 1M_{\odot}$ model (red curve) is computed in 83 timesteps (8.863 seconds), the $M_{\text{ZAMS}} = 2M_{\odot}$ model (orange curve) is computed in 128 timesteps (13.681 seconds), and the $M_{\text{ZAMS}} = 15M_{\odot}$ (blue curve) model in 65 timesteps (8.279 seconds). Note that the $M_{\text{ZAMS}} = 1M_{\odot}$ model is fast and takes

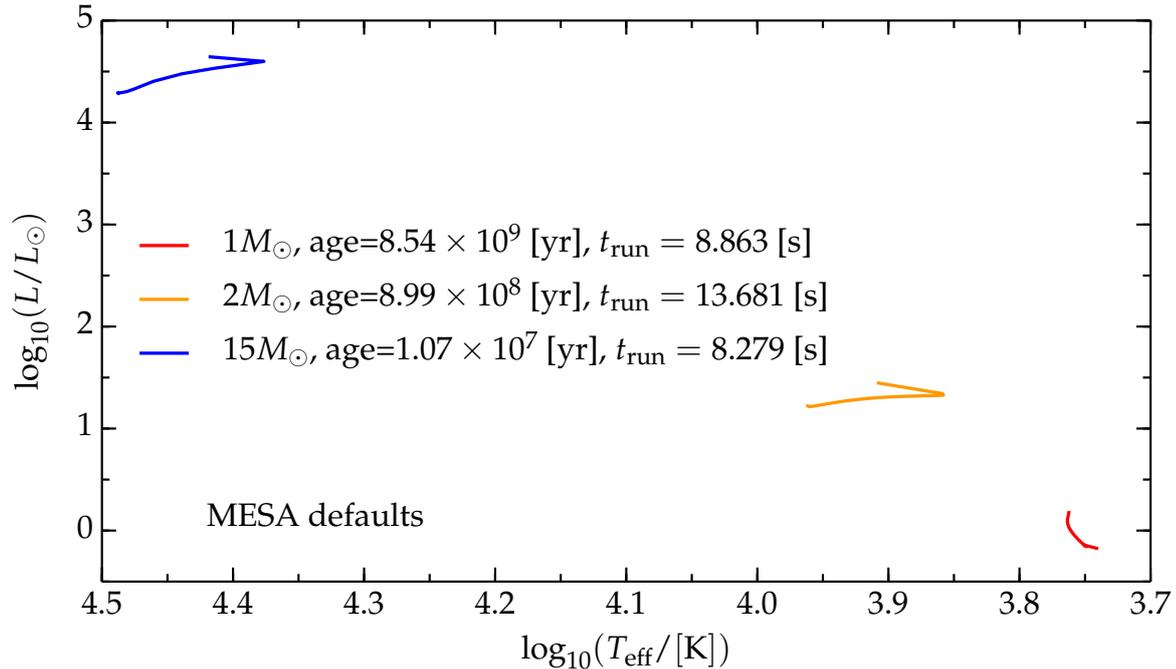


Figure 2.4: Comparison between the run times for the main sequence evolution of three Z_{\odot} stars with different M_{ZAMS} . The models in this figure are computed with the default MESA settings (see `$MESA_DIR/star/controls_defaults` and `$MESA_DIR/star/inlist_massive_defaults` for $M_{\text{ZAMS}} = 15M_{\odot}$) on a quad-core machine using GFortran of the Department of Physics at the University of Pisa.

a relatively small number of timesteps to be computed because there is a built-in ZAMS model of this mass, while for the $M_{\text{ZAMS}} = 15M_{\odot}$ model the speed up comes from the use of the default settings for massive stars, which have looser spatial and temporal resolution.

The use of higher resolution or larger nuclear reactions networks¹⁰, or the computation of more challenging evolutionary stages (see e.g. §1.3), can take much longer than the run-time shown in Fig. 2.4, often up to several days of computation.

2.1.3 MLT++

As mentioned in §1.3.1, the first challenging obstacle found in the simulation of massive stars is the presence of radiation dominated, nearly super-Eddington convective envelopes. In this environment, a numerical (and probably physical) instability develops unstable gas density and pressure inversions.

A hydrostatic code, such as MESA, tries to take smaller timesteps to deal with the density inversion (which, if it really forms in nature, is likely to develop a dynamical Rayleigh-Taylor instability), and it becomes impossible to pass this stage. This problem affects stars with $M_{\text{ZAMS}} \geq 20M_{\odot}$ and becomes more severe for higher initial masses [39], so of the set of models presented here, only those with $M_{\text{ZAMS}} = 15M_{\odot}$ avoid it.

To push the evolution through stages in which this problem arises, MESA adopts the following artificial workaround called “MLT++” – which must *not* be interpreted as a physical

¹⁰Higher spatial resolution and a larger nuclear network both increase the dimensions of the matrix partially shown in Fig. 2.2, while higher temporal resolution increases the number of computational steps for a given physical time.

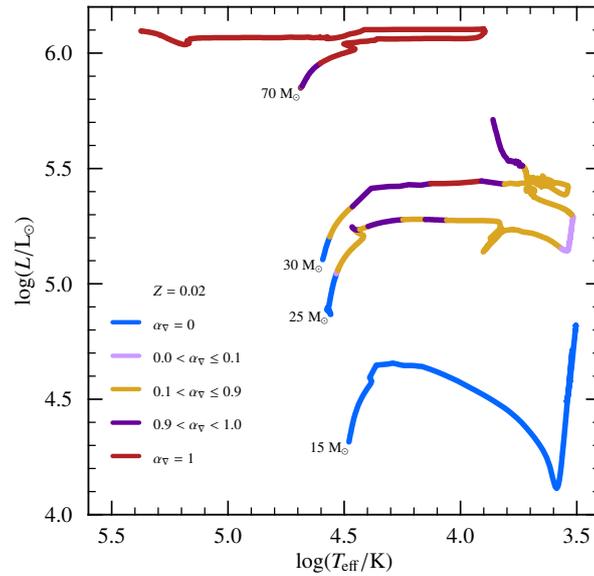


Figure 2.5: HR diagrams for different massive stars. The colors indicate the values of α_{∇} which is automatically computed at each timestep, see Eq. 2.13. Higher values correspond to a larger reduction of the superadiabaticity (which is the term on the left-hand side of Eq. 2.13). This is Fig. 42 of [39].

solution to the problem. The underlying idea is that when convection and radiation cannot provide the required energy flux (see also §1.3.1), another mechanism (or a different treatment of convection) may provide the flux missing to prevent the formation of the density inversion¹¹ (i.e. to maintain $L_{\text{rad}} < L_{\text{inv}}$, cf. Eq. 1.20, see §1.3.1). However, even if such a mechanism exists, no physical model is presently available for it. Therefore, when MESA detects the onset of a nearly super Eddington, radiation dominated convective envelope, the extra flux is attributed to convection (effectively increasing the free parameter α_{mlt}).

In MESA, the onset of a nearly super Eddington radiation dominated envelope is defined by (see §7.2 in [39] and the routine `$MESA_DIR/star/private/mlt.info.f` in a standard installation of the code)

$$\begin{cases} \beta \stackrel{\text{def}}{=} \frac{P_{\text{gas}}}{P_{\text{tot}}} \leq 0.35 , \\ \Gamma_{\text{E}} \stackrel{\text{def}}{=} \frac{L}{L_{\text{Edd}}} \geq 0.5 . \end{cases} \quad (2.12)$$

When the conditions of Eq. 2.12 are met and the superadiabaticity (i.e. the difference between the local temperature gradient and the adiabatic temperature gradient, see Eq. 2.13) is larger than some user-specified threshold (whose default is 10^{-3}), MESA modifies the temperature gradient $\nabla_{\text{loc}} \stackrel{\text{def}}{=} d \log(T) / d \log(P)$ to reduce the superadiabaticity and enforce a (quasi-)adiabatic stratification. This prevents the onset of a gas density and pressure inversion and allows MESA to avoid the numerical (and maybe physical) instability. More specifically, ∇_{loc} is modified in such way that the superadiabaticity

$$\nabla_{\text{loc}} - \nabla_{\text{ad}} \rightarrow \alpha_{\nabla} f_{\nabla} \cdot (\nabla_{\text{loc}} - \nabla_{\text{ad}}) \sim 0 , \quad (2.13)$$

where $\alpha_{\nabla} \equiv \alpha_{\nabla}(\beta, \Gamma_{\text{E}})$ is an efficiency factor that progressively turns on the MLT++ as L

¹¹The name “MLT++” stands for “MLT plus one” in a C-like syntax (Paxton, private communication). The “plus one” mechanism may be a different physical process, or even simply a larger convective-flux obtained with a multidimensional treatment of convection.

approaches L_{Edd} (the “modified” Eddington luminosity, see Eq. 1.14) in the radiation dominated region; $f_{\nabla} \ll 1$ is the maximum permitted reduction of the superadiabaticity.

The drawback of MLT++ is that this procedure affects the stellar surface luminosity and effective temperature (or radius), and consequently the mass loss rate. This approach also neglects unknown mass loss mechanisms, e.g. the instability of a nearly super Eddington radiation dominated envelope could be physical and trigger, at least in some cases, eruptive mass loss events. Fig. 2.5 shows the value of α_{∇} in different stages of the evolution for models with different M_{ZAMS} .

2.2 Modifications to the MESA Code

The design of MESA allows the user to call “personal” routines without the need of modifying the bulk of the code. This can be done locally in each directory from which the code is run, inserting the user-provided routines in the `run_star_extras.f` file. An extensive documentation on the use of this tool has been produced by the community of MESA users (see http://mesa.sourceforge.net/run_star_extras.html#toc-1-1). The reader is referred to the official MESA websites (<http://mesa.sourceforge.net/> and <http://mesastar.org/>) and mailing list (<http://sourceforge.net/p/mesa/mailman/>) for more details and support, respectively.

2.2.1 Implementation of Customized Timestep Controls

The timestep selection algorithm described in §2.1.1 does not explicitly check if the MESA timestep is smaller than the various (global) timescales of the star¹². Some preliminary tests showed that this timestep can indeed exceed the Kelvin-Helmholtz timescale, especially during the RSG phase. To avoid this, I implemented a customized timestep control (see §A.2.13). In particular, I tightened the default MESA timestep controls by manually enforcing

$$\Delta t_{n+1} \leq \min\{t_{\text{KH}}, t_{\dot{M}}, t_{\text{nuc}}, t_{\text{chem}}, t_{\gamma}\} , \quad (2.14)$$

where Δt_{n+1} is the timestep proposed at the end of the $n - th$ step for the next $(n + 1) - th$ step, and all the timescales on the right hand side refer to the $n - th$ step.

On the right hand side of the condition (2.14),

$$t_{\text{KH}} \stackrel{\text{def}}{=} \frac{3}{4} \frac{GM^2}{RL} , \quad (2.15)$$

is the Kelvin-Helmholtz timescale based on the global variables (which can be smaller than the timestep coming from the default implementation of the timestep controls in MESA, release 6794);

$$t_{\dot{M}} \stackrel{\text{def}}{=} \frac{M}{|\dot{M}|} , \quad (2.16)$$

is the mass change (loss through winds or gain from accretion) timescale;

$$t_{\text{nuc}} \propto \frac{M}{L/L_{\odot}} , \quad (2.17)$$

¹²In more recent MESA releases this check is now done explicitly, although the implementation is less strict than the one presented here.

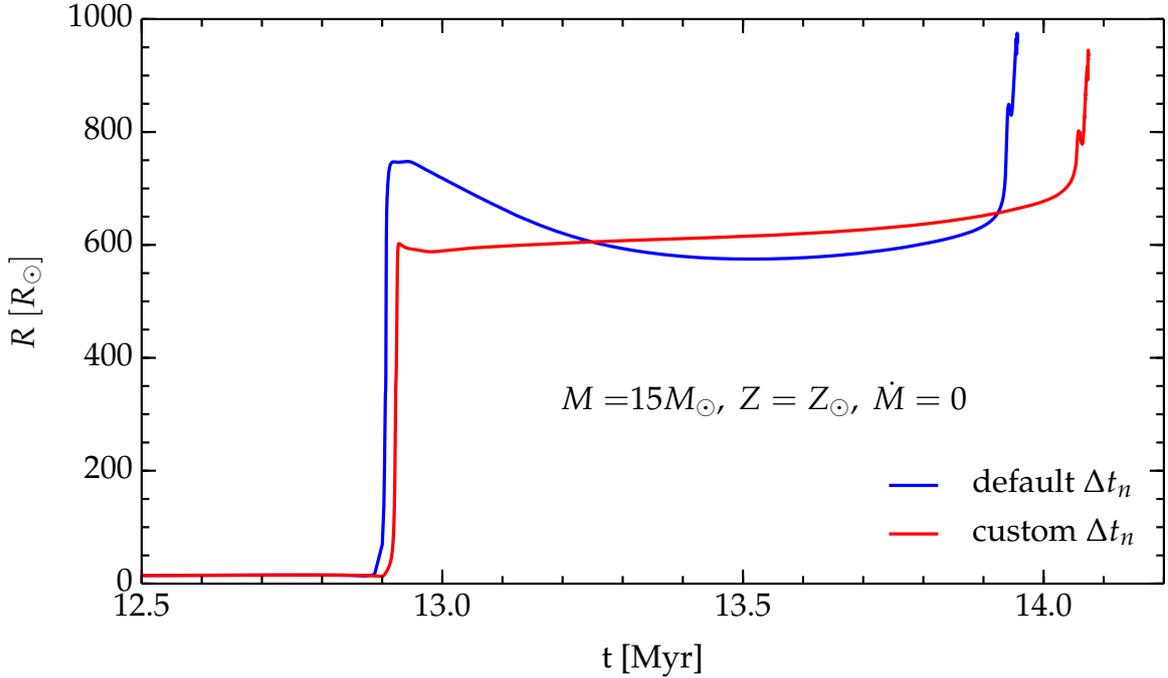


Figure 2.6: Radius evolution for a $M_{\text{ZAMS}} = 15M_{\odot}$, Z_{\odot} simulation without mass loss. The blue curve is computed with the default timestep control of MESA, revision 6794, the red curve uses the custom timestep (cf. 2.14).

is the nuclear timescale;

$$t_{\text{chem}} \stackrel{\text{def}}{=} \min_{k,i} \left[\frac{X_{i,k}}{\dot{X}_{i,k}|_{\text{burn}} + \dot{X}_{i,k}|_{\text{mix}}} \right], \quad (2.18)$$

is the chemical evolution timescale, where the denominator is the sum of the time derivatives of the abundances because of nuclear burning and mixing processes. Finally,

$$t_{\gamma} \stackrel{\text{def}}{=} \left(\frac{R - R_{\text{conv}}}{\lambda_{\gamma}} \right)^2 (\delta t), \quad (2.19)$$

is the photon diffusion time across the radiative portion of the envelope (whose outer and inner radii are the stellar radius $R \equiv R(m = M_{\text{tot}})$, and the convective core radius R_{conv} ¹³, respectively), assuming a random walk of the photons in the radial direction, with the same probability of moving inward or outward at each step and with mean free path λ_{γ} . The assumed time-scale for a single step of the photon (absorption/re-emission process) is the inverse of the probability per unit time of an atomic/ionic transition induced by the perturbation on the atomic hamiltonian by the photon, $\delta t \sim B_{fi}^{-1} \equiv 10^{-8}$ s, where B_{fi} is the Einstein coefficient for stimulated emission. Note that δt is set by hand. t_{γ} is an upper limit on the timescale on which the radiative envelope readjusts to surface phenomena such as mass loss, because not necessarily all of the envelope down to the edge of the convective core is affected by what happens at the surface.

¹³When the core is radiative, $R_{\text{conv}} = 0$.

Note that the right-hand side of (2.14) is evaluated with the quantities of the $n - th$ step, but it limits the $(n + 1) - th$ step (the timestepping is forward). I find that, among all the quantities on the right-hand side of (2.14), only the Kelvin-Helmholtz timescale t_{KH} effectively limits the timestep. All other timescales are always larger than the timestep proposed, so the timestep is never reduced because of them.

Figure 2.6 shows the effect of the improved custom timestep control on the radius during the SGB/RSG phase of a $M_{ZAMS} = 15M_{\odot}$, Z_{\odot} star computed without mass loss. Even if the difference is not enormous, the improvement is meaningful, especially during the RSG phase, when the mass loss rate is high (cf. Fig. 1.9) and depends on R (either directly or through a T_{eff} and L dependence).

2.2.2 Implementation of the Mass Loss Prescriptions

The wind mass loss of massive stars is usually divided into three separate phases whose definition is somewhat arbitrary. In fact, in nature, the evolution is continuous - there is no such thing as a threshold at which the mass loss rate changes abruptly. However, for computational purposes, thresholds are needed to define the domain of each mass loss rate algorithm. I adopted the following scheme, based on T_{eff} and the surface hydrogen abundance X_s , when the Kudritzki *et al.* algorithm is used during the hot phase:

- **Cool phase:** $T_{\text{eff}} < 15000 \text{ K};$
- **Hot phase (Kudritzki *et al.* only):** $T_{\text{eff}} \geq 15000 \text{ K};$
- **WR phase:** $X_s < 0.4, \forall T_{\text{eff}}.$

When the Vink *et al.* algorithm is used in the hot phase, slightly different thresholds are adopted, following the suggestions of [67]:

- **Cool phase:** $T_{\text{eff}} \leq 10000 \text{ K};$
- **Linear interpolation:** $10000 \text{ K} < T_{\text{eff}} < 11000 \text{ K};$
- **Hot phase (Vink *et al.* only):** $T_{\text{eff}} \geq 11000 \text{ K};$
- **WR phase:** $X_s < 0.4, \forall T_{\text{eff}}.$

That is, when the Vink *et al.* algorithm is used during the hot phase of the evolution, a linear interpolation between it and the cool wind mass loss rate is used to transition smoothly into the cool phase of the evolution.

These definitions are similar to those commonly adopted in the literature (see e.g. [18, 86, 87]). Note that the definition of the WR phase is the presence of a hydrogen-poor surface of the star. This is very different from the observational definition of a WR star, which is based on spectral features (hydrogen depletion *and* broad emission lines) that are not synthesized by stellar evolution codes.

The threshold chosen to divide the cool and hot phase is motivated by the fact that the radiation pressure is proportional to the product of the opacity and the flux (cf. Eq. 1.16), which peaks between 10000 – 15000 K because of iron recombination. Therefore, this temperature is a physically meaningful threshold for switching between wind mass loss algorithms.

Table 2.1: Combination of wind mass loss algorithm used in the present work.

Hot phase		Cool phase		WR phase		ID
Vink <i>et al.</i>	§1.4.5	de Jager <i>et al.</i>	§1.4.2	Nugis & Lamers	§1.4.7	VdJNL
Vink <i>et al.</i>	§1.4.5	de Jager <i>et al.</i>	§1.4.2	Hamann <i>et al.</i>	§1.4.8	VdJH
Vink <i>et al.</i>	§1.4.5	Nieuwenhuijzen <i>et al.</i>	§1.4.3	Nugis & Lamers	§1.4.7	VNJNL
Vink <i>et al.</i>	§1.4.5	Nieuwenhuijzen <i>et al.</i>	§1.4.3	Hamann <i>et al.</i>	§1.4.8	VNJH
Vink <i>et al.</i>	§1.4.5	van Loon <i>et al.</i>	§1.4.4	Hamann <i>et al.</i>	§1.4.8	VvLH
Vink <i>et al.</i>	§1.4.5	van Loon <i>et al.</i>	§1.4.4	Nugis & Lamers	§1.4.7	VvLNL
Kudritzkitzki <i>et al.</i>	§1.4.6	de Jager <i>et al.</i>	§1.4.2	Nugis & Lamers	§1.4.7	KdJNL
Kudritzkitzki <i>et al.</i>	§1.4.6	de Jager <i>et al.</i>	§1.4.2	Hamann <i>et al.</i>	§1.4.8	KdJH
Kudritzkitzki <i>et al.</i>	§1.4.6	Nieuwenhuijzen <i>et al.</i>	§1.4.3	Nugis & Lamers	§1.4.7	KNJNL
Kudritzkitzki <i>et al.</i>	§1.4.6	Nieuwenhuijzen <i>et al.</i>	§1.4.3	Hamann <i>et al.</i>	§1.4.8	KNJH
Kudritzkitzki <i>et al.</i>	§1.4.6	van Loon <i>et al.</i>	§1.4.4	Nugis & Lamers	§1.4.7	KvLNL
Kudritzkitzki <i>et al.</i>	§1.4.6	van Loon <i>et al.</i>	§1.4.4	Hamann <i>et al.</i>	§1.4.8	KvLH

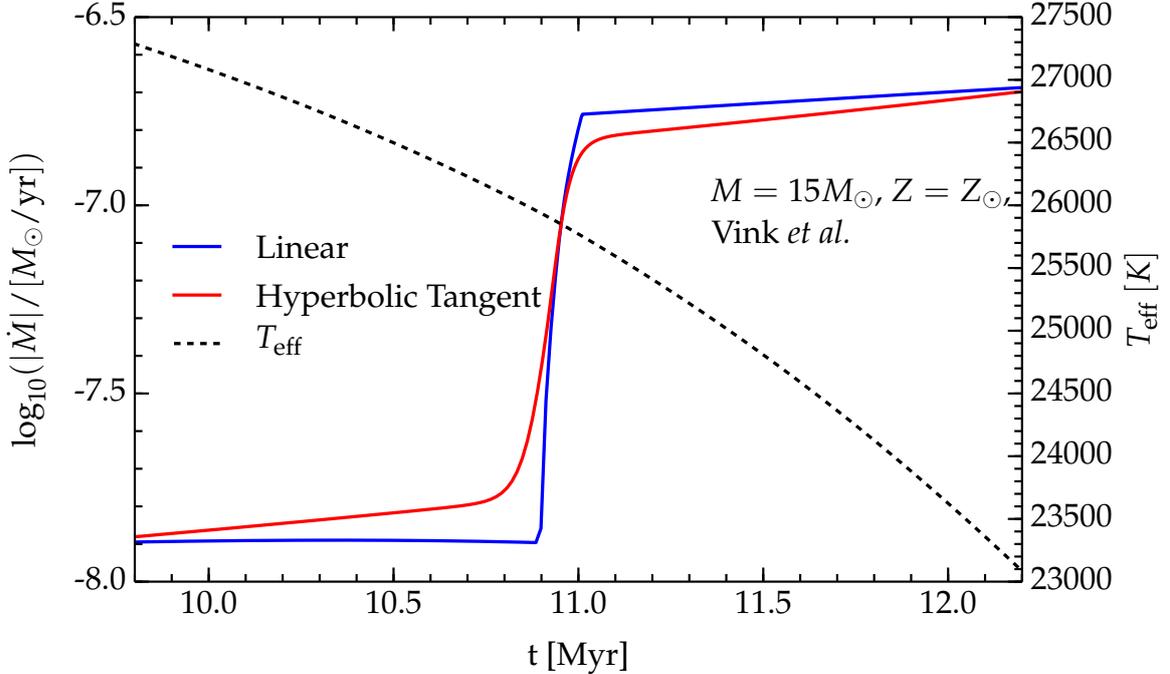


Figure 2.7: Mass loss rate as a function of time around the bistability jump of the Vink *et al.* mass loss rate (§1.4.5) in a $M = 15M_{\odot}$, $Z = Z_{\odot}$ model. The black dashed line is the effective temperature T_{eff} , reached near TAMS in these simulations. The blue curve corresponds to the mass loss rate yielded by the default MESA routine, the red curve (smoother) is the mass loss rate from the customized mass loss routine described in §2.2.2.

Each algorithm combination used in this work is built selecting a specific wind mass loss rate for each of the phases described above. The combinations used are summarized in Tab. 2.1. I give in Tab. 2.2 the scaling of each mass loss rate with the considered stellar parameters so that the reader can easily compare the sensitivity of each algorithm to different quantities.

Not all wind mass loss rates compared in this work are part of the default release of MESA. To modify the criterion to switch between different algorithms (according to the def-

Table 2.2: \dot{M} scaling with physical quantities (luminosity L , mass M , escape velocity v_{esc} and final velocity v_{∞} of the wind, thermal velocity v_{th} , effective temperature T_{eff} , radius R , Eddington ratio $\Gamma_{\text{E}} = L/L_{\text{Edd}}$, surface hydrogen and helium abundances and metallicity, X_s , Y_s , and Z_s , respectively). The scaling are obtained from the algorithms described in §1.4, and errors and overall multiplying factor are omitted for the sake of brevity.

ID	VdJNL	VNJNL	VvLNL	VdJH	VNJH	VvLH
Hot	$L^{2.210} M^{-1.339} \left(\frac{v_{\infty}}{2v_{\text{esc}}}\right)^{-1.601} T_{\text{eff}}^{1.07} Z^{0.85}$					if $T_{\text{eff}} < 22.5$ kK
	$L^{2.194} M^{-1.313} \left(\frac{v_{\infty}}{2v_{\text{esc}}}\right)^{-1.226} Z^{0.85} T_{\text{eff}}^{(0.933-10.92 \log(T_{\text{eff}}/40\text{kK}))}$					if $T_{\text{eff}} > 27.5$ kK
Cool	$L^{1.769} T_{\text{eff}}^{-1.676}$	$L^{1.24} M^{0.16} R^{0.81}$	$L^{1.05} T_{\text{eff}}^{-6.3}$	$L^{1.769} T_{\text{eff}}^{-1.676}$	$L^{1.24} M^{0.16} R^{0.81}$	$L^{1.05} T_{\text{eff}}^{-6.3}$
WR	$L^{1.29} Y_s^{1.73} Z_s^{0.47}$			$L^{1.5} 10^{-2.85 X_s}$		if $\log_{10}(L/L_{\odot}) > 4.5$
				$L^{6.8}$		if $\log_{10}(L/L_{\odot}) \leq 4.5$
ID	KdJNL	KNJNL	KvLNL	KdJH	KNJH	KvLH
Hot	$L^{1.779} v_{\text{th}}^{-1.169} [GM(1 - \Gamma_{\text{E}})]^{0.610}$					
Cool	$L^{1.769} T_{\text{eff}}^{-1.676}$	$L^{1.24} M^{0.16} R^{0.81}$	$L^{1.05} T_{\text{eff}}^{-6.3}$	$L^{1.769} T_{\text{eff}}^{-1.676}$	$L^{1.24} M^{0.16} R^{0.81}$	$L^{1.05} T_{\text{eff}}^{-6.3}$
WR	$L^{1.29} Y_s^{1.73} Z_s^{0.47}$			$L^{1.5} 10^{-2.85 X_s}$		if $\log_{10}(L/L_{\odot}) > 4.5$
				$L^{6.8}$		if $\log_{10}(L/L_{\odot}) \leq 4.5$

initions above) and control separately the efficiency of each algorithm, I implemented customized mass loss routines for each wind scheme employed in the `run_star_extras.f` (see §A.2.1–A.2.12). The formulae implemented for each wind algorithm are the ones provided in §1.4, but the errors (when available) are neglected¹⁴.

When possible (i.e. for the de Jager *et al.*, Nieuwenhuijzen *et al.*, van Loon *et al.*, Nugis & Lamers, and Kudritzki *et al.* algorithms) the custom mass loss routine calls MESA wind routines, which are copied in the `run_star_extras.f` from `$MESA_DIR/star/private/winds.f`. To be able to use the built-in routine for the Kudritzki *et al.* wind scheme, the MESA module `kuma.mod` is copied from into `$MESA_DIR/star/make` to `$MESA_DIR/include`.

¹⁴This is common in stellar astrophysics applications of these formulae. The sensitivity to each coefficient in the formulae could, in principle, be studied by measuring the rate of divergence between the evolutionary tracks computed varying one coefficient within its error.

The Vink *et al.* algorithm also has a built-in routine in MESA, but it uses a linear interpolation between the formula for above the bistability jump (Eq. 1.47) and below (Eq. 1.48). This approach gives a very steep, unphysical jump between the two regimes (thus, $M(t) \notin \mathcal{C}^2$). As an improvement, I implemented a hyperbolic tangent interpolation, which yields a smoother jump, see Fig. 2.7.

To study the effects of the possible overestimate of the mass loss rate caused, for example, by wind inhomogeneities (the “clumps”), I use three different values of the wind efficiencies $\eta = 1.0, 0.33, 0.1$. I use the same η during the entire evolution of each model, even if the wind scheme routines implemented permit the use of a different efficiency factor for each phase. The values of $\eta \equiv \sqrt{f_{cl}}$ (see §1.4.1) are those suggested in [7], and they span the range of clumping factors f_{cl} available in the literature.

2.3 Systematic Comparison of Wind Mass Loss Algorithms

To explore the effects on the stellar structure of different parametrizations and efficiencies of the wind mass loss, I ran a grid of solar chemical composition ($Z_{\odot} \equiv 0.019$ [25, 64], $Y_{\odot} = 0.27$ [39]) stellar models with $M_{ZAMS} = \{15, 20, 25, 30\}M_{\odot}$. The mass range explored is limited by the effects of the MLT++ (see §2.1.3), which are found to cause rapid and unphysical oscillation of the surface L , T_{eff} and R , that influence the mass loss rate and invalidate the comparison between different mass loss algorithms (see also §B.2). Such oscillations are found in all models needing the MLT++ workaround ($M_{ZAMS} \geq 20M_{\odot}$) when using the default outer boundary condition. The default atmospheric boundary condition, `simple_atm`, estimates the location of the photosphere (defined by $\tau = 2/3$), and its location is not accurate enough for the purposes of this study (see §B.2). It causes significant variations of the surface location between subsequent steps, that produce large and unphysical variations of the luminosity, effective temperature and radius of the star, causing variations in the mass loss rate. To reduce these oscillations of the surface properties, and eliminate them for $M_{ZAMS} \leq 30M_{\odot}$, I adopted an Eddington grey atmospheric model to evaluate the outer boundary pressure. This is done by setting `which_atm_option = Eddington_grey` in the parameter file (`inlist`, see §A.1.1). The Eddington grey atmospheric model assumes for the outer portion of the star

$$T^4 = \frac{3}{4}T_{\text{eff}}^4\left(\tau + \frac{2}{3}\right), \quad (2.20)$$

and integrates the stellar structure equations from the outermost cell to the photosphere, using τ as the independent variable, to find the photospheric boundary pressure.

For each initial mass, I computed simulations with each wind mass loss combination listed in Tab. 2.1, and for each wind mass loss combination I used three different efficiency factors $\eta = 1.0, 0.33, 1.0$. Since the efficiency factor η does not have a physical interpretation based on first principles, these values are chosen (arbitrarily) to test the suggestions of [7, 49]. According to [7, 49], wind mass loss rates are overestimated by a factor between 2 and 10, with 3 as the best guess for the overestimation factor.

The simulations are run from ZAMS¹⁵ to core oxygen depletion, defined by

$$X_c(^{16}\text{O}) \leq 0.4, \quad (2.21)$$

¹⁵However, the initial composition is slightly different from the composition of the built-in starting models. Therefore a short phase of readjustment occurs at the beginning of each run. This phase can be seen as a small loop in the low-luminosity, high-temperature corner of the HR diagrams.

[19]. However, since the condition Eq. 2.21 is already true at the beginning of the calculation, I check it only when a small amount of ^{28}Si is produced in the core (i.e. when some oxygen has been burned, $X_c(^{28}\text{Si}) \geq 0.01$). Note that ^{16}O and ^{28}Si are the only isotopes of oxygen and silicon included in the 21-isotope nuclear network used (`approx21.net`), since its purpose is not to follow the details of the nucleosynthetic processes. The stopping criterion is implemented in the `run_star_extras.f`, see §A.2.13.

After oxygen depletion, wind mass loss can be neglected¹⁶ because the evolution is so rapid that there is not enough time for it to remove a significant fraction of the mass. Moreover, post-oxygen depletion evolution cannot be simulated realistically with a 21-isotope nuclear network (see §B.3). Finally, to improve numerical stability, MESA artificially progressively decreases the mass loss rate when the central temperature $T_c \geq 10^9$ K, until it is completely turned off for $T_c \geq 2 \times 10^9$ K. This does not modify significantly the amount of mass shed, since the physical time spent at $T_c \geq 10^9$ K before oxygen depletion is also extremely short. I ran some test case without the artificial decrease of \dot{M} and until the onset of core collapse to confirm that the final mass of the star does not change after the condition Eq. 2.21 is reached. Note that the post-oxygen depletion evolution of the star may amplify minor features in the core structures caused by mass loss in early stages, [19].

To restart the stellar simulations with a guaranteed bit-to-bit identical evolution from oxygen depletion, I saved binary files (MESA “photos”) at the end of each run. To do this, I copy the module `model_out.mod` in `$MESA_DIR/include`.

I use the Ledoux criterion for convection, i.e. I consider as convectively unstable regions where

$$\nabla_{\text{loc}} \geq \nabla_{\text{ad}} - \frac{\varphi}{\delta} \nabla \mu, \quad (2.22)$$

where $\nabla \mu \stackrel{\text{def}}{=} d \log(\mu) / d \log(P)$, and φ and δ are positive constants depending on the EOS, see [21]. This criterion considers the effect of the chemical stratification on the convective stability of a region. I follow the suggestions of [19] for treating convective mixing and overshooting, but I also include thermohaline mixing (i.e. slow mixing on the thermal timescale in regions with a mean molecular weight inversion) using the default MESA option. This corresponds to the following values of the mixing length parameter α_{mlt} , the exponentially decreasing overshooting parameter f_{ov} (see Eq. 2 in [81]), semiconvection efficiency, and thermohaline coefficient:

- $\alpha_{\text{mlt}} = 2.0$;
- $f_{\text{ov}} = 0.025$ (for all the overshooting and undershooting regions);
- semiconvection with efficiency 0.01;
- thermohaline according to [88] with coefficient 2.

I carried out extensive work on the optimum choice of the spatial and temporal resolution for this study. I used two test-case models of $M_{\text{ZAMS}} = 15M_{\odot}$ and $M_{\text{ZAMS}} = 25M_{\odot}$ with the Vink *et al.*, de Jager *et al.* wind combination and $\eta = 1.0$, to test the convergence of both models with and without MLT++. Each test case was run twice, once with `mesh_delta_coeff = mesh_delta_coeff_for_highT = 1.0`, and once with `mesh_delta_coeff = mesh_delta_coeff_for_highT ≤ 0.2`.

¹⁶Other groups stop their calculations even earlier, at carbon depletion [45].

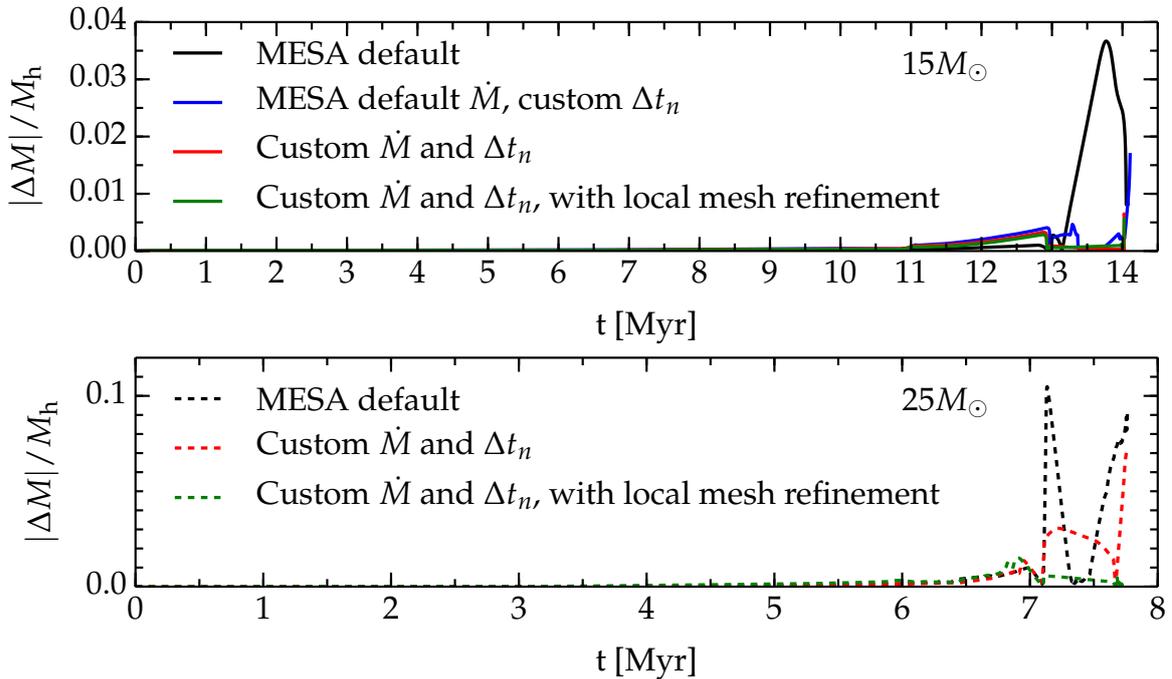


Figure 2.8: Relative mass difference for the test case models as a function of time. The upper (lower) panel is for the test case model with $M_{\text{ZAMS}} = 15M_{\odot}$ ($25M_{\odot}$), respectively. $\Delta M \stackrel{\text{def}}{=} M_{\text{std}} - M_{\text{h}}$ is the difference between the total mass at standard resolution (`mesh_delta_coeff=1.0`) and the total mass at five times higher resolution. The mass loss scheme used for these test cases is the combination of the Vink *et al.* rate and de Jager *et al.* rate with $\eta = 1.0$. The implementation of the custom version of the former rate is described in §2.2.2. The implementation of the custom timestep control is described in §2.2.1. The “local mesh refinement” is described in §2.3 and §A.2. Note the different scales on the two panels: the relative mass difference is higher at higher M_{ZAMS} .

Preliminary tests showed that the use of higher spatial resolution (i.e. lower value of `mesh_delta_coeff`, see §2.1) yielded significantly different results for the total mass at oxygen depletion $M_{\text{O depl}}$ and the structure of the star. To make sure the customized routine used were not causing this problem, I re-ran the test cases using the default timestep control, first without mass loss, then using the default mass loss (“Dutch” wind scheme, similar to VdJNL except for the implementation of the Vink *et al.* rate), and then progressively turned on the customized routines for the timestep control and the mass loss algorithm.

Using the customized timestep control (see §2.2.1), and the hyperbolic tangent interpolation of the Vink *et al.* mass loss rate (see §2.2.2) slightly improved the convergence, see Fig. 2.8. However, the level of convergence reached using only the customized routine was not sufficient, especially for the $M_{\text{ZAMS}} = 25M_{\odot}$ test case, where differences greater than $\sim 7\%$ were still found in the final mass, see Fig. 2.8.

Further investigations showed that “local mesh refinements” (see below) can decrease the discrepancies and resolve the convergence issue, at least for the surface variables which enter in the determination of the mass loss rate. However, significant quantitative differences are still present in the late $T_c \equiv T_c(\rho_c)$ tracks of the test case models (see §B.3.3). Such differences are not worrying for the present purposes, since they arise at $T_c \geq 10^9$ K (i.e. when the artificial mass loss decrease is started) and they occur in an evolutionary phase too rapid to

have effects at the stellar surface. Note that a proper simulation of the oxygen burning stage requires ~ 40 isotopes in the nuclear network (Arnett, private communication), therefore it is possible that the differences resulting from changing the resolution would disappear with a more physically consistent treatment of the core nuclear burning. The “local mesh refinement” (see §A.1.1) adopted increases the resolution in the following regions:

- **Stellar surface:** I constrain the variation of $\log_{10}(T)$ at the surface by setting `T_function2_weight = 110` (instead of the default value 0) in the `inlist`. This makes the outermost computational cell of the star small independent of the value of `mesh_delta_coeff`, ensuring that X_s is the same both in the high resolution and normal resolution runs, and that the switch to the WR wind algorithm does not depend on resolution, see §A.1.1;
- **Edges of burning regions:** To increase the resolution at the boundaries of regions where nuclear burning occurs, I constrain the spatial variation of $d \log(\epsilon_{\text{nuc}}) / d \log(P)$ by setting a multiplying factor to 0.15 (instead of the default value of 1) for the maximum change allowed for $d \log(\epsilon_{\text{nuc}}) / d \log(P)$ for each nuclear burning chain (PP, CNO, 3α , C burning, etc...), see §A.1.1;
- **Edge of cores of different composition:** I reduce the maximum spatial variation allowed for the abundances of several isotopes (^1H , ^4He , ^{12}C , ^{16}O , ^{20}Ne , ^{28}Si , ^{32}S , ^{54}Fe , ^{56}Fe). This ensures that I properly resolve the edges of the helium and carbon/oxygen cores. Specifically, I add the quantity $20 \times \log_{10}(X_i + 0.01)$ to those determining the merging or splitting of cells, see §A.1.1.

2.4 Simplified Simulation of Envelope Shedding Mass Loss Events

Motivated by the fundamental uncertainties in how mass loss proceeds for SN progenitors (discussed in §1.4.9), I used MESA to produce stellar models that have undergone mass loss that mimics the rapid processes mentioned in §1.4.9, such as violent pulsational instabilities, wave driven mass loss, RLOF in binaries, or any impulsive and catastrophic mass loss event. By impulsive event, I indicate any event that happens on a timescale shorter than the thermal timescale and/or the dynamical timescale, which therefore *must* happen out of thermal and hydrostatic equilibrium. The aim of this numerical experiment is to bypass the uncertainties in the mass loss process by *artificially* removing (part of) the hydrogen-rich envelope, and explore how impulsive mass loss influences the post-stripping evolution and the structure at the onset of core collapse, see Eq. 1.9. The simulated light curves produced by the explosion of the models described here are presented in Morozova *et al.*, submitted to ApJ, arXiv:1505.06746.

To produce pre-SN models with varying levels of mass stripping, I first computed a reference model with a zero-age main sequence (ZAMS) mass $M_{\text{ZAMS}} = 15M_{\odot}$ and solar metallicity $Z_{\odot} = 0.019$ [25, 64]. This M_{ZAMS} rules out all LBV-like phenomena, since LBVs should be much more massive [78]. This choice also allows me to avoid the problem of nearly super-Eddington radiation dominated convective envelopes (see §1.3.1 and §2.1.3).

In the reference model, as in all the stripped models, convection is treated as described in §2.3, except thermohaline mixing which is omitted. This does not influence significantly the results.

Besides the mass stripping, all models used a wind mass loss algorithm that combines the prescriptions of Vink *et al.* (for the hot phase), implemented according to §2.2.2, de Jager

et al. (for the cool phase), and Nugis & Lamers (for the WR phase), with efficiency $\eta = 1.0$. The reference model, and all stripped models, are computed using the default 21-isotope nuclear reaction network `approx21.net` until the onset of core collapse. The velocity of each computational cell was traced only near the onset of core collapse, i.e. from when $Y_e \leq 0.45$ in the center.

The temporal resolution was set using `varcontrol_target = 10-3` (which is the MESA default value for massive stars), and I used the customized timestep control described in §2.2.1. The spatial resolution was set by `mesh_delta_coeff = 1.0` and `mesh_delta_coeff_for_highT = 1.5`, which are the default values for massive stars, since the details of the core structure are not the main focus of this study.

The reason for using a coarser resolution at high temperature (i.e., in the core), was to reduce the run-time and obtain a solution to the stellar structure equations more easily. Even if this reduces the accuracy of the core structures, the 21-isotope nuclear reaction network cannot properly follow the late nuclear burning stages (neon, oxygen and silicon burning), see §B.3. In particular, this small network cannot properly account for the neutronization of matter. In `approx21.net` the neutronization treated approximately with a compound electron-capture reaction,



which is tuned to yield $Y_e(r=0) \equiv Y_e({}^{56}\text{Cr}) = 0.428$ in the center at the onset of core collapse. The choice of a single compound reaction to simulate neutronization is arbitrary, and the final Y_e obtained can differ of a few percent from the value obtained from calculations that use hundreds of active isotopes. This has a significant effect on the core structure, because Y_e enters quadratically in the effective Chandrasekhar mass, Eq. 1.7. However, since the explosion mechanism of SNe is still uncertain, see §1.2.4, most of the simulations dealing with light curves artificially inject energy (thermal bomb) or momentum (piston) in the stellar core to trigger the explosion (e.g. [13, 17] and references therein). For this, only the accuracy of the outer portion of the star is important, and the details of the core structure are irrelevant.

Fig. 2.9 shows the complete evolutionary track of the unstripped reference model on the HR diagram. At each of the marked points, I remove parts of the star’s envelope (see §2.4.1).

The star in Fig. 2.9 expands significantly during the red supergiant (RSG) phase. This expansion could trigger a possibly unstable RLOF event, were the star in a binary system, that quickly removes all or part of the hydrogen-rich envelope. Note, however, that these are non-rotating stellar models, and that any angular momentum loss and/or transfer that might happen in a RLOF event is completely neglected here. I stripped the star at three different times during the RSG phase to investigate whether the post-stripping evolution and the structure at the onset of core collapse are sensitive to when the mass loss occurs.

The physical conditions that define the moments when I stripped the star are summarized in Tab. 2.3 and are as follows:

- The *red circle* at $T_{\text{eff}} \equiv 10^4$ K corresponds to the middle of the subgiant branch (SGB). The models stripped here comprise the “mSGB” series.
- The *yellow triangle* marks the first moment when the stellar radius R exceeds half of its maximum value, i.e., $R \gtrsim 375 R_{\odot}$ (“hMR” series).
- The *cyan diamond* shows when the convective envelope stops its expansion and reaches its maximum extent (“MCE” series).

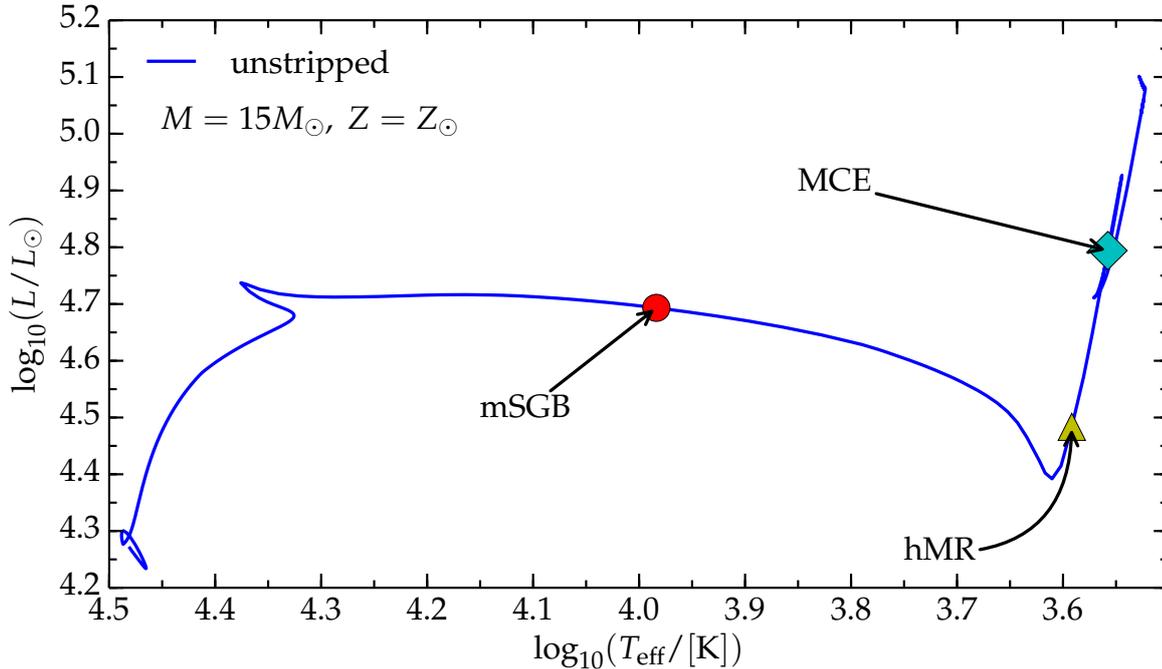


Figure 2.9: Evolutionary track on the HR diagram for the unstripped model used as reference for the comparison with the stripped models. Each marked point corresponds to one stripped series: the red dot corresponds to $T_{\text{eff}} = 10^4$ K (middle of SGB – mSGB), the yellow triangle to where the star reaches half of its maximum radius (hMR), and the cyan diamond to when the outer convective envelope stops expanding (MCE). The reference model is re-run from ZAMS to reach each point.

Table 2.3: Series of stripped models. The radius (R), age and helium core mass (M_{He}) at each stripping point are reported. X_c is the abundance of hydrogen in the central computational cell; $v_{\text{conv}}^{\text{surf}}$ is the unweighted average convective velocity in the outermost 150 computational cells, $v_{\text{conv}}^{\text{env}}$ is the unweighted average convective velocity in the 150 computational cells above the outermost lower boundary of a convective region. If these two differ by less than 10%, the envelope has roughly reached its maximum extent. Note that the radius at TAMS (age ~ 12.93 Myr) is only $\sim 1.2 R_{\odot}$.

Series Name	Stripping Moment	Age [Myr]	$R [R_{\odot}]$	$M_{\text{He}} [M_{\odot}]$
mSGB	$T_{\text{eff}} = 10^4$ K	13.02	79.8	3.81
hMR	$R \gtrsim 375 R_{\odot}$	13.03	381.6	3.87
MCE	$X_c = 0$ and $(v_{\text{conv}}^{\text{surf}} - v_{\text{conv}}^{\text{env}}) / v_{\text{conv}}^{\text{surf}} \leq 0.1$	13.03	638.1	3.88

The time difference between these evolutionary points is only $\sim 10^4$ years, which is small compared to the lifetime of the unstripped reference model, 1.413×10^7 years. However, it is large compared to the Kelvin-Helmholtz timescale ($t_{\text{KH}} \sim 1250$ years at mSGB, $t_{\text{KH}} \simeq 424$ years at hMR, and $t_{\text{KH}} \simeq 124$ years at MCE), and of course much larger than the global dynamic timescale (i.e. the free fall time). Therefore, the nuclear burning stage is the same for all the marked points¹⁷: the star is burning hydrogen in a shell, and has an almost

¹⁷In the paradigm of RLOF, these events would all correspond to a “case B” (i.e. post-main-sequence) mass transfer.

completely inert helium core. A tiny helium burning region exists in the center at hMR and MCE, but its energy generation rate is negligible.

The structure of the outermost envelope and the radius of the unstripped reference model (see Tab. 2.3) change much faster and are significantly different at the three stripping points. The structure, and, in particular, the temperature gradient in the outer region of the star, determine the star’s response to the mass loss event.

At mSGB most of the outer envelope is still radiative, and only a small region at the base of the envelope, between mass coordinate $6.5 \lesssim m/M_{\odot} \lesssim 5.5$, is convective. Therefore, the mSGB series is a prototype of mass stripping event happening when the envelope is (mostly) radiative.

At hMR, a shallow convective zone has formed in the outer envelope, which extends from the surface down to $m \sim 9.5M_{\odot}$, i.e. roughly in the outermost $5M_{\odot}$. The deeper convective layer that existed at mSGB is now stable and radiative. Thus, models of the hMR series explore the response to violent mass loss of an envelope that is only partially convective. Since the stopping criterion directly involves the radius of the star, it mimics more naturally the condition for a RLOF event. However, feedback on the Roche Lobe radius and the parameters of the binary are ignored, and the amount of mass removed is specified *by hand*. It is also questionable whether in a RLOF mass can be lost as fast as I imposed on these models.

Finally, by definition, at MCE the convective portion of the envelope has reached its maximum extent of $\sim 8M_{\odot}$, occupying the region from $m \simeq 6M_{\odot}$ to the surface. At this point, the temperature gradient in the outermost region is roughly adiabatic. Stars with adiabatic temperature gradients in their outer envelopes respond to mass loss by expanding, because of the reduction in the pressure at the outer boundary. In the RLOF paradigm, this leads to dynamically unstable mass loss, that removes the entire hydrogen envelope. The models in the MCE series are prototypes of impulsive mass loss events happening from a convective envelope.

2.4.1 The Stripping Process

To instantaneously remove mass from the reference model and mimic a fast and short mass loss event (such as an eruption, or unstable RLOF in a binary), I used the MESA module `adjust_mass`, by setting `relax_initial_mass = .true.` in the parameter file (`inlist`, see §A.1.2) for each run, and specified the new value of the desired total mass.

The new (smaller) value of the total mass is reached using 80 “pseudo-evolution” steps (i.e. the structure is evolved using timesteps, but the time coordinate is not updated and remains constant). During each step, MESA removes the largest amount of mass from the envelope that it can while still finding a hydrostatic solution to the equations of stellar structure. In most cases, of the 80 pseudo-evolution steps used, 75 are sufficient to reach the desired mass, and the last 5 pseudo-evolution steps have mass loss set to zero. Both during and at the end of the pseudo-evolution, the structure is always in global hydrostatic equilibrium, therefore, when the regular evolution resumes, no readjustment phase occurs.

At each of the points indicated in Fig. 2.9, I produce a series of stripped models. In each series, I remove mass in steps of $1M_{\odot}$ per model, until only a thin (radiative) layer remains above the hydrogen shell. The amount of mass to remove from each model was chosen arbitrarily: each $1M_{\odot}$ step removes a significant fraction of the total mass of the reference model, but it permits to remove only partially the envelope and explore how the structure

changes with varying stripping levels. The precise value of the mass of the thin layer left above the hydrogen burning shell depends on the stripping point considered, and it always falls in the $1M_{\odot}$ ballpark. If such layer were also removed, the hypothesis of hydrostatic equilibrium (embedded in the MESA code) would break down. To show this, consider the hydrostatic equilibrium condition (cf. Eq. 2.4 with $a_k = 0$)

$$\frac{dP}{dr} = -\frac{GM(r)\rho(r)}{r^2} \equiv g(r) , \quad (2.24)$$

which must hold at any location r in the star. For the location of the edge of the hydrogen burning shell (however it may be defined) $r = r_{\text{sh}}$, during the stripping process, the gravitational acceleration (per unit volume),

$$g_{\text{sh}} \stackrel{\text{def}}{=} -\frac{GM(r_{\text{sh}})\rho(r_{\text{sh}})}{r_{\text{sh}}} , \quad (2.25)$$

is (roughly) constant, since none of the quantities on the right-hand side is allowed to vary significantly (in nature, the event simulated happens on a timescale much shorter than the thermal timescale). Therefore, to maintain hydrostatic equilibrium, the pressure gradient must not change significantly at this location. The pressure gradient at the shell boundary can be approximated by

$$\left. \frac{dP}{dr} \right|_{\text{sh}} \sim \frac{P_{\text{sh}} - P_{\text{out}}}{R_{\text{sh}} - R_{\text{out}}} < 0 . \quad (2.26)$$

If a significant amount of mass is left outside the shell, $R_{\text{out}} \gg R_{\text{sh}}$ and $P_{\text{out}} \ll P_{\text{sh}}$ (the total pressure decreases outward), thus,

$$\left. \frac{dP}{dr} \right|_{\text{sh}} \sim -\frac{P_{\text{sh}}}{R_{\text{out}}} , \quad (2.27)$$

and both quantities on the right-hand side do not vary much during the stripping process (see also §4.2 and Tab. 4.1). On the other hand, if all the mass outside the hydrogen burning shell is removed or very little is left, then $R_{\text{out}} \sim R_{\text{sh}}$, while the pressures cannot vary much (they would on a thermal timescale, but the simulated event happens faster), therefore the pressure gradient at the shell boundaries increases, while the gravitational pull remains roughly constant. This is contrary to the hydrostatic equilibrium hypothesis, which cannot hold anymore: any result MESA would produce when all the mass outside the hydrogen shell is removed cannot be physically meaningful (see also §B.1).

Note that during the stripping process the energy generation in the star is *not* determined by the energy loss at the surface as would happen during the regular stellar evolution, but rather is almost constant¹⁸ and the outer layer must readjust to the energy generation rate.

The maximum amount of mass I was able to remove is limited by the global hydrostatic equilibrium condition. The amount of mass removed is indicated (in units of M_{\odot}) by the number in the model name in Table (4.3).

Once the mass is removed, each stripped star is then evolved to core collapse as described above. Note, that a model is not stripped more than once during its evolution. That is, each series of stripped models is produced by evolving the star from the ZAMS to the desired stripping point before any mass is stripped.

¹⁸Small variations are allowed to find a solution during the pseudo-evolution.

The procedure described above does not provide insight into the physical mechanism(s) triggering and powering the envelope (partial) shedding event. The conditions for its onset are chosen with a certain degree of arbitrariness, and the likelihood of such extreme mass loss events during these evolutionary stages may be questionable. These models are thus an attempt to study the response to an impulsive mass loss event, while still satisfying the very restrictive condition of global hydrostatic equilibrium. In principle, since this restriction does not exist in nature, this could result in a discrepancy between the actual and the simulated post-episodic mass loss structure and evolution. Therefore, this simple mass stripping procedure may be affected by a large systematic error resulting from the choice of timing for the stripping and the assumption of global hydrostatic equilibrium.

CHAPTER 3

Results: Wind Algorithms Comparison

*Le vent l' emportera
Tout disparaîtra, mais
Le vent nous portera*

[Noir Désir, “Le Vent Nous Portera”, Element of Crime]

3.1 Overview

In this chapter, I present the grid of models computed to compare and contrast the different wind mass loss algorithms available in the literature and commonly adopted for the simulation of massive star evolution. The aim of this study is *not* to determine which algorithm is in some sense “the best”, but rather to quantify and describe the differences in the stellar structure and evolution rising from different algorithmic treatments of the same physical phenomenon: the steady, radiatively driven driven wind mass loss (see also §1.4.1).

Figures 3.1 and 3.2 show the total mass as a function of time for $M_{\text{ZAMS}} = \{15, 20\} M_{\odot}$ and $M_{\text{ZAMS}} = \{25, 30\} M_{\odot}$, respectively. Note that the scale of each plot is different. The spread in the final masses increases for higher initial masses, resulting directly in a larger uncertainty in the mass loss and final mass of these stars. Another striking feature is the large spread in the masses and mass loss rates during the (post-TAMS) giant and supergiant phases, i.e. precisely when the mass loss rate is the highest and most of the mass is lost. The maximum difference between the final masses could be used to define systematic uncertainties for the final mass inferred from the simulation of massive stars. These are very large (up to $\sim 50\%$ of M_{ZAMS}), confirming that mass loss is (one of) the most important sources of uncertainty in the evolution of massive stars. This uncertainty as been heretofore explored only insufficiently, see e.g. [45, 46].

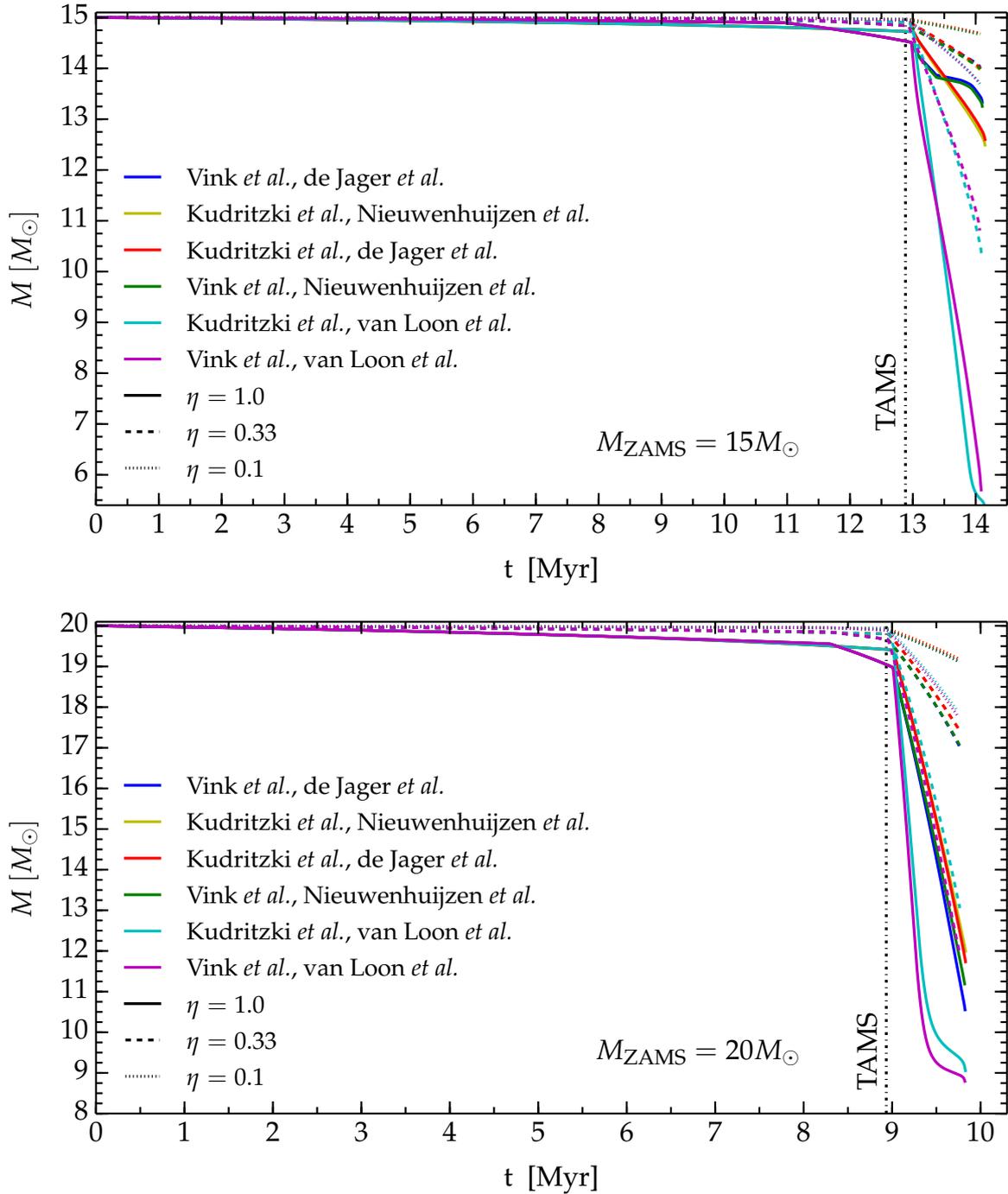


Figure 3.1: Total mass as a function of time. Each color corresponds to a given wind scheme. Simulations of stars that do not reach $X_s \leq 0.4$ during their evolution and differing only in the WR wind scheme are not shown. Dashed curves correspond to $\eta = 0.33$, and dotted curves correspond to $\eta = 0.1$, regardless of the color. The dot dashed vertical line indicates roughly the TAMS. The top panel is for $M_{\text{ZAMS}} = 15M_{\odot}$, and the bottom panel for $M_{\text{ZAMS}} = 20M_{\odot}$. Note the difference in vertical scale between the panels.

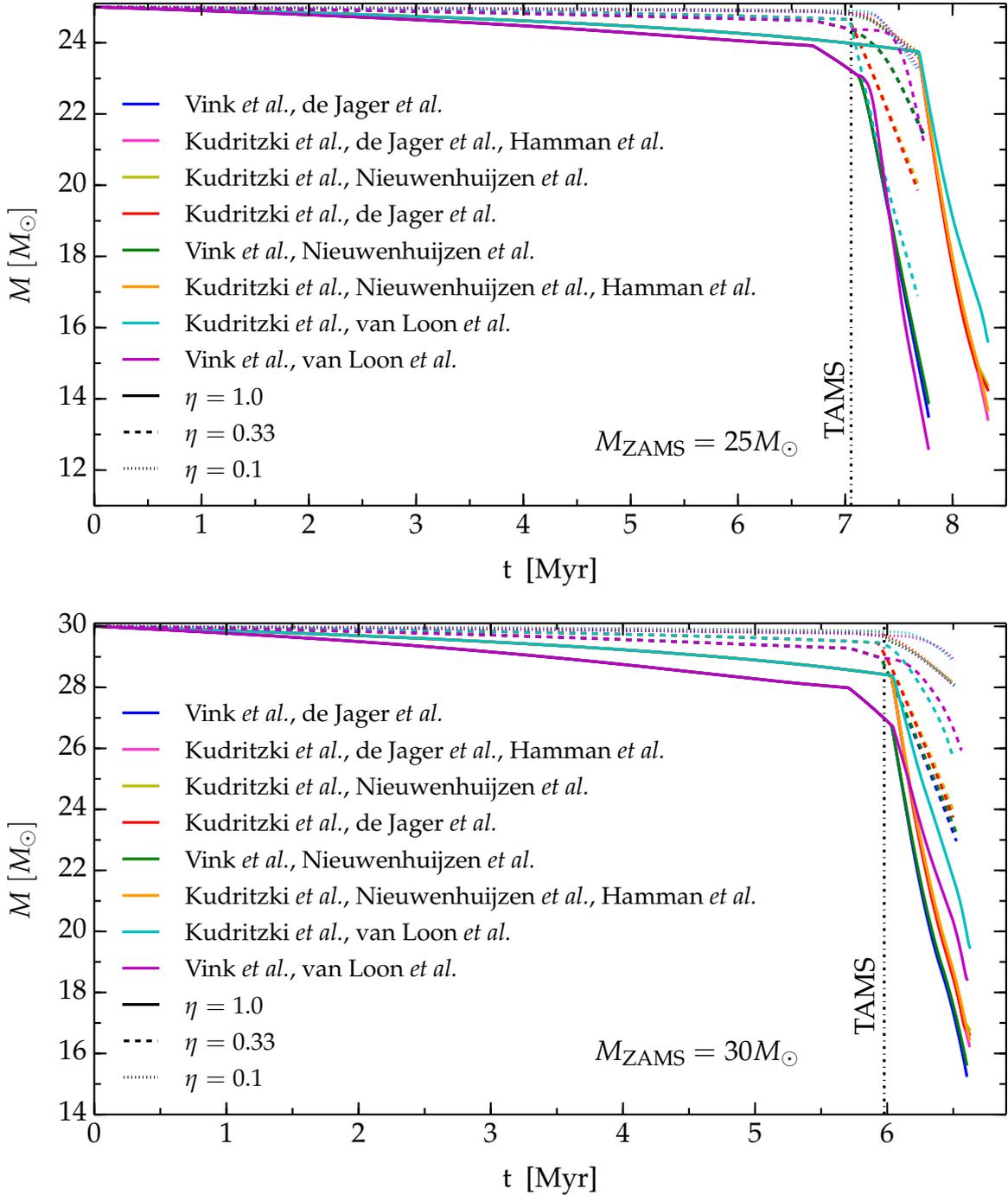


Figure 3.2: Total mass as a function of time. Each color corresponds to a given wind scheme. Simulations of stars that do not reach $X_s \leq 0.4$ during their evolution and differing only in the WR wind scheme are not shown. Models reaching the WR stage are those using the rates from Kudritzki *et al.* combined with de Jager *et al.* or Nieuwenhuijzen *et al.*, with $\eta = 1.0$. Dashed curves correspond to $\eta = 0.33$, and dotted curves correspond to $\eta = 0.1$, regardless of the color. The dot dashed vertical line indicates roughly the TAMS. The top panel is for $M_{\text{ZAMS}} = 25 M_{\odot}$, and the bottom panel for $M_{\text{ZAMS}} = 30 M_{\odot}$. Note the difference in vertical scale between the panels.

Figure 3.3 shows the relative final mass (at oxygen depletion, $X_{16\text{O}} \leq 0.4$, [19]) as a function of the initial mass. It illustrates the large spread found varying the efficiency and/or the wind algorithm. Note that the effect of varying η are larger than varying the mass loss algorithm: points indicated by markers of the same shape (corresponding to the same η) cluster together. This theoretical uncertainty prevents the reconstruction of M_{ZAMS} from the (observed) final masses. Note that the spread is larger for $M_{\text{ZAMS}} = 15M_{\odot}$, at that it decreases increasing M_{ZAMS} . This is because all wind mass loss algorithm prescribe a high mass loss rate for more massive (and thus more luminous) stars, while for lower mass stars only algorithms using the van Loon *et al.* rate (for the cool phase of evolution) yield a significantly lower final mass. Note also that the decrease of the spread increasing M_{ZAMS} does not tend to converge toward a single final value for $M_{\text{O depl}}/M_{\text{ZAMS}}$, but rather it tends to a large band. This could be an indication of an irreducible theoretical uncertainty. It is possible that each single mass loss algorithm captures (in a simplified way) some, but not all, the crucial aspects of the radiatively driven mass loss processes. Therefore, it is possible to get an idea of the uncertainty deriving from the modeling assumptions by running models with the same initial conditions (M_{ZAMS} , Z , rotation, etc.) with all the available algorithms. Similar plots for other evolutionary stages (e.g. at carbon ignition instead of at oxygen depletion) are not shown since they do not show different features.

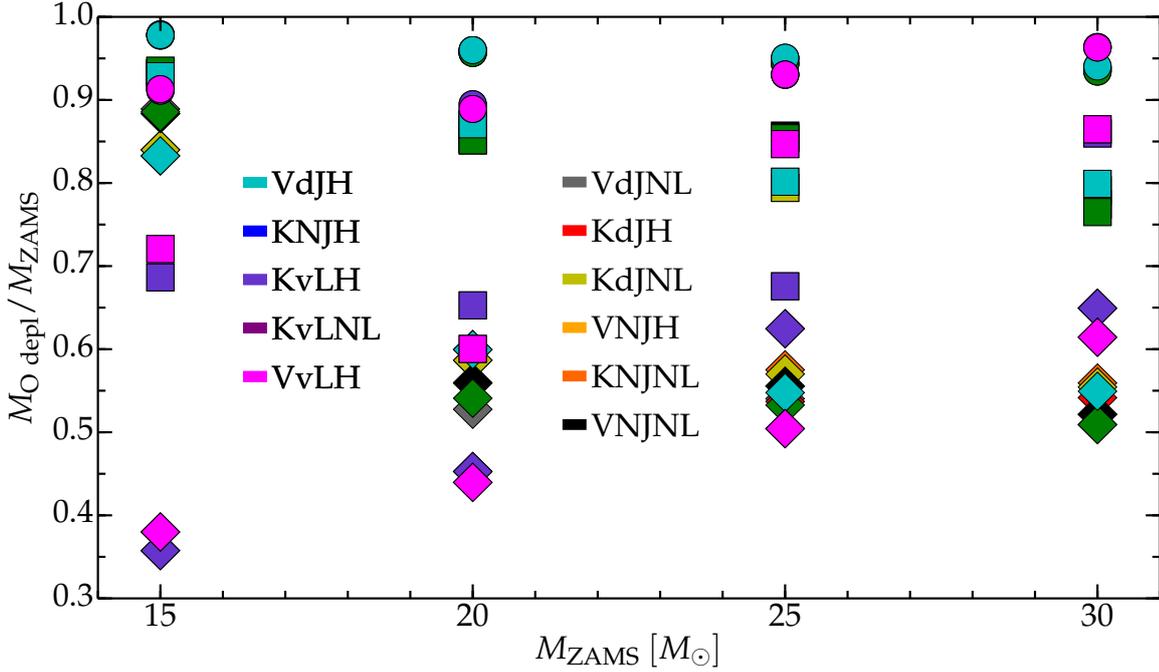


Figure 3.3: Relative final mass as a function of M_{ZAMS} . Diamonds, squares and circles correspond to $\eta = 1.0$, $\eta = 0.33$, and $\eta = 0.1$, respectively. Note that many points overlap.

Table 3.2 summarizes the properties of the computed models at oxygen depletion. The compactness parameter $\zeta_{\mathcal{M}}^{\text{O depl}}$ [89] is defined by

$$\zeta_{\mathcal{M}}^{\text{O depl}} \equiv \zeta_{\mathcal{M}}(t_{\text{O depl}}), \quad \zeta_{\mathcal{M}}(t) \stackrel{\text{def}}{=} \frac{\mathcal{M}/M_{\odot}}{R(\mathcal{M})/1000\text{km}}, \quad (3.1)$$

where I set $\mathcal{M} = 2.5M_{\odot}$. This is roughly the mass threshold separating the most massive

NSs from the least massive stellar BHs [16, 19, 89]. Note that $\xi_{\mathcal{M}}$ is necessarily a function of time, as the stellar core contracts during the evolution, increasing its temperature and density, and to sustain the star through the nuclear processing of heavier nuclei (see also Fig. 3.8).

To avoid introducing more arbitrary thresholds, I define the edge of the helium and carbon-oxygen cores as follows:

$$\text{H core} \stackrel{\text{def}}{=} \max \left(\frac{dX}{dm} \right) , \quad (3.2)$$

$$\text{CO core} \stackrel{\text{def}}{=} \max \left[\max \left(\frac{dY}{dm} \right) , \left(\frac{dX_{12\text{C}}}{dm} \right) \right] . \quad (3.3)$$

Equation 3.2 sets the helium core boundary to the location (going inward) where the hydrogen abundance as a function of the mass coordinate $X(m)$ is the steepest, and can be generalized for any other core. The reason for defining a carbon-oxygen core, instead of two separated carbon-rich and oxygen-rich cores, is that the separation between the two is not always clear, and the abundances of carbon and oxygen at the boundaries depends on the details of the nuclear burning process. This also explains the definitions in Eq. 3.2 and Eq. 3.3 : it is difficult to find a threshold for the abundance of some isotopes that can define clear boundaries of the cores for any M_{ZAMS} . I find a good agreement between the core masses obtained with Eq. 3.2 and Eq. 3.3, and the usual boundary definitions (e.g. innermost location where $X \leq 0.01$ and $Y \leq 0.01$ for the He core and CO core, respectively), unless the latter yield a zero mass core, because the threshold value is not reached anywhere. Note that the only carbon isotope included in the nuclear reaction network (approx21.net) is ^{12}C , since its purpose is not to follow the details of the nucleosynthesis, but just to provide ϵ_{nuc} (although, see the discussion in §B.3.2).

The rows of Tab. 3.2 give triplets of values for some stellar parameters for each wind algorithm varying the efficiency η , while the columns show the same stellar parameters for a given efficiency η varying the mass loss algorithm. The wind schemes are labeled according to Tab. 2.1, and their scalings with the physical quantities determining the mass loss rate are reported in Tab. 2.2.

As Fig. 3.1, Fig. 3.2, Fig. 3.3 and Tab. 3.2 show, η has a larger effect on the resulting stellar structure than the choice of mass loss algorithm. However, the structure at oxygen depletion is *not* uniquely determined by the parameters shown. Some evolutionary behaviors appear depending on the wind scheme (see also Fig. 3.4–3.7).

Although, in principle, the wind schemes compared are just different algorithmic representations of the same physical phenomenon, they are *not* equivalent to each other, neither from the theoretical point of view (because of the different choices of physical parameters to express \dot{M} , see §1.4 and Tab. 2.2), nor from the modelling point of view (because they yield significantly different evolutionary tracks and final structures).

It is hard to find clear trends in Tab. 3.2 – except maybe the extremely large mass loss rate resulting from the Van Loon *et al.* algorithm (§1.4.4, wind schemes containing “vL” in the name), compared to the other cool phase algorithms (i.e. de Jager *et al.* and Nieuwenhuijzen *et al.*) for $M_{\text{ZAMS}} \leq 25M_{\odot}$, see §3.3.

Not all the 144 (=4 M_{ZAMS} × 2 hot wind schemes × 3 cool wind schemes × 2 WR wind schemes × 3 values of η) models in this grid are truly different. Models differing only in the WR wind scheme, and that do not reach the condition to switch to it, i.e. $X_s < 0.4$,

are effectively identical (e.g. s15VdJNL.1.0 and s15VdJH.1.0). This redundancy reduces the number of non-identical models in the grid to 76. Of these, 30 end their evolution¹ as RSG with $\log_{10}(T_{\text{eff}}/[\text{K}]) \leq 3.6$; 38 end as YSG with $\log_{10}(T_{\text{eff}}/[\text{K}]) \leq 3.8$; eight end as WR with $X_s < 0.4$, and none reach oxygen depletion as BSG (i.e. with $\log_{10}(T_{\text{eff}}/[\text{K}]) > 3.8$), see Tab. 3.1. The temperature threshold to define BSG, RSG and YSG are somewhat arbitrary, and come from [47]. Since most of the RSG in this grid are “hot-RSG”, see Fig. 3.4–Fig. 3.7, the RSG-to-YSG ratio is very sensitive to the value chosen for the corresponding threshold. For example, assuming $\log_{10}(T_{\text{eff}}/[\text{K}]) \leq 3.68$ to separate RSG from YSG, the number of RSG increases to 54, and only 14 YSG are left.

Table 3.1: Final state of all the computed models. Red Super-Giants (RSG) correspond to $\log_{10}(T_{\text{eff}}/[\text{K}]) \leq 3.6$, Yellow Super-Giants (YSG) correspond to $3.6 \leq \log_{10}(T_{\text{eff}}/[\text{K}]) \leq 3.8$ [47], no Blue-Supergiant (BSG) are found in this grid. WR stars have $X_s \leq 0.4$, regardless of their surface temperature. The first column indicates the mass loss algorithm combination used, see Tab. 2.1 and Tab. 2.2 for details. Note the redundancy between lines differing only in the WR wind scheme.

$M_{\text{ZAMS}} [M_{\odot}]$		15			20			25			30		
Wind Scheme	$\eta =$	0.1	0.33	1.0	0.1	0.33	1.0	0.1	0.33	1.0	0.1	0.33	1.0
VdJNL		RSG	RSG	RSG	RSG	RSG	YSG	YSG	YSG	YSG	YSG	YSG	YSG
VdJH		RSG	RSG	RSG	RSG	RSG	YSG	YSG	YSG	YSG	YSG	YSG	YSG
VNJNL		RSG	RSG	RSG	RSG	RSG	YSG	YSG	YSG	YSG	YSG	YSG	YSG
VNJH		RSG	RSG	RSG	RSG	RSG	YSG	YSG	YSG	YSG	YSG	YSG	YSG
VvLNL		RSG	RSG	YSG	RSG	RSG	YSG	YSG	YSG	YSG	YSG	YSG	YSG
VvLH		RSG	RSG	YSG	RSG	RSG	YSG	YSG	YSG	YSG	YSG	YSG	YSG
KdJNL		RSG	RSG	RSG	RSG	RSG	RSG	YSG	YSG	WR	YSG	YSG	WR
KdJH		RSG	RSG	RSG	RSG	RSG	RSG	YSG	YSG	WR	YSG	YSG	WR
KNJH		RSG	RSG	RSG	RSG	RSG	RSG	YSG	YSG	WR	YSG	YSG	WR
KNJNL		RSG	RSG	RSG	RSG	RSG	RSG	YSG	YSG	WR	YSG	YSG	WR
KvLNL		RSG	RSG	YSG	RSG	RSG	YSG	YSG	YSG	YSG	YSG	YSG	YSG
KvLH		RSG	RSG	YSG	RSG	RSG	YSG	YSG	YSG	YSG	YSG	YSG	YSG

The lack of BSG is not completely surprising, considering the maximum $M_{\text{ZAMS}} = 30M_{\odot}$ used is not very high, and standard stellar evolution theory predicts for stars with $15M_{\odot} \lesssim M_{\text{ZAMS}} \lesssim 30M_{\odot}$ a RSG pre-SN stage (see also §1.2.3 and, e.g., [29]).

The only simulations reaching the threshold ($X_s \leq 0.4$, see §2.2.2) for switching to the WR wind scheme are for $M_{\text{ZAMS}} = \{25, 30\} M_{\odot}$ using the Kudritzki *et al.* rate (during the hot phase) and de Jager *et al.* or Nieuwenhuijzen *et al.* rate (during the cool phase), with $\eta = 1.0$.

The small number of WR stars produced in this grid has several (degenerate) explanations:

- The maximum M_{ZAMS} used (see §2.3) is only $30M_{\odot}$, and the progenitors of WR stars may be more massive on the ZAMS. Both observations and numerical simulations carried out with standard assumptions infer $M_{\text{ZAMS}} \gtrsim 25\text{--}30 M_{\odot}$ as the typical lower boundary for the initial mass of a single star capable of losing its entire hydrogen envelope and becoming a WR, see e.g. [5, 30, 90, 91].
- The choices of the description of convection (i.e. using the Ledoux criterion, the parameters chosen for the mixing length and overshooting, the description of semiconvec-

¹The simulations are stopped at oxygen depletion $X_c(^{16}\text{O}) \leq 0.4$, however, the subsequent evolution is so fast that it cannot change significantly the surface properties of the stars.

tion and thermohaline mixing), that enter the determination of the core dimensions, the luminosity and thus the mass loss rate. They also determine the mixing of heavier elements into the outer layers, which is necessary to reduce the surface abundance of hydrogen below the adopted threshold. Note that the influence of these parameters on the production of WR stars could be tested with efficient stellar evolution codes such as MESA. Rapid rotation, neglected in this work, may play a role both in the mixing and mass loss processes, see e.g. [92, 93] and references therein.

- The treatment of nearly super-Eddington radiation dominated envelopes (see §1.3.1, §2.1.3 and [39]). Simulations carried out with other codes (e.g. FRANEC, Limongi, private communication) remove all the mass outside the location where $L = L_{\text{Edd}}$ or (e.g. the Geneva code, [94]) impose an arbitrary, extreme and constant mass loss rate $\dot{M} \sim 10^{-4} - 10^{-3} M_{\odot} \text{ yr}^{-1}$ in the region of the HR diagram corresponding to LBVs, thus more easily producing WR stars. These, however, cannot be realistic approaches, since the local “modified” Eddington luminosity can be exceeded in stellar interiors (cf. §1.3.1), and since the LBVs are (by definition) *variable* stars, and they experience both extreme mass outburst (at constant bolometric L , [95]) and quiescent phases.
- It is possible that binarity has the dominant role in stripping the hydrogen envelope (e.g. [96]) and producing WR stars. In this scenario, the lack of WR stars in the grid presented here would be expected.

The differences between the two WR mass loss algorithms compared in this work (Hamann *et al.* – §1.4.8, and Nugis & Lamers – §1.4.7) are discussed in §3.4.

Figures 3.4–3.7 show the evolutionary tracks on the HR diagrams of all computed models, allowing a quick eyeball comparison of the results.

CHAPTER 3. RESULTS: WIND ALGORITHMS COMPARISON

Table 3.2: Radius ($R_{\text{O depl}}$), total mass ($M_{\text{O depl}}$), He core mass (M_{He}) and CO core mass (M_{CO}) in solar units, and compactness parameter at oxygen depletion for each M_{ZAMS} and efficiency η used. Models that do not reach the threshold to switch to the WR wind algorithm ($X_s \leq 0.4$) during their evolution and differing only in the WR wind scheme are not listed. The values referring to WR stars are highlighted and in red bold font. The first column indicates the mass loss algorithm combination used, see Tab. 2.1 and Tab. 2.2 for details.

$M_{\text{ZAMS}} = 15 M_{\odot}$																
Wind Scheme	$R [R_{\odot}]$			$M_{\text{O depl}} [M_{\odot}]$			$M_{\text{He}} [M_{\odot}]$			$M_{\text{CO}} [M_{\odot}]$			$\zeta_{2.5}^{\text{O depl}}$			
$\eta =$	0.1	0.33	1.0	0.1	0.33	1.0	0.1	0.33	1.0	0.1	0.33	1.0	0.1	0.33	1.0	
VdJNL	924	928	946	14.68	14.02	13.34	5.15	5.07	5.01	3.32	3.26	3.25	0.146	0.156	0.147	
VNJNL	933	933	951	14.66	13.98	13.25	5.15	5.07	5.01	3.32	3.26	3.21	0.130	0.137	0.165	
VvLNL	947	986	673	13.69	10.81	5.70	5.15	5.06	4.84	3.29	3.21	3.04	0.150	0.168	0.115	
KdJNL	933	944	940	14.68	13.97	12.60	5.17	5.16	4.95	3.29	3.31	3.12	0.164	0.123	0.168	
KNJNL	928	939	942	14.67	13.93	12.49	5.17	5.16	4.95	3.29	3.29	3.15	0.126	0.156	0.166	
KvLNL	951	973	561	13.67	10.30	5.36	5.17	5.15	4.95	3.32	3.25	3.13	0.153	0.150	0.137	
$M_{\text{ZAMS}} = 20 M_{\odot}$																
Wind Scheme	$R [R_{\odot}]$			$M_{\text{O depl}} [M_{\odot}]$			$M_{\text{He}} [M_{\odot}]$			$M_{\text{CO}} [M_{\odot}]$			$\zeta_{2.5}^{\text{O depl}}$			
$\eta =$	0.1	0.33	1.0	0.1	0.33	1.0	0.1	0.33	1.0	0.1	0.33	1.0	0.1	0.33	1.0	
VdJNL	1044	1041	863	19.14	17.02	10.55	7.45	7.53	7.69	4.97	4.95	5.02	0.197	0.196	0.186	
VNJNL	1042	1045	910	19.13	17.07	11.19	7.45	7.53	7.19	4.93	4.99	4.92	0.200	0.195	0.195	
VvLNL	1040	950	581	17.78	12.01	8.79	7.45	7.52	7.62	4.96	4.90	4.93	0.202	0.188	0.193	
KdJNL	1039	1029	947	19.20	17.39	11.73	7.40	7.33	7.41	4.95	4.84	4.86	0.149	0.162	0.207	
KNJNL	1040	1030	949	19.20	17.41	11.99	7.40	7.32	7.41	4.95	4.87	4.87	0.200	0.167	0.188	
KvLNL	1040	987	672	17.90	13.05	9.05	7.39	7.33	7.38	4.94	4.86	4.77	0.190	0.180	0.171	
$M_{\text{ZAMS}} = 25 M_{\odot}$																
Wind Scheme	$R [R_{\odot}]$			$M_{\text{O depl}} [M_{\odot}]$			$M_{\text{He}} [M_{\odot}]$			$M_{\text{CO}} [M_{\odot}]$			$\zeta_{2.5}^{\text{O depl}}$			
$\eta =$	0.1	0.33	1.0	0.1	0.33	1.0	0.1	0.33	1.0	0.1	0.33	1.0	0.1	0.33	1.0	
VdJNL	969	989	820	23.57	21.36	13.51	9.79	9.19	9.24	6.48	6.20	6.17	0.221	0.191	0.198	
VNJNL	972	983	831	23.61	21.42	13.89	9.47	9.19	9.23	6.70	6.13	6.21	0.221	0.163	0.141	
VvLNL	971	990	760	23.27	21.17	12.61	9.79	9.19	9.24	6.56	6.12	6.29	0.232	0.193	0.095	
KdJNL	930	935	689	23.75	19.85	14.25	9.59	9.88	10.32	6.62	6.94	6.99	0.174	0.110	0.220	
KdJH	930	935	632	23.75	19.85	13.43	9.59	9.88	10.19	6.62	6.94	7.19	0.174	0.110	0.236	
KNJH	929	944	652	23.77	20.04	13.69	9.59	9.88	10.18	6.52	6.69	7.13	0.193	0.211	0.202	
KNJNL	929	944	695	23.77	20.04	14.38	9.59	9.88	10.22	6.52	6.69	7.24	0.193	0.211	0.225	
KvLNL	946	906	784	23.26	16.89	15.62	9.59	9.90	10.70	6.62	6.82	6.99	0.203	0.170	0.222	
$M_{\text{ZAMS}} = 30 M_{\odot}$																
Wind Scheme	$R [R_{\odot}]$			$M_{\text{O depl}} [M_{\odot}]$			$M_{\text{He}} [M_{\odot}]$			$M_{\text{CO}} [M_{\odot}]$			$\zeta_{2.5}^{\text{O depl}}$			
$\eta =$	0.1	0.33	1.0	0.1	0.33	1.0	0.1	0.33	1.0	0.1	0.33	1.0	0.1	0.33	1.0	
VdJNL	897	775	586	28.02	22.96	15.28	11.02	11.86	11.45	7.27	8.15	8.09	0.215	0.216	0.201	
VNJNL	897	768	605	28.07	23.22	15.65	11.02	11.84	11.55	7.12	8.40	8.17	0.221	0.221	0.187	
VvLNL	901	807	676	28.89	25.94	18.43	10.99	11.03	11.81	7.38	7.41	8.20	0.219	0.217	0.216	
KdJNL	880	770	693	28.15	23.73	16.63	11.40	11.89	11.39	7.83	7.94	7.98	0.216	0.211	0.215	
KdJH	880	770	680	28.15	23.73	16.25	11.40	11.89	11.39	7.83	7.94	8.10	0.216	0.211	0.188	
KNJH	857	779	691	28.21	23.97	16.48	11.37	11.89	11.42	7.68	8.07	8.01	0.226	0.209	0.228	
KNJNL	857	779	700	28.21	23.97	16.78	11.37	11.89	11.39	7.68	8.07	8.18	0.226	0.209	0.210	
KvLNL	882	776	743	28.91	25.79	19.48	11.39	11.84	11.73	7.82	8.21	7.70	0.206	0.214	0.236	

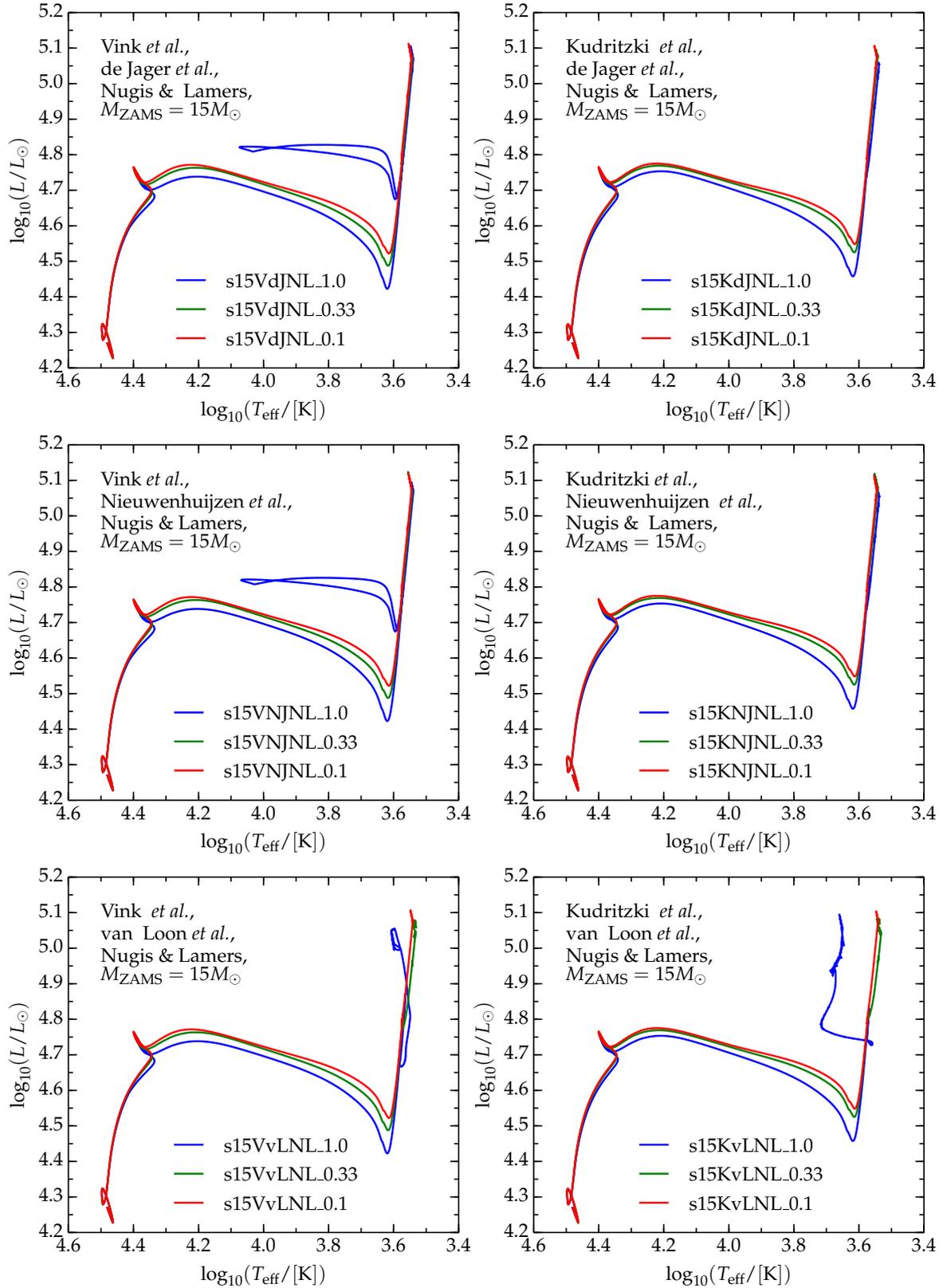


Figure 3.4: HR diagrams for the $M_{\text{ZAMS}} = 15M_{\odot}$ models. Each panel corresponds to a specific wind scheme, as indicated. Blue, green and red curves correspond to $\eta = 1.0$, $\eta = 0.33$, and $\eta = 0.1$, respectively.

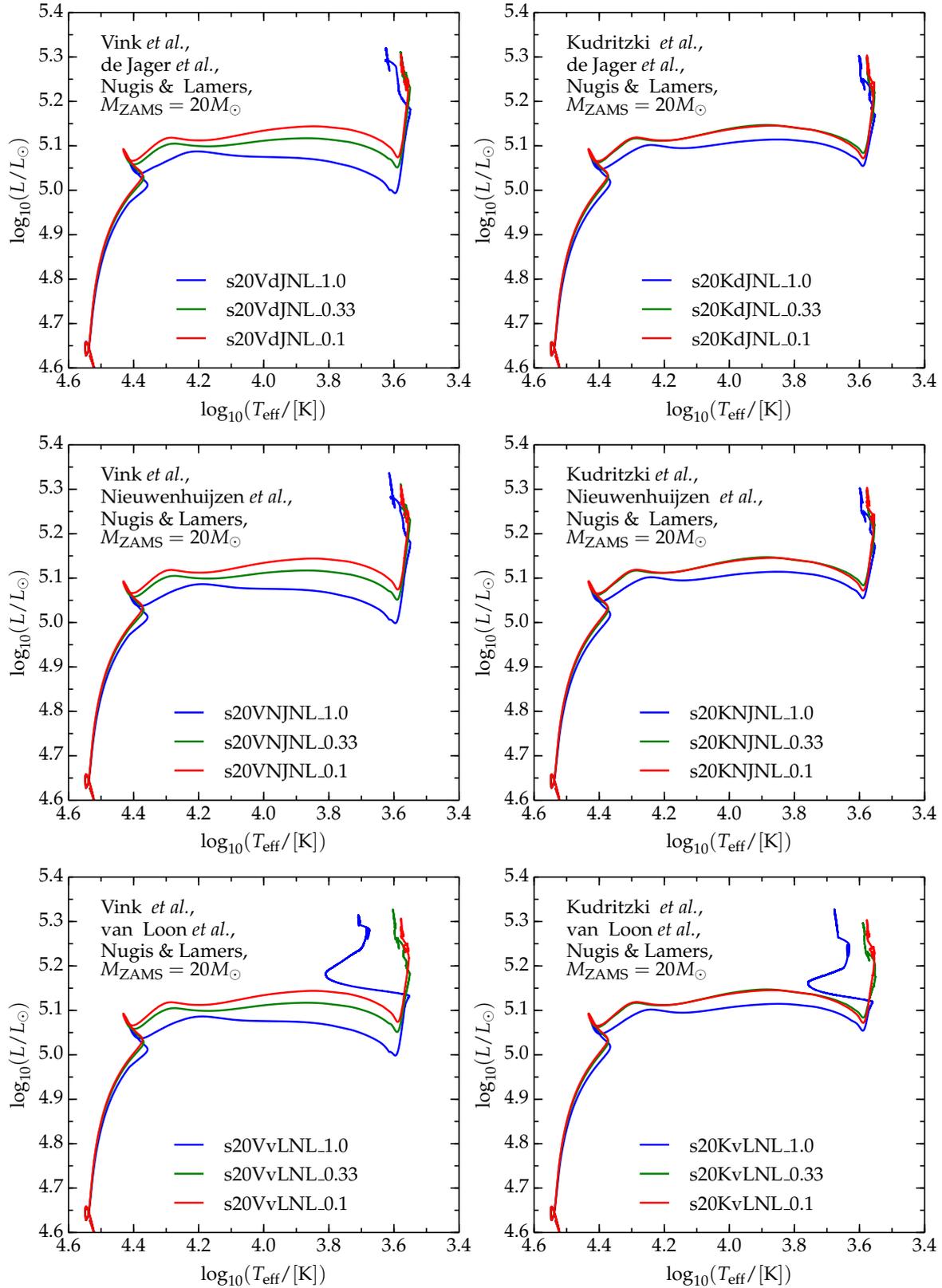


Figure 3.5: HR diagrams for the $M_{\text{ZAMS}} = 20M_{\odot}$ models. Each panel corresponds to a specific wind scheme, as indicated. Blue, green and red curves correspond to $\eta = 1.0$, $\eta = 0.33$, and $\eta = 0.1$, respectively.

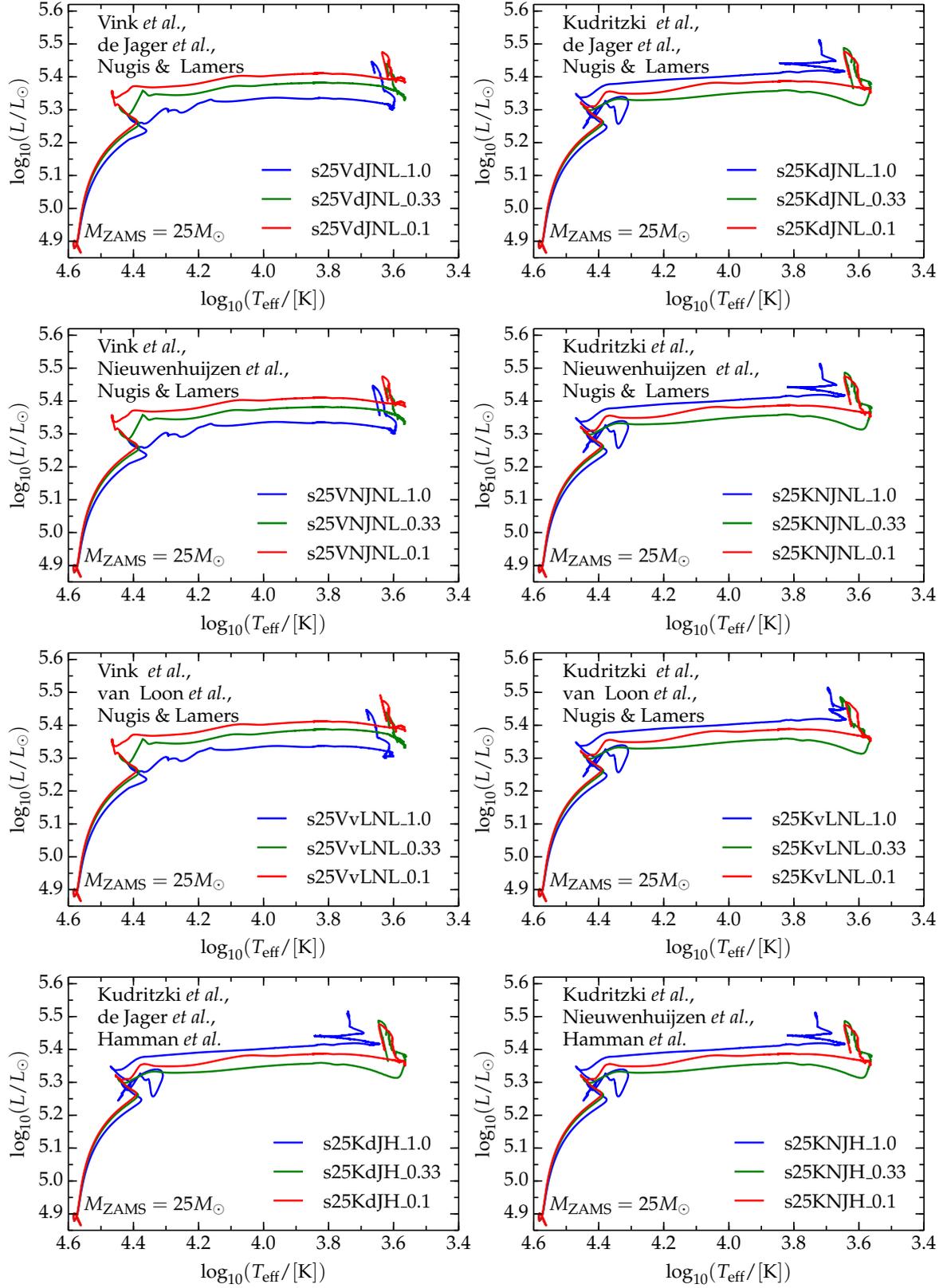


Figure 3.6: HR diagrams for the $M_{\text{ZAMS}} = 25M_{\odot}$ models. Each panel corresponds to a specific wind scheme, as indicated. Blue, green and red correspond to $\eta = 1.0$, $\eta = 0.33$, and $\eta = 0.1$, respectively. The blue curves in the panels on the right of the first and second row, and in the last row reach oxygen depletion as WR stars.

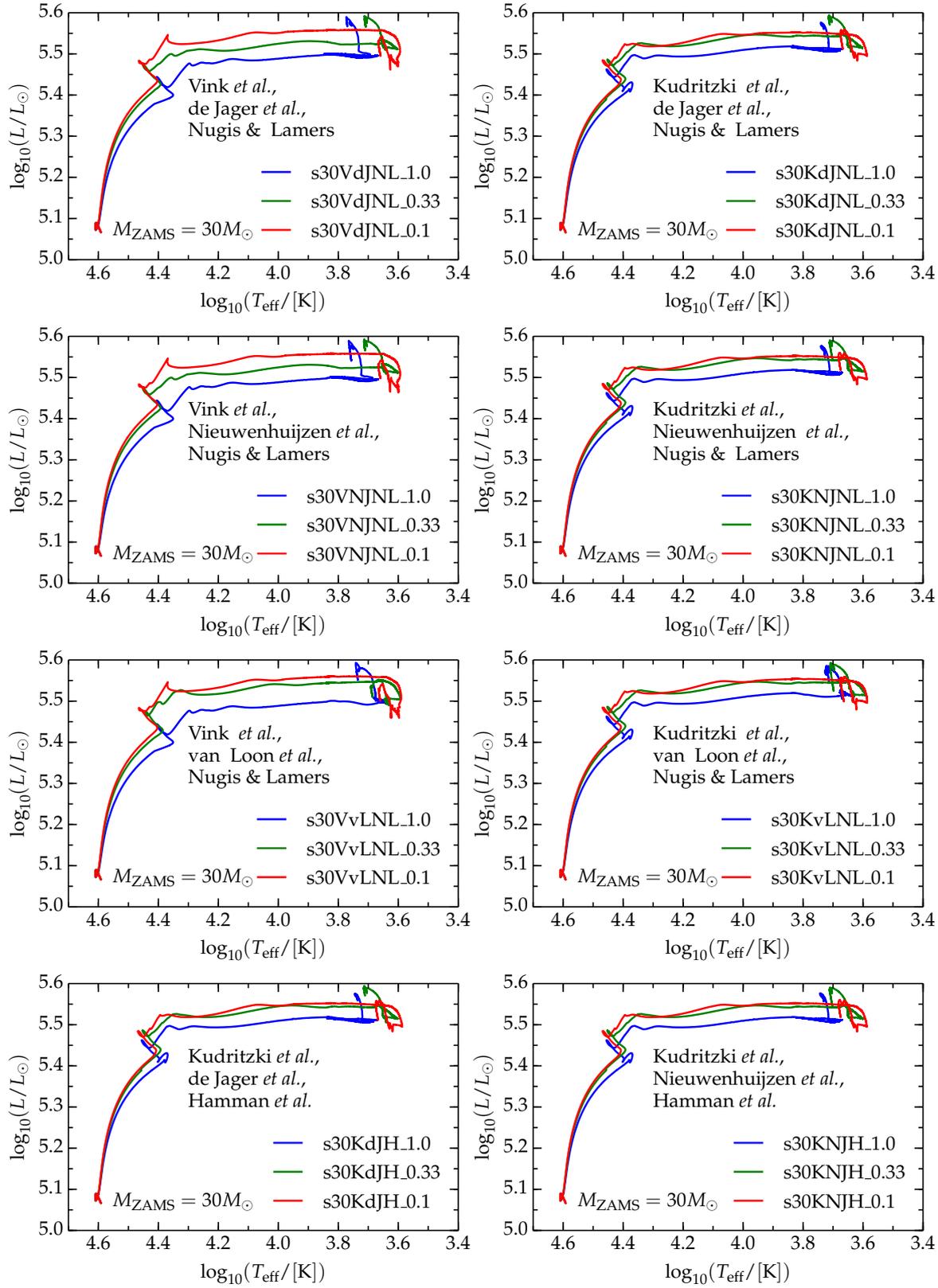


Figure 3.7: HR diagrams for the $M_{\text{ZAMS}} = 30M_{\odot}$ models. Each panel corresponds to a specific wind scheme, as indicated. Blue, green and red correspond to $\eta = 1.0$, $\eta = 0.33$, and $\eta = 0.1$, respectively. The blue curves in the panels on the right of the first and second row, and in the last row reach oxygen depletion as WR stars.

3.1.1 The Compactness at Oxygen Depletion

The last three columns of Tab. 3.2 show the compactness parameter at oxygen depletion for each model. One of the aims of this work is to better understand the influence of mass loss on this parameter and, more generally, on the pre-SN core structure. However, because of difficulties encountered in obtaining spatially converged structures² during the advanced burning stages (see also §2.3 and §B.3.3) – when mass loss has already been shut down in MESA – it is hard to draw any conclusions from the results presented here.

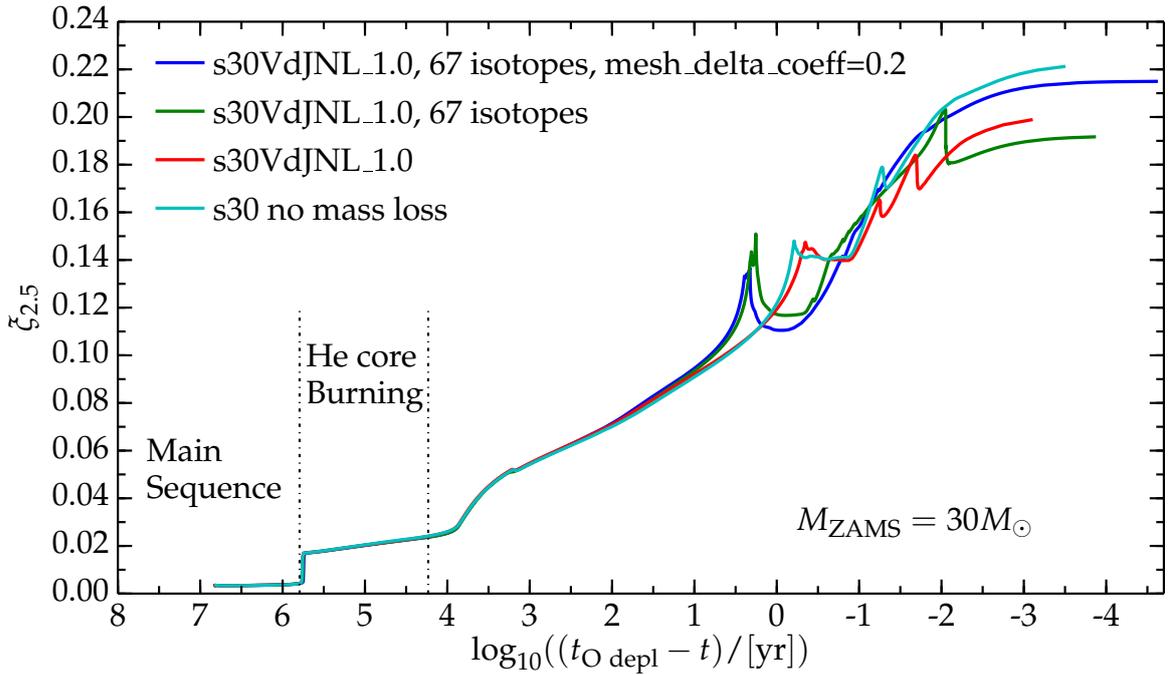


Figure 3.8: Evolution of the compactness parameter $\zeta_{2.5}$ for $M_{\text{ZAMS}} = 30M_{\odot}$ simulations with different parameters. The cyan curve is computed without mass loss; all the others use the Vink *et al.*, de Jager *et al.*, and Nugis & Lamers combination with $\eta = 1.0$ and they differ in the nuclear reaction network (approx21.net for the red curve, mesa.67.net for the blue and the green curves), and in the spatial resolution (mesh_delta_coeff), which is (approximately) five times higher for the blue curve. The dot dashed lines mark the TAMS and the end of helium core burning. The differences between the tracks are an effect of the different spatial discretization (see text).

The four computations in Fig. 3.8 are for a $M_{\text{ZAMS}} = 30M_{\odot}$ star, computed with the Vink *et al.*, de Jager *et al.* and Nugis & Lamers (VdJNL) mass loss rate combination and $\eta = 1.0$. The blue curve shows a run computed with a 67-isotope nuclear reaction network (mesa.67.net) at “high spatial resolution” (mesh_delta_coeff=0.2, corresponding to 34852 computational cells at oxygen depletion), i.e. roughly five times the standard resolution (mesh_delta_coeff=1.0, corresponding to 7829 computational cells at oxygen depletion). All other curves shown were computed with standard resolution (mesh_delta_coeff=1.0). The green curve is computed using the same 67-isotope nuclear reaction network (mesa.67.net),

²By spatially converged structure, I mean stellar structures that do not vary significantly increasing the spatial resolution (i.e. the number of computational cells) of the mesh adopted for the numerical simulations.

the red and the cyan curve use the reduced network adopted for most of this work, i.e. the 21-isotope `approx21.net`. The difference between these last two curves is that mass loss is completely shut down for the cyan curve (i.e. $\eta = 0$).

The comparison between the red and cyan tracks in Fig. 3.8 shows the effect of mass loss on the compactness parameter. The effect is smaller or comparable to changing the nuclear reaction network (compare the red and green curves), the spatial resolution (compare the green and the blue curves), or both (compare the red and the blue curves). The various numerical experiments performed³ indicate that a variation of some of the parameters causes changes in the spatial grid provided by MESA, which in turn causes variations in the compactness during (late-)carbon burning, neon and oxygen burning. Therefore, even if mass loss influences the compactness, its effect (at least until oxygen depletion) is too small to be disentangled from the effects of the spatial resolution of the numerical simulation in the late phases.

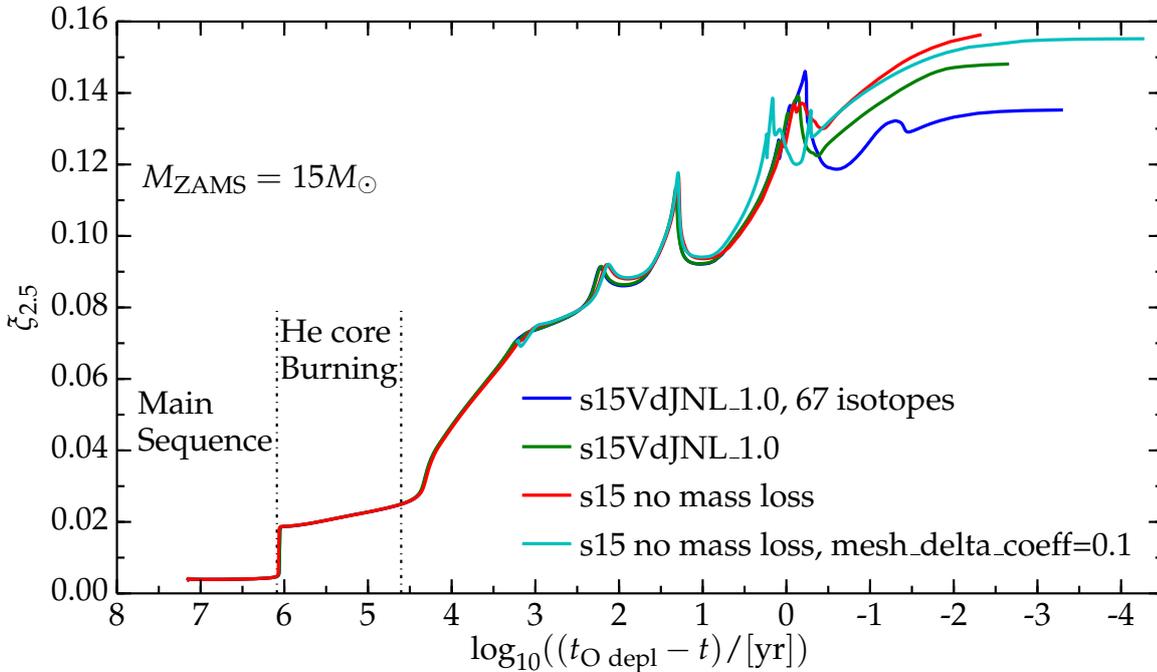


Figure 3.9: Evolution of the compactness parameter for $M_{\text{ZAMS}} = 15M_{\odot}$, solar metallicity models. The blue, red, and green curves are computed with standard resolution (`mesh_delta_coeff=1.0`). The blue curve use the 67-isotope nuclear reaction network (`mesa_67.net`), the other curves use the 21-isotope network used throughout this entire work (`approx21.net`). The green and the blue curves use the Vink *et al.*, de Jager *et al.* mass loss rates, the red and the cyan are computed without mass loss ($\eta = 0$). The cyan curve is not computed from ZAMS, but it is obtained restarting the red curve with a higher spatial resolution from $\log_{10}((t_{\text{O depl}} - t)/[\text{yr}]) \sim 3$.

Figure 3.8 shows an example of the evolution of the compactness parameter as a function of time. Note that the overall qualitative behavior of $\zeta_{2.5}$ as a function of time does not

³I also ran models turning off the wind mass loss in each of the evolutionary phases, but the spread in the final results is always smaller or comparable to the spread obtained when changing the resolution.

change for different M_{ZAMS} , see also Fig. 3.9 – which is another example of the difficulties in obtaining a converged core structure for another M_{ZAMS} . The compactness parameter $\zeta_{2.5}$ stays almost constant during the main sequence, it increases during the hydrogen shell burning phase and then, during core helium burning, it continues increasing slightly. Finally, $\zeta_{2.5}$ starts increasing much more rapidly during core carbon burning. The details of the compactness behavior are much more uncertain in the last few years before oxygen depletion (i.e. for $\log_{10}((t_{\text{O depl}} - t)/[\text{yr}]) \lesssim 1.5$ for the $M_{\text{ZAMS}} = 30M_{\odot}$ models in Fig. 3.8), but an overall trend of increase is clear and robust (see also [19]). This uncertainty arises because of the difficulties encountered in obtaining a properly resolved and converged (i.e. without variations with increasing the number of computational cells) core structure in the spatial domain.

It is worth emphasizing that the post-oxygen depletion evolution of the star amplifies the compactness [19]. Therefore, even if the effect of mass loss on $\zeta_{2.5}$ until this stage is small, it may be the seed of significant differences in the compactness (and, more generally, in the structure) at the onset of core collapse. However, with the adopted resolution, and, more importantly, because of the lack of a convergence to the same behavior with increasing the resolution, it is impossible to identify clearly what the effect might be.

Using a larger nuclear reaction network (providing a more realistic energy generation rate, see §B.3) for neon and oxygen burning is certainly needed to improve these simulations. Moreover, a finer spatial grid, likely made *ad hoc* by selecting the regions that need higher resolution, would allow one to rule out the influence of the resolution on the results, although at an increased computational cost.

3.2 Hot Phase Mass Loss

By definition, the hot phase of the evolution occurs when $T_{\text{eff}} \geq T_{\text{eff}}^{\text{hot}} \stackrel{\text{def}}{=} 15000 \text{ K}$, i.e. $\log_{10}(T_{\text{eff}}/[\text{K}]) \geq 4.18$, and with $X_s > 0.4$ (see §2.2.2). This is the longest phase in the evolution of a star, since it corresponds roughly to the main sequence. The main sequence is physically defined by the core nuclear burning, while here it is more meaningful to define the different phases based on the surface of a star, from which mass is lost (both in nature and in these numerical simulations). Because of the different definitions, the “hot evolutionary phase” is slightly longer than the main sequence, including the overall contraction phase and ending at the start of the Sub-Giant Branch.

During the hot phase, two wind mass loss algorithms are commonly adopted in the simulations of massive star evolution: Vink *et al.* (§1.4.5) and Kudritzki *et al.* (§1.4.6). Tab. 3.3 summarizes some physical properties of the stellar models when T_{eff} drops below $T_{\text{eff}}^{\text{hot}}$ (i.e. when the mass loss rate switches to the cool phase algorithm) for each M_{ZAMS} and η factor.

Figure 3.10 shows the mass loss rate as a function of time and luminosity for $\eta = 1.0$ and each M_{ZAMS} . The tracks stop when $T_{\text{eff}} \equiv T_{\text{eff}}^{\text{hot}}$. Analogous curves with different values of the efficiency η are not shown since they are qualitatively similar, apart from a downward shift on the ordinate. Note that this is an important result: a modification of the mass loss rate \dot{M} with all the other parameters held constant might, in principle, significantly change the surface properties of the star and cause a divergence of the tracks on the (t, \dot{M}) and (L, \dot{M}) planes, instead of a mere shift.

The differences in the mass loss rate have a relevant effect both on the total mass and the helium core mass at the end of the hot phase of the evolution (see §3.2.2), while the

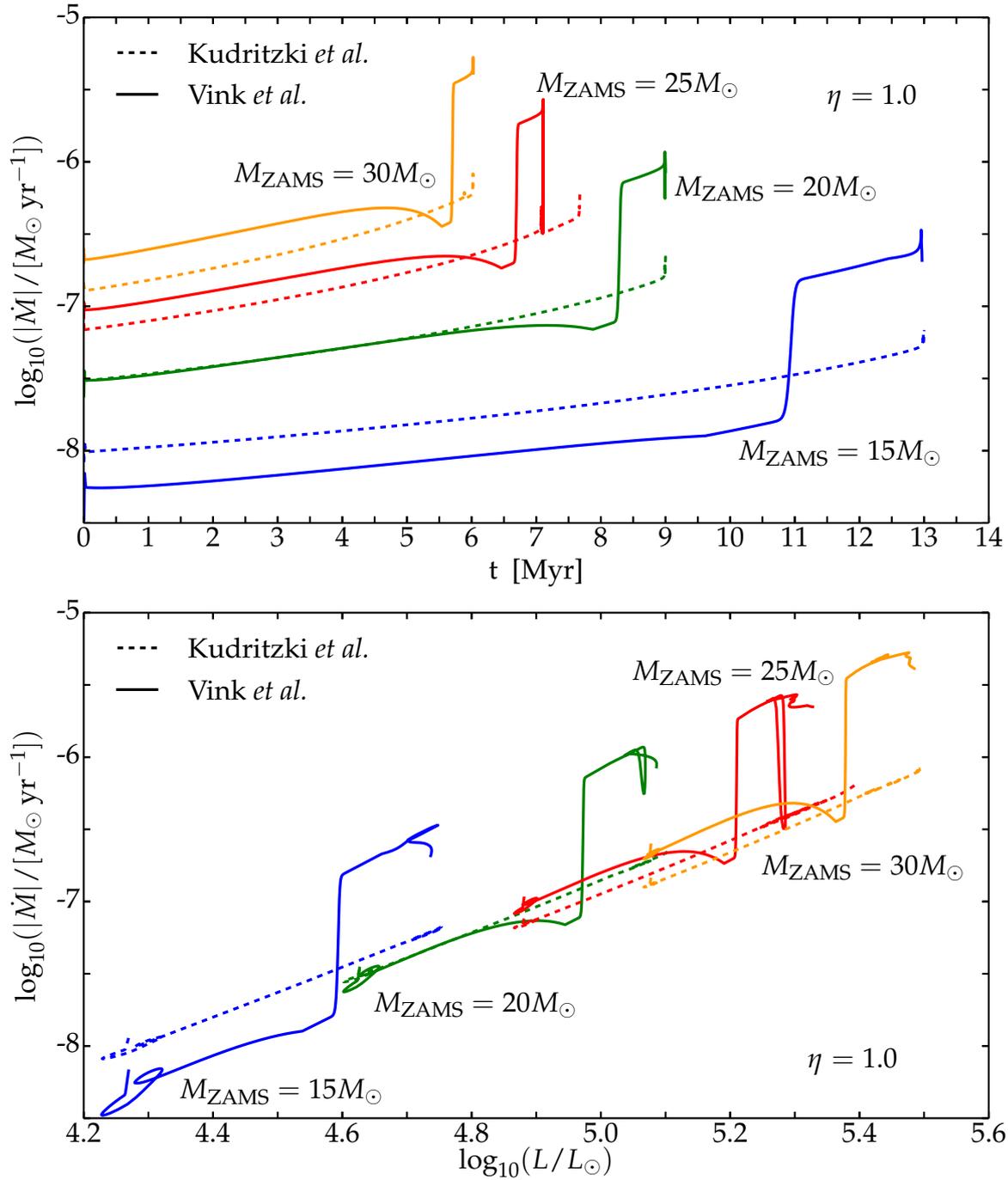


Figure 3.10: Hot phase ($T_{\text{eff}} \geq 15000$ K) mass loss rate as a function of time (top panel) and luminosity (bottom panel) for $\eta = 1.0$. The qualitative behavior found with lower values of η is similar except for a downward shift on the y-axis. Dashed curves use the Kudritzki *et al.* algorithm (§1.4.6), while solid curves use the Vink *et al.* algorithm (§1.4.5). Each color corresponds to a specific value of M_{ZAMS} .

compactness of these models at $T_{\text{eff}} = T_{\text{eff}}^{\text{hot}}$ is almost unaffected, as Tab. 3.3 shows. However, differences in the total mass, helium core mass and structure (e.g. temperature gradient) may be the seed of significant variations in $\zeta_{2.5}$ during later stages (cf. Fig. 3.8, for example).

CHAPTER 3. RESULTS: WIND ALGORITHMS COMPARISON

 Table 3.3: Radius (R), luminosity (L), total mass (M_{tot}), helium core mass (M_{He}) in solar units, and compactness parameter ($\zeta_{2.5}^{T_{\text{eff}}=15[\text{kK}]}$) when $T_{\text{eff}} \equiv T_{\text{eff}}^{\text{hot}} = 15000$ K, i.e. when MESA switches from the hot wind scheme to the cool wind scheme.

Hot wind	M_{ZAMS} [M_{\odot}]	η	R [R_{\odot}]	L [$10^5 L_{\odot}$]	M_{tot} [M_{\odot}]	M_{He} [M_{\odot}]	$\zeta_{2.5}^{T_{\text{eff}}=15[\text{kK}]}$
Vink <i>et al.</i>	15	1.0	34.94	5.45	14.51	4.73	0.013
Kudritzki <i>et al.</i>	15	1.0	35.25	5.64	14.72	4.50	0.014
Vink <i>et al.</i>	15	0.33	35.70	5.77	14.84	4.47	0.014
Kudritzki <i>et al.</i>	15	0.33	36.08	5.83	14.91	4.39	0.014
Vink <i>et al.</i>	15	0.1	35.94	5.85	14.95	4.41	0.015
Kudritzki <i>et al.</i>	15	0.1	36.25	5.90	14.97	4.37	0.015
Vink <i>et al.</i>	20	1.0	52.41	12.19	18.98	6.30	0.015
Kudritzki <i>et al.</i>	20	1.0	53.06	12.47	19.41	6.23	0.017
Vink <i>et al.</i>	20	0.33	52.55	12.56	19.67	6.31	0.017
Kudritzki <i>et al.</i>	20	0.33	53.42	12.96	19.80	6.26	0.017
Vink <i>et al.</i>	20	0.1	53.80	12.96	19.90	6.30	0.017
Kudritzki <i>et al.</i>	20	0.1	53.96	12.97	19.94	6.29	0.017
Vink <i>et al.</i>	25	1.0	69.25	21.27	23.09	8.23	0.015
Kudritzki <i>et al.</i>	25	1.0	73.95	24.68	23.75	8.60	0.015
Vink <i>et al.</i>	25	0.33	72.10	22.72	24.38	7.98	0.016
Kudritzki <i>et al.</i>	25	0.33	70.12	21.50	24.65	8.54	0.016
Vink <i>et al.</i>	25	0.1	74.32	24.28	24.81	8.25	0.016
Kudritzki <i>et al.</i>	25	0.1	71.44	22.89	24.89	8.41	0.016
Vink <i>et al.</i>	30	1.0	82.14	30.54	26.77	10.38	0.014
Kudritzki <i>et al.</i>	30	1.0	83.76	31.20	28.39	9.96	0.015
Vink <i>et al.</i>	30	0.33	84.31	32.18	28.99	10.35	0.015
Kudritzki <i>et al.</i>	30	0.33	86.39	33.28	29.47	10.56	0.015
Vink <i>et al.</i>	30	0.1	87.56	34.41	29.71	10.04	0.016
Kudritzki <i>et al.</i>	30	0.1	86.52	33.98	29.84	10.37	0.016

The differences in the total mass found at the end of the hot phase are easily understood. The total mass lost is

$$\Delta M \equiv \int_0^{t(T_{\text{eff}}=T_{\text{eff}}^{\text{hot}})} |\dot{M}| dt \sim \langle |\dot{M}| \rangle \Delta t_{\text{hot phase}}, \quad (3.4)$$

and the duration of the hot phase of evolution $\Delta t_{\text{hot phase}}$ is nearly identical for models of the same M_{ZAMS} , while the average mass loss rate $\langle |\dot{M}| \rangle$ depends on the algorithm adopted. In general, the total amount of mass lost using the Kudritzki *et al.* rate is lower than using the Vink *et al.* rate. In other words, the average mass loss rate of Kudritzki *et al.* is lower than of the Vink *et al.*, and the difference decreases with decreasing η , see Tab. 3.3. The average mass loss rate from Vink *et al.* is higher because of its detailed treatment of the bistability jump, see §3.2.1 for more details.

The differences in the helium core mass with the two algorithms are relatively small (few percent of the total mass), but extremely important. They arise in the post-main sequence evolution included in the hot phase, as discussed in more detail in §3.2.2. The relevance of these differences can be understood in terms of the effects they have in the subsequent Supergiant stage, see §3.3.3.

The result that the compactness at the end of the hot phase is not affected by the mass loss rate, and more importantly, that the compactness has roughly the same value for all models, is not really surprising. For $M_{\text{ZAMS}} \geq 15M_{\odot}$ the convective core is much larger than $\mathcal{M} = 2.5M_{\odot}$, which is the mass coordinate chosen to evaluate $\zeta_{\mathcal{M}}$. Furthermore, hydrogen burning occurs via the CNO cycle, whose energy generation rate is very temperature sensitive. Convection, and, more in general, chemical mixing are treated identically in all models. Thus, the differences in the central temperature of models of different mass are small. Therefore, the chemical composition is homogeneous within the innermost $2.5M_{\odot}$, and the thermal structure is very similar in all models. Even the post-main sequence evolution included in the hot phase does not significantly affect the inner $2.5M_{\odot}$ (see also §3.2.2), so all models have roughly the same $\zeta_{2.5}^{T_{\text{eff}}=15[\text{kK}]}$, regardless of their total mass and mass loss rate.

Figures 3.4–3.7, show that using the Kudritzki *et al.* wind scheme on the main sequence produces a very small difference in the evolutionary tracks on the HR diagram between models with $\eta = 0.1$ and $\eta = 0.33$. Additionally, Fig. 3.6 and, to a lesser extent, Fig. 3.7 shows a “loop-like” feature toward the end of the main sequence for models using the Kudritzki *et al.* mass loss rate with $\eta = 1.0$. These loops span the interval $4.3 \lesssim \log(T_{\text{eff}}/[\text{K}]) \lesssim 4.5$ and $5.25 \lesssim \log(L/L_{\odot}) \lesssim 5.3$ for $M_{\text{ZAMS}} = 25M_{\odot}$, while they are much smaller in amplitude, and concentrated around $\log(T_{\text{eff}}/[\text{K}]) \sim 4.4$ and $\log(L/L_{\odot}) \sim 5.4$ for $M_{\text{ZAMS}} = 30M_{\odot}$. They can be explained by considering the M , L and $v_{\text{th}} \propto \sqrt{T}$ dependence of the Kudritzki *et al.* rate (see Tab. 2.2). Note that $v_{\text{th}} = c_s$, where c_s is the local speed of sound, and it would be preferable to express the mass loss rate as a function of the speed of sound at the critical point where $g(r) = g_{\text{rad}}$, i.e. where the mass flux is determined – because the sound speed enters directly in the equation of motion of the gas. Note also that the local temperature T where the mass flux is determined may be very different than the effective temperature T_{eff} , which is only a different way of expressing the radiative flux of the star. However, Fig. 3.11, shows that, at least during the loop at the end of the main sequence, the Kudritzki *et al.* \dot{M} is rather insensitive to T_{eff} , and seems to depend almost solely on L .

Models with $M_{\text{ZAMS}} \geq 25M_{\odot}$ reach a high luminosity toward the end of the main sequence (see the first local maximum of the green curve in Fig. 3.11 – $\log_{10}(L/L_{\odot}) \sim 5.3$ at $t \sim 7.073$ Myr). Therefore, the mass loss rate peaks, and (if $\eta = 1.0$) the total mass is significantly reduced and the luminosity decreases. Therefore, the evolutionary track on the HR diagram is shifted slightly downward, and the mass loss rate is also reduced. With the now decreased mass loss rate, the mass of the star remains high, and the regular evolution (as would happen even without mass loss) makes the luminosity increase again, producing another mass loss peak (at $t \sim 7.0765$ Myr). This changes again the luminosity and produces the loop-like feature at the end of the main sequence for models using the Kudritzki *et al.* algorithm. The process works as if the initial increase of L and thus \dot{M} shifted downward the evolutionary track, producing a “self-damping” of the mass loss rate. MESA re-computes self-consistently the models at each timestep, considering the new boundary condition from modifications at the surface, so the shift on the HR diagram is not caused by interpolation between pre-computed tracks.

The loops increase the amount of time spent by the stars in a given portion of the HR diagram. For $M_{\text{ZAMS}} = 25M_{\odot}$, the increase is of ~ 0.5 Myr (see Fig. 3.1), while it is lower for $M_{\text{ZAMS}} = 30M_{\odot}$, since the evolution of these stars is intrinsically faster because of their higher mass. The longer time spent on the end of the main sequence directly translates

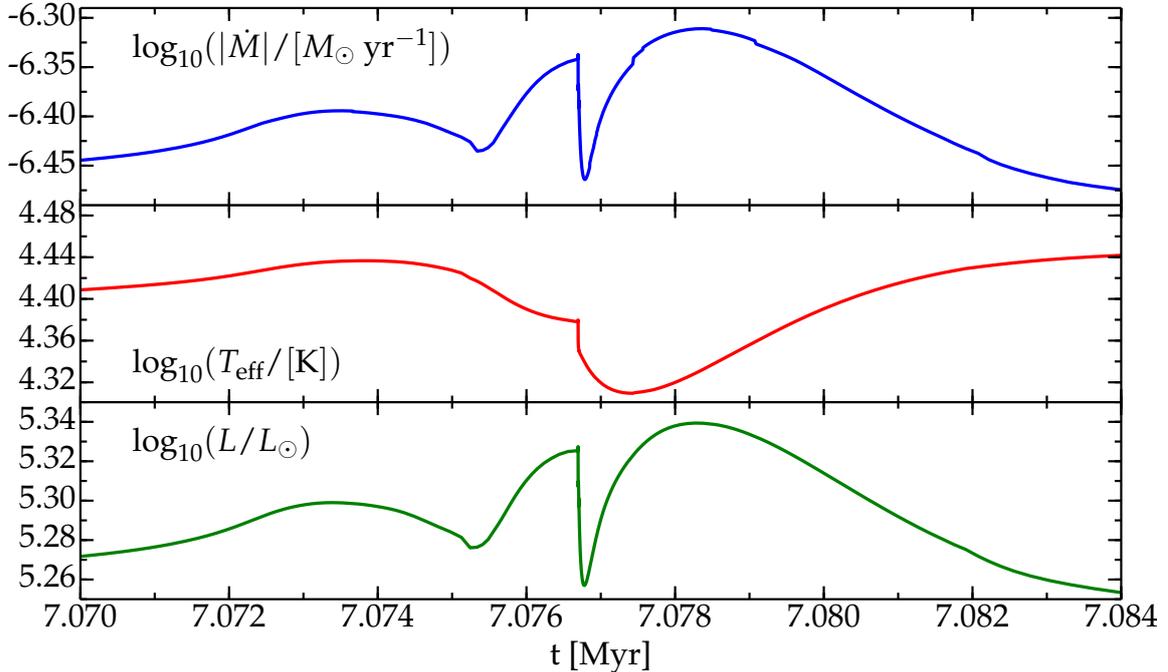


Figure 3.11: Mass loss rate (top panel), effective temperature (center panel), and luminosity (bottom panel) of the $M_{\text{ZAMS}} = 25M_{\odot}$ model computed with the Kudritzki *et al.* algorithm and $\eta = 1.0$ during the “loop” at the end of the main sequence (cf. Fig. 3.6). The initial bump of the luminosity causes an increase of the mass loss rate, that reduces the mass and therefore the luminosity. Then, L increases again, as it would in the regular evolution even without mass loss, and this cause again a second increase in the mass loss rate, resulting in the loop seen in Fig. 3.6.

in a prediction on the relative number of stars in that portion of the HR diagram⁴ – that can be tested observing large scale stellar populations (e.g. field stars – for which however the observational mass determination is harder). Fig. 3.12 shows the number of $M_{\text{ZAMS}} = 25M_{\odot}$ stars as a function of L and T_{eff} . The number of stars in each L and/or T_{eff} bin is proportional to the amount of time spent by the evolutionary track within that bin regardless of the evolutionary stage of the star (e.g. if the bin may be crossed multiple times). A similar histogram can be produced also for $M_{\text{ZAMS}} = 30M_{\odot}$, and it is not shown since it does not show significant differences.

The “loop-like” feature of the $M_{\text{ZAMS}} \geq 25M_{\odot}$ Kudritzki *et al.*, $\eta = 1.0$ computations predicts a star count excess at $\log_{10}(L/L_{\odot}) \sim 5.3$ and $\log_{10}(T_{\text{eff}}/[\text{K}]) \sim 4.42$ in the right panels of Fig. 3.12. The Vink *et al.* and Kudritzki *et al.* mass loss rates produce different cut-offs in the upper luminosity and lower temperature for stars of $M_{\text{ZAMS}} = 25M_{\odot}$. Fig. 3.12 also shows that it is very difficult to use population statistics to determine η , and that with higher η the total mass is reduced more and so the evolutionary time increases, causing a (very small) relative over-population at low luminosities.

Interestingly, as a result of this loop Kudritzki *et al.*, $\eta = 1.0$, $M_{\text{ZAMS}} = 25M_{\odot}$ models have larger radii than models with lower η , despite being less massive. This results in a higher photospheric luminosity ($L \propto R^2$) for given T_{eff} , see Fig. 3.6. Moreover, these models

⁴Note that, at this stage of the evolution, $M(t) \sim M_{\text{ZAMS}}$ is still approximately true.

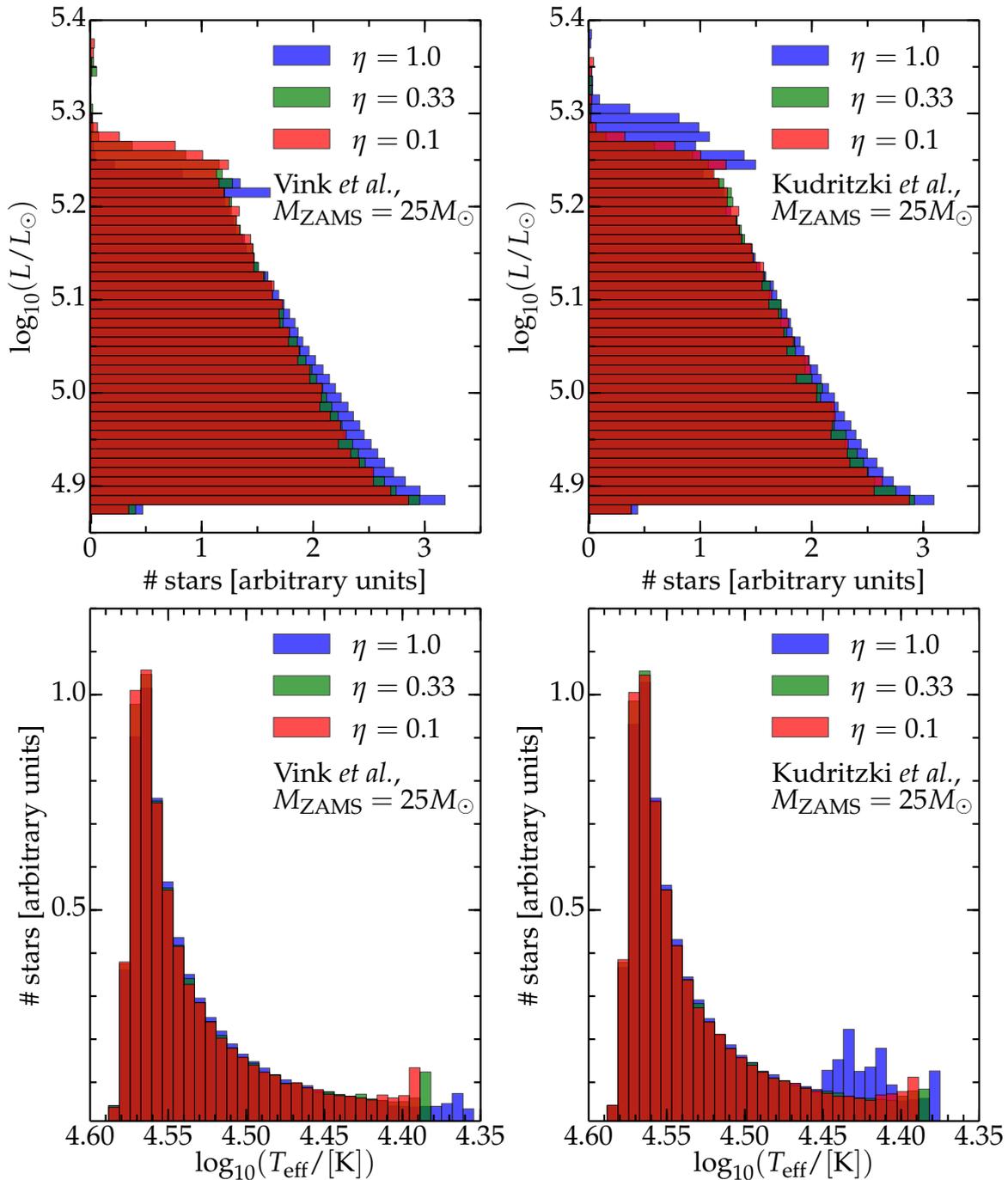


Figure 3.12: Number of stars (proportional to the time spent in the corresponding location of the HR diagram, regardless of the evolutionary stage) as a function of L (top panels) and T_{eff} (bottom panels). The left (right) panels use the Vink *et al.* (Kudritzki *et al.*) mass loss rate.

do not reach as low a T_{eff} as the other $M_{\text{ZAMS}} = 25M_{\odot}$ models (either using Vink *et al.* algorithm during the hot phase or different η), and they reach oxygen depletion as yellow supergiants without ever reaching the red side of the HR diagram ($\log_{10}(T_{\text{eff}}/[\text{K}]) \leq 3.6$, [47]). It is also worth noting that the Kudritzki *et al.*, $\eta = 1.0$ $M_{\text{ZAMS}} = 25M_{\odot}$ models have a longer main sequence duration – because they evolve more slowly during the loop, acting like lower total mass stars. This increases the time to reach oxygen depletion, see the upper panel of Fig. 3.2.

The “loop-like” feature and its effects are less pronounced in the $M_{\text{ZAMS}} = 30M_{\odot}$ models since their evolution is much faster and the initial increase of the mass loss rate is smaller (cf. orange curve in the top panel of Fig. 3.10). Models using lower values of η do not have strong enough mass loss to trigger this loop (L increases at the end of the main sequence as well, but the corresponding increase of \dot{M} is not sufficient), while models using the Vink *et al.* algorithm have a very different mass loss rate in the temperature range $4.3 \leq \log_{10}(T_{\text{eff}}/[\text{K}]) \leq 4.5$ (i.e. where the loop-like feature appears), because of the bistability jump, see §3.2.1.

3.2.1 Effects of the detailed Treatment of the Bistability Jump in the Vink *et al.* Algorithm

Figure 3.10 shows that the treatment of the bistability jump in the Vink *et al.* algorithm produces the large increase of the mass loss rate toward the end of the solid curves: \dot{M} increases by roughly one order of magnitude when⁵ $T_{\text{eff}} \sim T_{\text{jump}} \stackrel{\text{def}}{=} 25000$ K. The details of the very end of the curves depend on morphology of the overall contraction phase and the beginning of the Sub-Giant Branch. In the $M_{\text{ZAMS}} = 25M_{\odot}$ model, there is a large drop after the first increase (around $t \sim 7$ Myr), because T_{eff} decreases and crosses the bistability jump temperature at the end of the main sequence and then increases again during the overall contraction phase (see Fig. 3.6). A similar behavior is also found for $M_{\text{ZAMS}} = \{20, 30\}M_{\odot}$ with lower η .

Figure 3.10 also shows that for the $M_{\text{ZAMS}} = 15M_{\odot}$ models, the Vink *et al.* mass loss rate is lower than the Kudritzki *et al.* before the bistability jump (i.e. at higher temperature, when the most common Fe ion is Fe IV). Increasing M_{ZAMS} the difference between the two schemes at high temperatures decreases, until the Vink *et al.* rate is higher for almost the entire hot phase (i.e. both above and below T_{jump}).

The relative increase of the Vink *et al.* mass loss rate with respect to the Kudritzki *et al.* with increasing M_{ZAMS} is due to their different dependencies on the luminosity L and total mass M (see also Tab. 2.2). Note that at temperatures lower than $T_{\text{jump}} \simeq 25000$ K (i.e. after the bistability jump in the evolution of stars), the Vink *et al.* mass loss rate is always higher, cf. Fig. 3.10.

The Vink *et al.* algorithm used for this study (see also Fig. 2.7 and §A.2) is a modification of the default MESA routine for these wind scheme (see `$MESA_DIR/star/private/wind.f`) to smooth the jump. Note that the algorithm described in [25, 64] prescribes an inversion of the function $\dot{M}(T_{\text{eff}})$ (see also Fig. 1.12). Instead, the routine used here interpolates between the formulae below and above the temperature region of the inversion. The steepness of the solid curves in Fig. 3.10 indicates, however, that the duration of the bistability jump is very brief and the corresponding inversion lasts very shortly.

The presence of the bistability jump in the Vink *et al.* algorithm is the most noticeable qualitative difference compared with the Kudritzki *et al.* algorithm. This leads to a significant difference in the helium core structure of the models, as I discuss in the next section §3.2.2.

⁵Obviously, this happens at different times for different M_{ZAMS} : ~ 11 Myr for $15M_{\odot}$, ~ 8.5 Myr for $20M_{\odot}$, ~ 6.5 Myr for $25M_{\odot}$, and ~ 5.5 Myr for $30M_{\odot}$.

3.2.2 Effects of the Hot Phase Mass Loss on the Helium Core

The helium core mass at the end of the hot phase (i.e. when T_{eff} decreases below $T_{\text{eff}}^{\text{hot}} \stackrel{\text{def}}{=} 15000$ K), and the helium abundance profile at the helium core boundary, are important parameters for the subsequent evolution (e.g. [28] and references therein). During the subsequent stages, the hydrogen burning shell propagates outward from the helium core boundary. This helium rich region exists because of the well known contraction of the convective core during the main sequence, see §1.2.1. When the convective core contracts, no more mixing happens in the regions left at its outer boundary. Because of convection, these regions had the same chemical composition of the hydrogen burning core, therefore they remain partially hydrogen depleted (i.e. helium rich). If an exponential overshooting scheme is used, the hydrogen and helium profile left are smooth, see the dashed curves in Fig. 3.13. The details of the structure and composition in this region can trigger a blueward evolution on the HR diagram [28], and produces, in some cases, extended blue loops, e.g. see Fig. 3.4 (see also §3.3.3).

The helium profiles at TAMS (dashed curves) and at the end of the hot evolutionary phase (solid curves) are shown in Fig. 3.13 for the $M_{\text{ZAMS}} = 15M_{\odot}$ models computed with $\eta = 1.0$. Note that the two helium profiles at TAMS are practically identical, regardless of the mass loss algorithm.

The helium core mass at the end of the hot evolutionary phase is determined by the post-

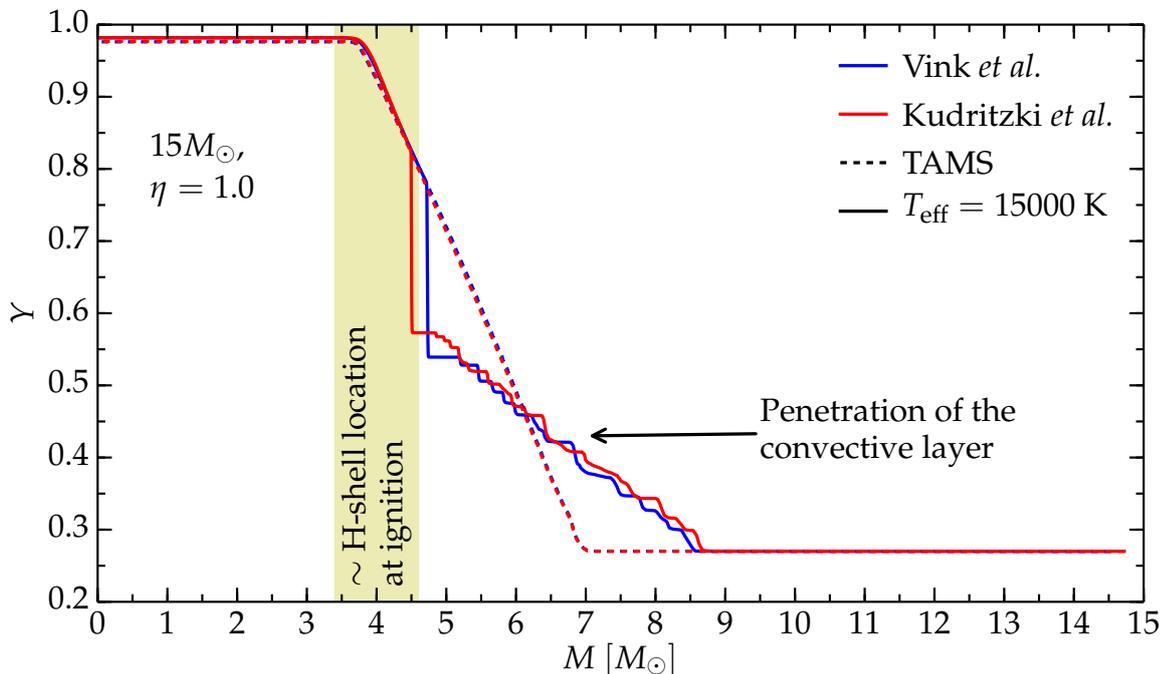


Figure 3.13: Helium abundance profile for $M_{\text{ZAMS}} = 15M_{\odot}$, $\eta = 1.0$ models computed with the Kudritzki *et al.* rate (red curves) and Vink *et al.* rate (blue curves), respectively. Dashed curves are at TAMS, solid curves are at the end of the hot evolutionary phase. The shadowed region indicates the approximate location of the hydrogen shell at shell ignition. The step-like profile of the solid curves, indicated by the arrow, is produced by the penetration of the convective region into the region that was convective at ZAMS. Each step corresponds to a different penetration episode.

main sequence evolution, specifically by the penetration depth of the envelope convective region above the hydrogen burning shell, see Fig. 3.13 and Fig. 3.14. The convective region exists because of the rapid expansion of the envelope – in response to the continued contraction of the inert helium core – that steepens the temperature gradients in the inner envelope (see also §1.2.3). Convective mixing injects fresh hydrogen from the outer layers into the region where hydrogen is partially depleted. This transforms the smooth⁶ helium abundance profile into a series of abrupt steps (each corresponding to a different episode of penetration). This creates a clear edge of the helium core, whose location depends on the depth of penetration of the convective region. Note that this process depends strongly on the parameters adopted for convective mixing (described in §2.3).

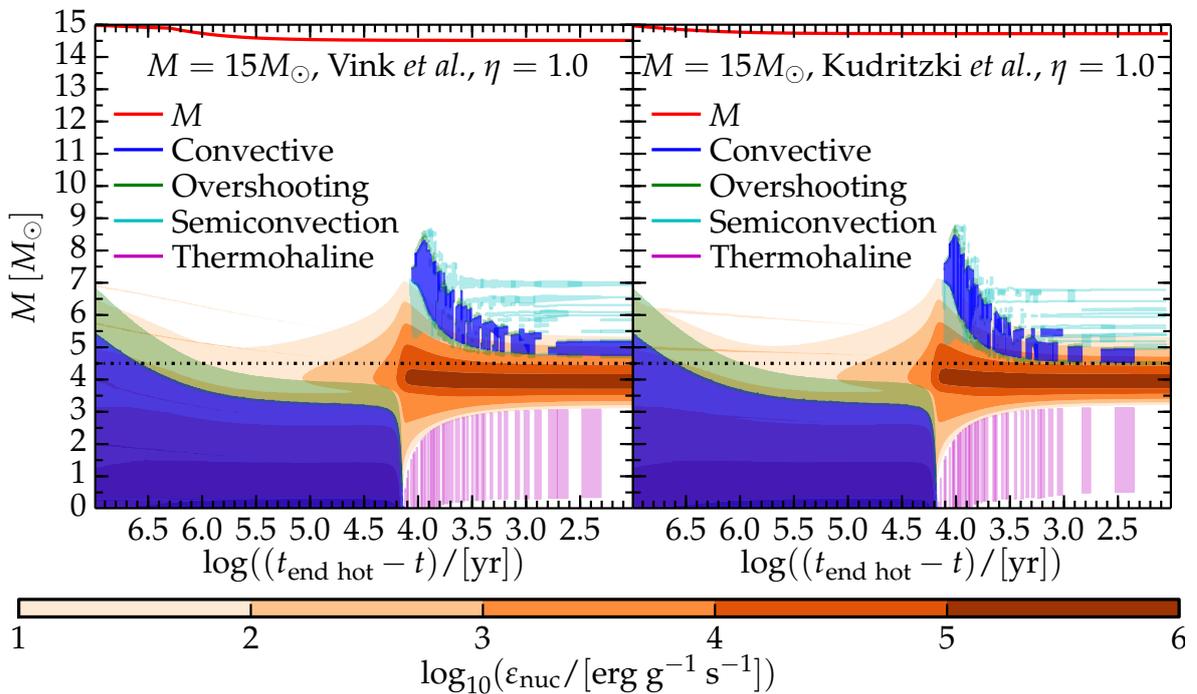


Figure 3.14: Kippenhahn plots for the energy generation rate and mixing processes in $M_{\text{ZAMS}} = 15M_{\odot}$ models computed to the end of the hot evolutionary phase. The model on the left panel uses the Vink *et al.* rate, the model on the right uses the Kudritzki *et al.* rate, both with efficiency $\eta = 1.0$. The black dot dashed line marks the edge of the helium core at the end of the hot phase for the model on the right panel, to emphasize the different penetration depth of the convective layers on the two panels.

This is especially relevant for the $M_{\text{ZAMS}} = 15M_{\odot}$ models, whose density profile at $T_{\text{eff}} = T_{\text{eff}}^{\text{hot}}$ is shown in Fig. 3.15 (see also Fig. 3.13). The helium core boundary is the rapid drop in density (because of the change in the chemical composition and mean molecular weight μ).

When using the Kudritzki *et al.* wind scheme and $\eta = 1.0$, the helium core mass at $T_{\text{eff}} = T_{\text{eff}}^{\text{hot}}$ is similar to that obtained with the Vink *et al.* with $\eta = 0.33$ (see also Tab. 3.3). But, even more importantly, the helium core mass M_{He} at that point from the Vink *et al.*, $\eta = 1.0$ models is much greater (by at least $\sim 0.2M_{\odot}$) than M_{He} for all the other models, see

⁶Such a profile would not appear to be smooth if a step-function overshooting algorithm were used, instead of the exponential overshooting used in MESA, see Eq. 2 in [81].

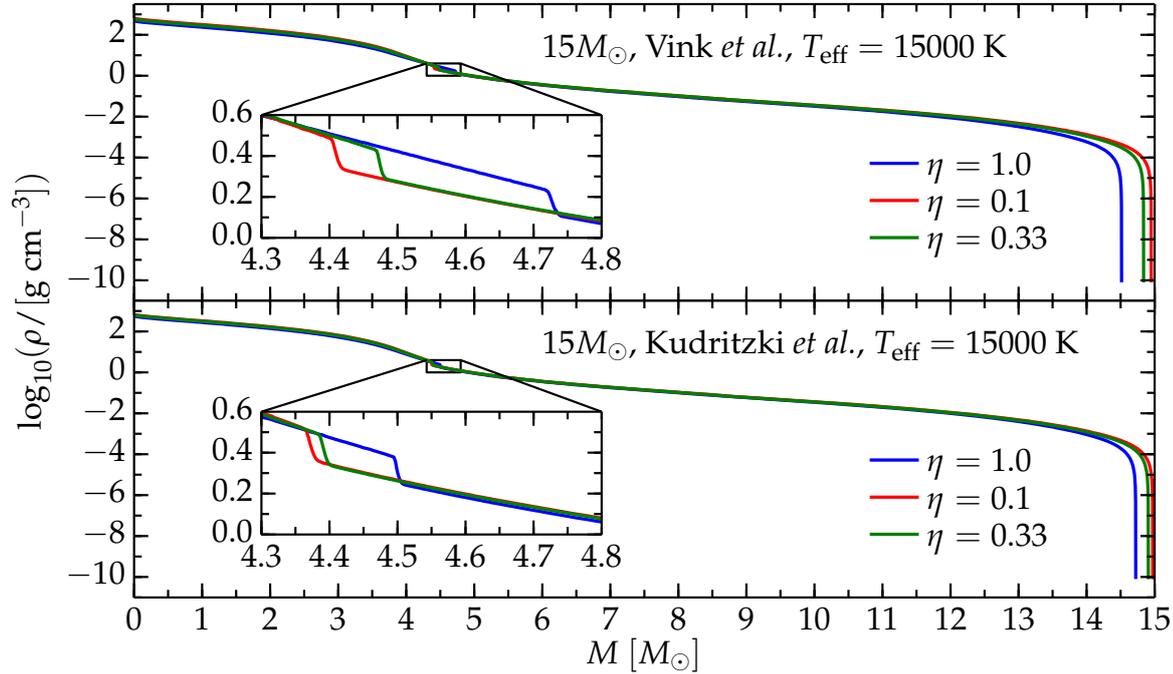


Figure 3.15: Density structure when $T_{\text{eff}} = 15000$ K for the $M_{\text{ZAMS}} = 15M_{\odot}$ models computed with the Vink *et al.* (upper panel) or Kudritzki *et al.* (lower panel) hot phase mass loss rate. The zoomed plots focus on the edge of the helium core.

Fig. 3.15. This perplexing result can be explained in terms of the total mass of the models at the end of the hot phase of evolution. The downward mixing of hydrogen by the convective shell happens after the Vink *et al.* rate exceeds that of Kudritzki *et al.* – because of the detailed treatment of the bistability jump. Models that recently had a lower mass loss rate (either because of η or the chosen algorithm) have a more massive envelope. Therefore, a slightly higher energy generation rate (in the hydrogen burning shell) is needed to sustain this more massive envelope. So the region including the hydrogen shell and the base of the envelope contracts slightly, to increase the shell temperature and energy generation rate. However, because of the steep temperature dependence of the CNO-cycle energy generation rate (cf. Eq. 1.6), a very small increase in temperature is enough to provide the required luminosity. The outermost envelope responds by expanding (and thus cooling) more (as the *R* and *L* entries in Tab. 3.3 confirm). The result is a steeper temperature gradient above the hydrogen shell for more massive models, since their hydrogen shell is slightly hotter, with a more extended envelope and lower effective temperatures. The steeper temperature gradient makes a larger region convectively unstable, therefore the convective base of the envelope penetrates deeper – producing deeper hydrogen mixing and a smaller helium core (however, see also §3.3.3). This is possible because the envelope (and, in general, the star) is not a rigid structure, but instead each region re-adjusts on the thermal timescale, that depends on the local thermodynamical state each layer.

While the mass difference between the helium cores of the $M_{\text{ZAMS}} = 15M_{\odot}$ models at the end of the hot phase may seem small ($\sim 1\%$), it affects the subsequent evolution: all the $M_{\text{ZAMS}} = 15M_{\odot}$ models computed with the Vink *et al.* wind scheme and $\eta = 1.0$ show a blueward evolution (either a complete blue loop during helium core burning or a deviation toward higher T_{eff} , depending on the cool phase mass loss rate adopted, see also §3.3.3) in later stages of their evolution, see Fig. 3.4. In contrast, models using $\eta < 1$ or the Kudritzki

et al. rate (regardless of η) do not experience such loops.

The tailoring of the helium core extent by the penetration of the convective base of the envelope seems to be inefficient in the $M_{\text{ZAMS}} \geq 25M_{\odot}$ simulations. The analysis of the $M_{\text{ZAMS}} = 25M_{\odot}$ models is complicated by the loop-like feature (discussed in §3.2, and shown in Fig. 3.6) produced by the Kudritzki *et al.* rate. However, the correlation between a larger M_{He} at $T_{\text{eff}} = 15000$ K and a blueward evolution during the RSG phase of the evolution is found in all models (compare Tab. 3.3 and Fig. 3.4–3.7).

3.3 Cool Phase Mass Loss

Figures 3.1 and 3.2 show that most of the mass is lost after the main sequence, especially during the cool phase. This is shown in Fig. 3.16, which shows the mass loss rate for all models computed with $\eta = 1.0$. The mass loss rate during this phase is several orders of magnitude larger than on the hot phase ($\dot{M} \sim 10^{-5.25} - 10^{-4.75} M_{\odot}\text{yr}^{-1}$ during the cool phase, while $\dot{M} \lesssim 10^{-6.25} M_{\odot}\text{yr}^{-1}$ earlier in the hot phase). Note that, in Fig. 3.16, different values for η does *not* result in a simple shift of the plot on the ordinates: the morphology of the curves for $\eta = 1.0$ is also qualitatively different from the the curves corresponding to lower η . For $\eta = 1.0$, pronounced blue loops or blueward evolutionary features (i.e. decreases of the mass loss rate) are found.

The increase in the mass loss rate during the cool phase can be understood in terms of the stellar radius: during this phase the stellar radius can exceed several hundred R_{\odot} , therefore, they have small effective surface gravity ($g \propto R^{-2}$), and it is easier for particles to leave the gravitational well of the star. Note that this aims to be a physical justification, none of the mass loss algorithms used include an *explicit* dependence $\dot{M} \equiv \dot{M}(g)$.

During the cool phase, mass loss can be dust-driven, see e.g. [62], although the process of dust formation in RSG envelopes is still debated in the literature. Even using algorithms based on line-driven processes, wind mass loss during this stage is both very large and very uncertain. The empirical determination of the mass loss rate in this phase is complicated by the possible impulsive mass loss events (e.g. wave driven, pulsational instabilities, LBVs eruptions), which can dramatically reduce the total mass without any contribution from stellar winds.

The cool mass loss rate hardly influences the helium and carbon-oxygen core masses and the final radii. For $M_{\text{ZAMS}} = \{15, 20\}M_{\odot}$ models, the van Loon *et al.*, $\eta = 1.0$ rate produces smaller radii (see §3.3.3), but this is not true for higher initial masses (see §3.3.2).

The modeled effective temperature at oxygen depletion depends on the amount of mass lost during the cool phase and on how deep the layer revealed by mass loss is. Since the subsequent, post-oxygen-depletion evolution is too fast to influence the stellar surface, the effective temperature at oxygen depletion (roughly) corresponds to the effective temperature at the onset of core collapse. Therefore, mass loss may play a role in the solution of the so-called ‘‘RSG problem’’ (see §1.2.3). However, eruptive mass loss could be much more important winds to solve the problem. In fact, in the grid presented here, no proper BSG (with $\log_{10}(T_{\text{eff}}/[\text{K}]) \gtrsim 3.8$, [47]) is produced, even if many models – especially those with $\eta = 1.0$ – reach oxygen depletion after a slight blueward shift on the HR diagram, and may be considered as YSG (with $3.6 < \log_{10}(T_{\text{eff}}/[\text{K}]) < 3.8$, [47]), see Fig. 3.5–3.7.

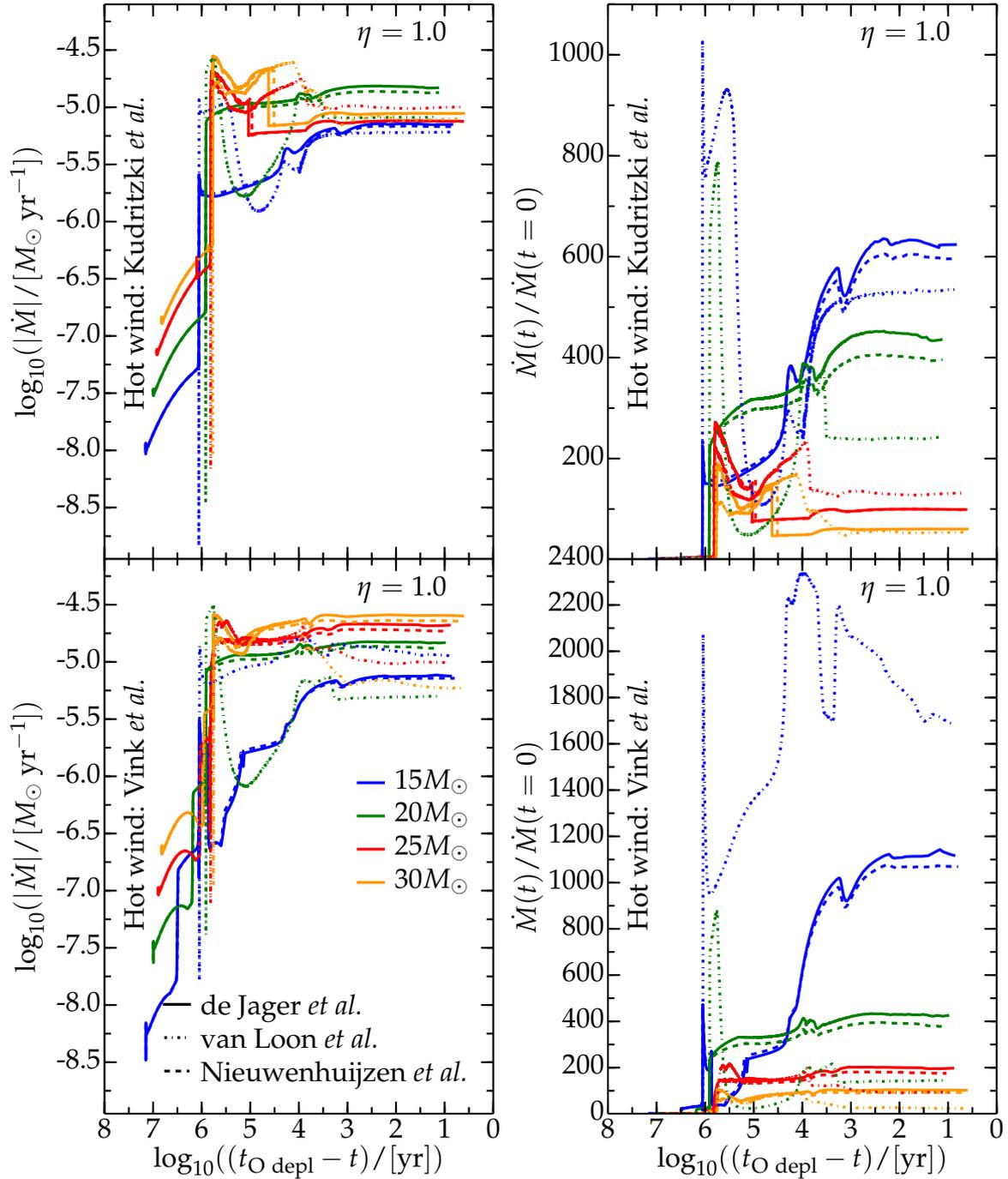


Figure 3.16: Mass loss rate with different algorithms and $\eta = 1.0$. Solid, dashed and dot dashed curves correspond to the use of the de Jager *et al.*, the Nieuwenhuijzen *et al.*, and the van Loon *et al.* cool wind algorithm, respectively. Each color corresponds to an initial M_{ZAMS} . The upper (lower) panel is for models computed using the Kudritzki *et al.* (Vink *et al.*) scheme during the hot phase of the evolution. Note that I do not show the \dot{M} reduction that is artificially applied by MESA when $T_c \geq 10^9$ K. During blue loops, \dot{M} decreases as the star shrinks in radius. WR models are *not* omitted: they correspond to the models with $M_{\text{ZAMS}} \geq 25M_{\odot}$ using the de Jager *et al.* or Nieuwenhuijzen *et al.* schemes in the upper panel (i.e. red and orange solid and dashed curves). The switch to the WR wind scheme corresponds to the vertical decrease of \dot{M} around $\log_{10}((t_{\text{O depl}} - t)/[\text{yr}]) \sim 5$.

3.3.1 Comparison of the Jager *et al.* and Nieuwenhuijzen *et al.* Algorithms

It is very hard to find systematic qualitative and quantitative differences between the de Jager *et al.* and the Nieuwenhuijzen *et al.* mass loss rates – as already noticed in [45]. The spread between the final total masses, the helium core and carbon-oxygen core masses, and the stellar radii increases slightly with η , but it is always very small – see Tab. 3.2.

The lack of significant differences is, however, not surprising, as both algorithms are derived from the same data set. From Tab. 2.2, and Eq. 1.42 and Eq. 1.45, it may seem that the two algorithms have little in common as they use different variables to express \dot{M} , and therefore they have also different coefficients. But, it must be stressed that the variables chosen are likely not independent (e.g. because of the well known empirical relations $L \equiv L(M)$), and they may not be a complete set – in the sense that they may not completely and uniquely parametrize the mass loss rate. This last caveat applies to all mass loss algorithms compared in this thesis.

From the theoretical point of view, it may be preferable to use the de Jager *et al.* rate. This can be interpreted directly as a relation between the mass loss rate and the position on the HR diagram (as it is in the form $\dot{M} \equiv \dot{M}(L, T_{\text{eff}})$), but it has a meaning only in the sense of a *statistical average* over the possible behaviors at a given point of the HR diagram. While the Nieuwenhuijzen *et al.* rate gives very similar results, it suffers from the model-dependence introduced to determine the total mass of the “average” star at any point of the HR diagram, which is conceptually disturbing.

3.3.2 The Van Loon *et al.* Algorithm

Models of $M_{\text{ZAMS}} \lesssim 25M_{\odot}$ computed using the van Loon *et al.* rate, regardless of η , reach oxygen depletion with much smaller final masses than those computed with other cool phase mass loss algorithms. This is shown in Fig. 3.1, Fig. 3.2, and Tab. 3.2. In contrast, for $M_{\text{ZAMS}} \gtrsim 25M_{\odot}$ – the exact threshold depending also on which hot phase mass loss is used – $M_{\text{O depl}}$ is larger for models using the van Loon *et al.* rate. This is because at high M_{ZAMS} the van Loon *et al.* rate is so large, that the star loses rapidly a large amount of mass and reveals its hotter and deeper regions sooner. At higher effective temperatures, the mass loss rate is generally smaller ($T_{\text{eff}} \propto R^{-2}$ for a given L , so higher T_{eff} correspond to smaller R and larger g). In fact, the red and orange dot-dashed curves in Fig. 3.16 (especially in the top panel) first rise above the others, and then decrease when T_{eff} is so high that \dot{M} is reduced.

The rather extreme mass loss rate provided by the van Loon *et al.* algorithm is caused by its very steep temperature dependence, $\dot{M} \propto T_{\text{eff}}^{-6.3}$, cf. Tab. 2.2. This comes from the physical picture assumed for the description of the wind, i.e. the dust-driven (instead of line-driven) process.

It is worth reminding the reader that, although the van Loon *et al.* mass loss rate is widely used for the simulation of massive RSG, see e.g. [46–48, 61], it was derived from a sample of dust-enshrouded pulsating AGB stars in the LMC, and was found to have a good agreement with a small sample of galactic RSGs for which mass loss had been determined with many assumption from observations (see §5 of [62]). One reason the van Loon algorithm is widely used is because of its extremely high mass loss rate, thought to be suitable (with an appropriate choice of $\eta \geq 1$) for reproducing the maximum mass shed by a star during its RSG evolution (including eruptive, dynamical events). Nonetheless, the physical conditions in the layers of the star at which mass loss is determined may be significantly different from

those assumed by Van Loon *et al.*

Fig. 3.4 and Fig. 3.5 show that the high mass loss rate provided by Van Loon *et al.* with $\eta = 1.0$ drives a blueward evolution in all the models with $M_{\text{ZAMS}} \leq 20M_{\odot}$, regardless of the initial conditions (i.e. which hot phase mass loss algorithm is used previously). The precise morphology of the evolutionary track on the HR diagram, however, depends on the structure remaining after the hot phase. The blueward evolution of all models in the grid is studied in more detail in §3.3.3.

3.3.3 On the Morphology of the Blueward Evolution and Blue Loops

Figure 3.4 shows two different kinds of blueward evolution for $M_{\text{ZAMS}} = 15M_{\odot}$ models computed with $\eta = 1.0$ (blue curves). The first is the blue loop found in the top left and center left panels of Fig. 3.4. The second, characterized by the absence of a loop (the tracks do not go back to the red and cool side of the HR diagram), is found in the two bottom panels of Fig. 3.4, and analogous tracks can also be seen in the two bottom panels of Fig. 3.5, referring to simulations with $M_{\text{ZAMS}} = 20M_{\odot}$.

The first kind of blueward evolution, i.e. the proper blue loop of the top and center left panels of Fig. 3.4, lasts $\sim 6 \times 10^5$ yr and happens during helium core burning. Fig. 3.17 shows the radius of the $M_{\text{ZAMS}} = 15M_{\odot}$ models, the blueward evolution correspond to a decrease in radius. The blueward evolution can also be seen as a decrease of the mass loss rate (because of the decrease in R and thus the increased surface gravity of the star) in the dashed and dot-dashed blue curves in the bottom panel of Fig. 3.16.

Both the HR diagrams and Fig. 3.17 show that the differences between the combination of algorithms Vink *et al.* plus either de Jager *et al.* or Nieuwenhuijzen *et al.* with $\eta = 1.0$ are very small (as discussed in §3.3.1), especially for $\eta < 1$. Moreover, the use of the same cool mass loss algorithms (de Jager *et al.* or Nieuwenhuijzen *et al.*) combined with the Kudritzki *et al.* mass loss rate on the hot side of the evolution does not result in blue loops (see the top and center right panels in Fig. 3.4). This indicates that the first kind of blueward evolution (i.e. the blue loops) must be caused by the structural differences developed during the hot phase of the evolution.

The blueward evolutions of the first kind (i.e. the blue loops in the top left and center left panels of Fig. 3.4) for models of $M_{\text{ZAMS}} = 15M_{\odot}$ start at ~ 13.3 Myr, when the hydrogen burning shell reaches the edge of the helium core⁷. This also corresponds roughly to when the convective envelope reaches its maximum extent, completely mixing the chemical composition above the hydrogen shell and smoothing the steps of Fig. 3.13.

Therefore, at the onset of the first kind of blueward evolution, the hydrogen burning shell reaches a region where much more fuel is available, and this results in a momentary increase of the energy generation rate. The blue loops are a result of the re-adjustment (on a thermal timescale) of the stellar structure to the increased shell luminosity. For models with $\eta < 1$, no blueward evolution occurs because the edge of the helium core lies much deeper in the star, and the surface is effectively decoupled from the shell when the latter reaches the edge of the helium core. All re-adjustments take place at the base of the envelope without affecting the surface (note also that the radii of models with $\eta < 1$ are larger).

⁷Note that the hydrogen shell has only consumed the hydrogen left in the helium rich region so far, see Fig. 3.13. Therefore the helium core mass is still roughly the same it was at the end of the hot evolutionary phase.

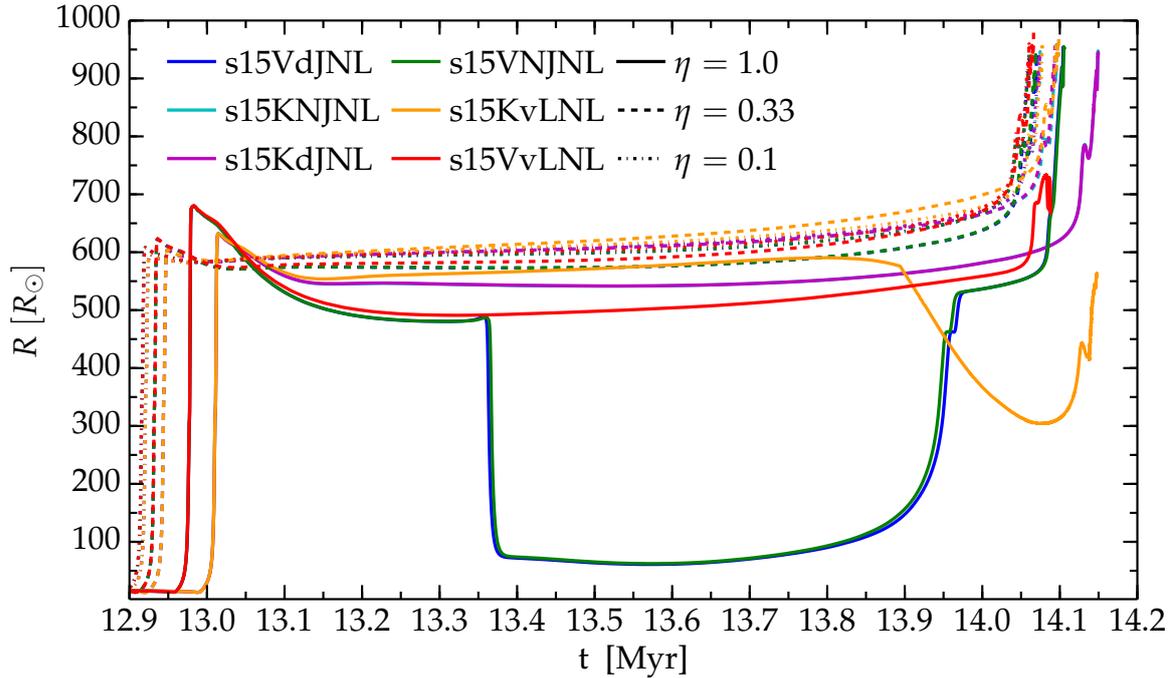


Figure 3.17: Radius as a function of time for the $M_{\text{ZAMS}} = 15M_{\odot}$ simulations. Solid, dashed, and dot dashed curves correspond to $\eta = 1.0$, $\eta = 0.33$, and $\eta = 0.1$, respectively. The wind scheme used are labeled according to Tab. 2.1. The blue loops correspond to the radius decrease between ~ 13.35 and ~ 13.39 Myr (green and blue solid curves). The second kind of blueward evolution can be seen as a radius decrease in the late stages for the orange (starting at ~ 13.85 Myr) and red (starting at ~ 14.05 Myr) solid curves.

If the van Loon *et al.* rate with efficiency $\eta = 1.0$ is used, the total mass of the star at $t \simeq 13.35$ Myr is already much reduced (see the top panel of Fig. 3.1) – in particular, it is already smaller than the final total mass obtained using the other mass loss algorithms. However, the radius of these models is roughly equal to the radii of other models using $\eta = 1.0$ (see Fig. 3.17), so the envelope density is much lower. Fig. 3.18 shows the density profile of $M_{\text{ZAMS}} = 15M_{\odot}$ models computed with the Vink *et al.* rate during the hot phase, and $\eta = 1.0$. As expected, the density of the outer portion of the model using the van Loon *et al.* cool wind scheme is much lower, while models using the Nieuwenhuijzen *et al.* and the de Jager *et al.* rate are practically identical. Note that the hydrogen shell source, indicated by the shaded region, lies at the edge of the helium core, cf. Tab. 3.3. The lower envelope densities explain why the models using the Vink *et al.* and van Loon *et al.* combination with $\eta = 1.0$ do not experience the first kind of blueward evolution: the thermal timescale of the outer envelope for these models is longer, and the re-adjustment is confined to the inner envelope, without affecting the surface of the star.

As models using the van Loon *et al.*, $\eta = 1.0$ rate continue their expansion, without any early blueward evolution, they continue to experience a high mass loss rate (see also Fig. 3.16). This, in turn, exposes deeper layers of the star and causes the second type of blueward evolution, i.e. the one found in the bottom panels of Fig. 3.4 and Fig. 3.5, where the van Loon *et al.* rate with $\eta = 1.0$ is used, regardless of the hot wind scheme adopted. Therefore, both the damping of the first kind of blueward evolution and the onset of the second kind of blueward evolution later on are an effect of the very high mass loss rate provided by the van Loon *et al.* algorithm.

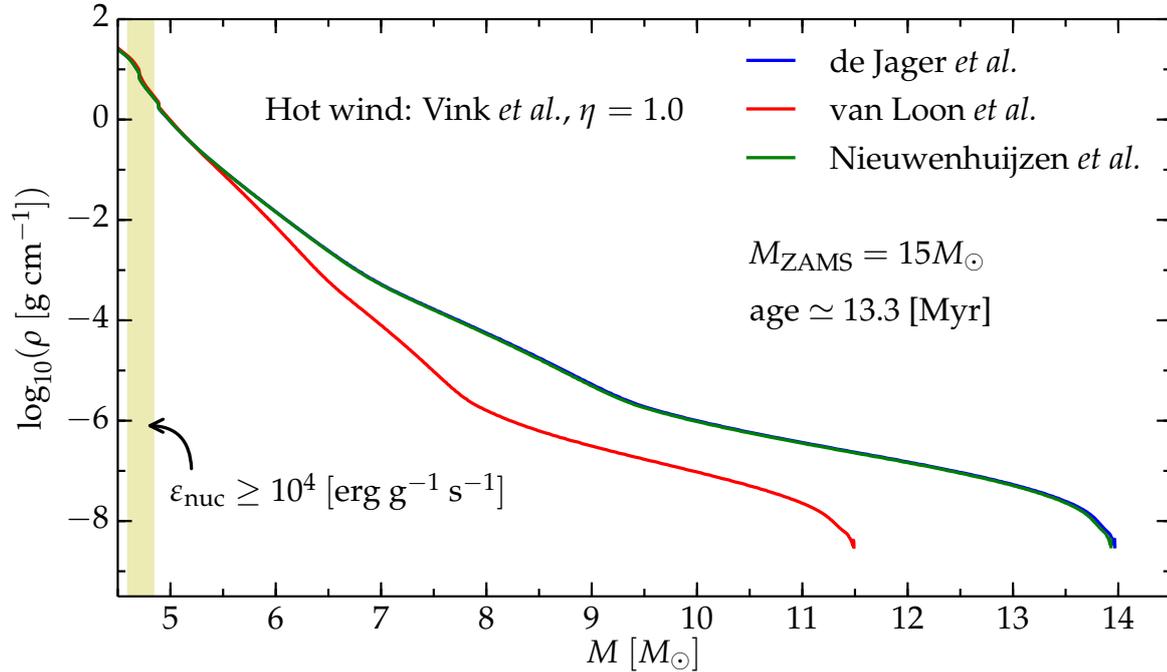


Figure 3.18: Outer density profiles of $M_{\text{ZAMS}} = 15 M_{\odot}$ models before the onset of the blue loops (first kind of blueward evolution). The Vink *et al.* rate is used during the hot phase, the legend indicates the cool wind scheme, the efficiency is $\eta = 1.0$. The shaded region corresponds roughly to the hydrogen burning shell, which lies inside the helium core.

Other examples of blue loops (of the first kind) happening during helium core burning can be seen in $M_{\text{ZAMS}} = \{25, 30\} M_{\odot}$ models using the Kudritzki *et al.* rate with $\eta = 1.0$ during the hot phase, regardless of the cool wind scheme adopted (see Fig. 3.6–3.7). However, for models using the van Loon *et al.* rate during the cool phase of evolution the extent of the loop is smaller.

The presence and morphology of the blue loop are highly uncertain, because they depend strongly on details of the core contraction during the main sequence and the convective envelope penetrating into the old core at the end of the hot evolutionary phase. Moreover, the spatial and temporal variations of quantities in the star must be properly resolved⁸, since the precise location of the helium core edge is so important for the onset of these blue loops. A possible observational probe of the presence of such blue loops is the estimate of the relative populations of red, blue and yellow supergiants in a cluster, to compare with the evolutionary tracks presented here. However, even if this were possible, the large degeneracy between the parameters (e.g. overshooting length, α_{mlt} , wind efficiency and scheme, etc...) that may cause a blue loop would not yield any conclusions regarding the correctness of the the wind algorithm used during the hot phase and its efficiency (i.e. Vink *et al.* with $\eta = 1.0$).

3.4 WR Mass Loss

As already mentioned in §2.2.2, the algorithmic definition of a WR star used in this work, i.e. $X_s < 0.4$, is far from the observational definition of a WR star. A real star is classified

⁸This has been carefully checked for the simulations presented here, see §2.3.

as a WR star based on features in its spectrum, namely, the hydrogen depletion⁹ (e.g. [97]), the presence of broad emission lines (e.g. [98, 99]) indicating a strong and dense wind with a very steep density gradient, and high ratio v_∞/v_{esc} . Moreover, without computing the synthetic stellar spectra, it is impossible to discriminate between the various sub-classes of WR stars defined in the literature (WN, WC, WO, WNL, e.g. [90, 98, 100]). However, it is common practice in stellar evolution simulations to adopt a simple condition on the surface hydrogen abundance (sometimes coupled to a condition on T_{eff}) to switch to a WR wind mass loss algorithm, something that would never happen in a real star.

The purpose of this section is to compare two WR mass loss algorithms: Nugis & Lamers (§1.4.7), and Hamann *et al.* (§1.4.8). Other mass loss algorithms for WR winds are available in the literature, but they apply to specific subsets of WR stars, e.g. Grafener *et al.* [43] propose an algorithm for WR stars of the WNL subclass, expected to be more massive than $30M_\odot$.

The models available for this comparison lose enough hydrogen envelope to reach the $X_s < 0.4$ threshold, i.e. those with $M_{\text{ZAMS}} = \{25, 30\}M_\odot$ using the combination of the Kudritzki *et al.* and either Nieuwenhuijzen *et al.* or de Jager *et al.* rates, with efficiency $\eta = 1.0$. This means that eight ($=2 M_{\text{ZAMS}} \times 2$ cool wind schemes $\times 2$ WR wind schemes) of the 144 simulations in this grid require the switch to the WR mass loss algorithm. The HR diagrams for these models are the blue curves in the uppermost right, top center right, and bottom panels of Fig. 3.6 ($M_{\text{ZAMS}} = 25M_\odot$) and Fig. 3.7 ($M_{\text{ZAMS}} = 30M_\odot$).

Other models get very close to this algorithmic WR star definition: $M_{\text{ZAMS}} = 25M_\odot$ models using the Kudritzki *et al.* and the van Loon *et al.* rates with $\eta = 1.0$ (blue curve in the lower center left panel of Fig. 3.6) reach oxygen depletion with $X_s = 0.401$; $M_{\text{ZAMS}} = 30M_\odot$ models using the Vink *et al.* rate and the de Jager *et al.* rate (blue curve in the uppermost left panel of Fig. 3.7) or the Nieuwenhuijzen *et al.* (blue curve in the top center left panel of Fig. 3.7) with $\eta = 1.0$ end with $X_s = 0.404$ and $X_s = 0.406$, respectively. A small variation of the parameter set adopted may make these stars reach the $X_s < 0.4$ threshold. For $M_{\text{ZAMS}} \leq 20M_\odot$ no models approach the WR threshold.

It is worth emphasizing that none of the models using a lower efficiency $\eta < 1.0$ (e.g. to account for the wind clumpiness) get close to the WR threshold. In other words, if the wind mass loss rates are indeed overestimated – as it has been suggested many times (e.g. [7, 49]) – then single massive stars with $M_{\text{ZAMS}} \leq 30M_\odot$ are not WR progenitors¹⁰. This fact does not rule out the possibility that wind rates are overestimated, since other WR formation channels are possible and may be dominant (e.g. binaries, eruptive and/or pulsational events that can remove the hydrogen envelope partially or completely), and/or WR progenitors may be more massive on the ZAMS (see also §3.1, and, e.g. [30, 90]). However, the lack of WR models with $\eta < 1.0$ contradicts the so-called “Conti scenario” (see e.g. [5, 78, 91] and references therein) for the evolution of single massive stars, as e.g. [78] also suggests. The “Conti scenario” is the standard empirical picture for the evolution of a single massive star, which, for $M_{\text{ZAMS}} \lesssim 40M_\odot$ predicts the evolution from O giant, to RSG, to (in some cases) LBV and finally WR star, before exploding as type Ib/Ic SN, [5]. To be more detailed, the complete picture of the “Conti scenario” is the following evolutionary sequence, [7, 91]:

$$\text{O} \rightarrow \text{Of/WNH} \rightarrow \text{RSG} \rightarrow \text{LBV} \rightarrow \text{WN} \rightarrow \text{WC} \rightarrow \text{WO} \rightarrow \text{Type Ib/Ic SN} .$$

⁹Note that spectroscopy provides only *relative* abundances.

¹⁰However, this result depends on the many assumptions made to produce the models: the treatment of convection and over/undershooting which determine the core dimensions, the algorithmic definition of a WR star adopted, etc. See also §3.1.

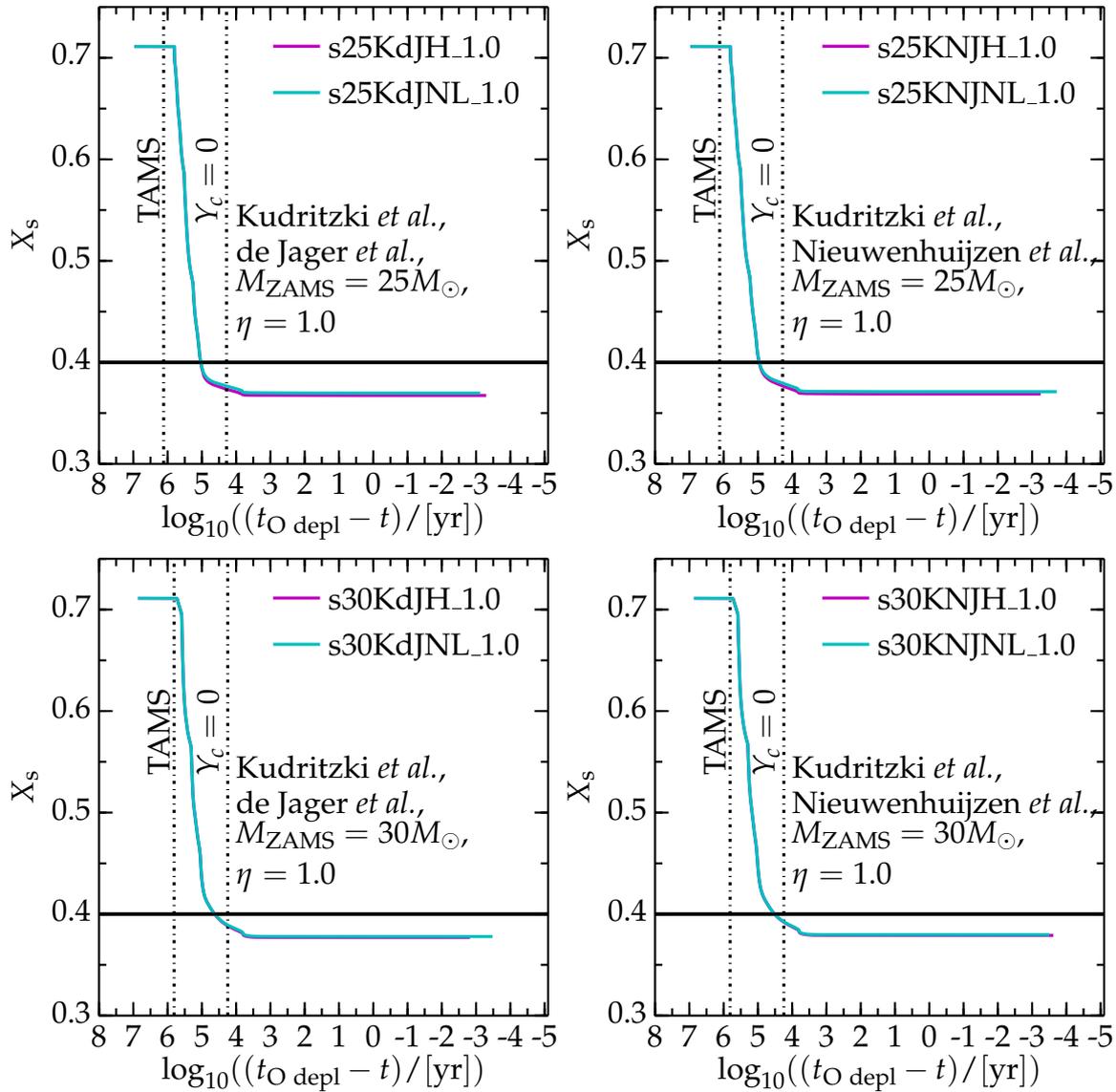


Figure 3.19: Surface abundance of hydrogen for the models reaching the threshold for switching to the WR mass loss rate. Cyan curves use the Nugis & Lamers WR mass loss rate, magenta curves use the Hamann *et al.* rate. The top panels are for $M_{\text{ZAMS}} = 25M_{\odot}$, the bottom panels are for $M_{\text{ZAMS}} = 30M_{\odot}$. The left panels use the de Jager *et al.* cool mass loss rate, right panels use the Nieuwenhuijzen *et al.*, while all models shown use the Kudritzki *et al.* hot mass loss rate. Dot-dashed vertical lines indicate the TAMS and the end of helium core burning ($Y_c = 0$). The horizontal black line marks the threshold for switching to the WR mass loss algorithm.

Note that in this scenario the LBV phase is an intermediate stage between RSG and WR stars, which is another weak point of this scenario, see e.g. [78].

Figure 3.19 shows the surface abundance of hydrogen X_s as a function of time for the eight models reaching the WR stage. All models become WR stars during late-helium core burning, and this phase lasts $\lesssim 10^5$ yr, see also Tab. 3.4. This would be consistent with the expected duration for WR stars of $M \lesssim 30M_{\odot}$ inferred from population statistics, e.g. [90]. However, the phase during which mass loss is on with full efficiency is shorter, since MESA progressively decreases the mass loss rate when $T_c \geq 10^9$ K.

Table 3.4: Radius ($R_{\text{O depl}}$), total mass ($M_{\text{O depl}}$), He core mass (M_{He}), CO core mass (M_{CO}) and mass lost during the WR phase (ΔM_{WR}) in solar units, compactness parameter at oxygen depletion, and duration (Δt_{WR}) of the WR phase for all the models reaching this stage. The first column indicates the mass loss algorithm combination used, see Tab. 2.1 and Tab. 2.2 for details. The results of the first five columns overlap with those highlighted in Tab. 3.2.

$M_{\text{ZAMS}} = 25 M_{\odot}, \eta = 1.0$							
Wind Scheme	R [R_{\odot}]	$M_{\text{O depl}}$ [M_{\odot}]	M_{He} [M_{\odot}]	M_{CO} [M_{\odot}]	$\zeta_{2.5}^{\text{O depl}}$	Δt_{WR} [Myr]	ΔM_{WR} [M_{\odot}]
KdJNL-WR	689	14.25	10.32	6.99	0.220	0.11	0.66
KdJH-WR	632	13.43	10.19	7.19	0.236	0.11	1.48
KNJH-WR	652	13.69	10.18	7.13	0.202	0.09	1.24
KNJNL-WR	695	14.38	10.22	7.24	0.225	0.09	0.55
$M_{\text{ZAMS}} = 30 M_{\odot}, \eta = 1.0$							
Wind Scheme	R [R_{\odot}]	$M_{\text{O depl}}$ [M_{\odot}]	M_{He} [M_{\odot}]	M_{CO} [M_{\odot}]	$\zeta_{2.5}^{\text{O depl}}$	Δt_{WR} [Myr]	ΔM_{WR} [M_{\odot}]
KdJNL-WR	693	16.63	11.39	7.98	0.215	0.04	0.30
KdJH-WR	680	16.25	11.39	8.10	0.188	0.04	0.68
KNJH-WR	691	16.48	11.42	8.01	0.228	0.03	0.53
KNJNL-WR	700	16.78	11.39	8.18	0.210	0.03	0.24

The total amount of mass lost during the WR phase is relatively small (see also Tab. 3.4),

$$\frac{\Delta M_{\text{WR}}}{M_{\text{ZAMS}} - M_{\text{O depl}}} \lesssim 10\% , \quad (3.5)$$

and it is smaller for the $M_{\text{ZAMS}} = 30M_{\odot}$ models since they evolve faster. During the WR phase, the amount of mass shed with the Hamann *et al.* rate is higher by a factor between 2 to 3 than with the Nugis & Lamers rate. Since the WR models presented here have $\log_{10}(L/L_{\odot}) > 4.5$ during their entire WR evolution, the Hamann *et al.* rate has a steeper dependence on L than the Nugis & Lamers rate (see Tab. 2.2), which explains the higher rate and total mass loss using the Hamann *et al.* algorithm. Models computed with the Hamann *et al.* rate have slightly smaller photospheric radii at oxygen depletion. However, as mentioned in §1.4.7 (see also [74]), the concept of a photosphere is problematic for the optically thick atmospheres and winds of WR stars: the optically thick surface that is observed is not in hydrostatic equilibrium. In this light, the small difference in radius found in the results of these simulation is likely not very significant, but it is a result of the use of the different WR wind algorithms.

Table 3.2 shows some general trends when comparing models reaching the WR algorithmic threshold and models with the same efficiency $\eta = 1.0$ that do not (because of the use of a different wind scheme). The radii at oxygen depletion of WR stars with $M_{\text{ZAMS}} = 25M_{\odot}$ tend to be smaller ($R \lesssim 700R_{\odot}$) than non-WR stars at the same stage ($R \gtrsim 750R_{\odot}$). This is not true for $M_{\text{ZAMS}} = 30M_{\odot}$ models: WR stars of this initial mass have radii $R \lesssim 700R_{\odot}$, but the non-WR models can be smaller (even by $\sim 100R_{\odot}$). The differences in final masses $M_{\text{O depl}}$, and helium and carbon-oxygen core masses are not particularly pronounced, some of the non-WR models can be less massive than WR models (e.g. s25VvLNL.1.0, which, however, is very close to the WR threshold, see above), or end with a larger helium (e.g. s25KvLNL.1.0) or carbon-oxygen (e.g. s25VNJNL.1.0) core. This illustrates the importance of chemical mixing up to the surface to reach the WR algorithmic threshold: while mass loss reveals deeper layers, these are sufficiently hydrogen depleted to switch to the WR wind

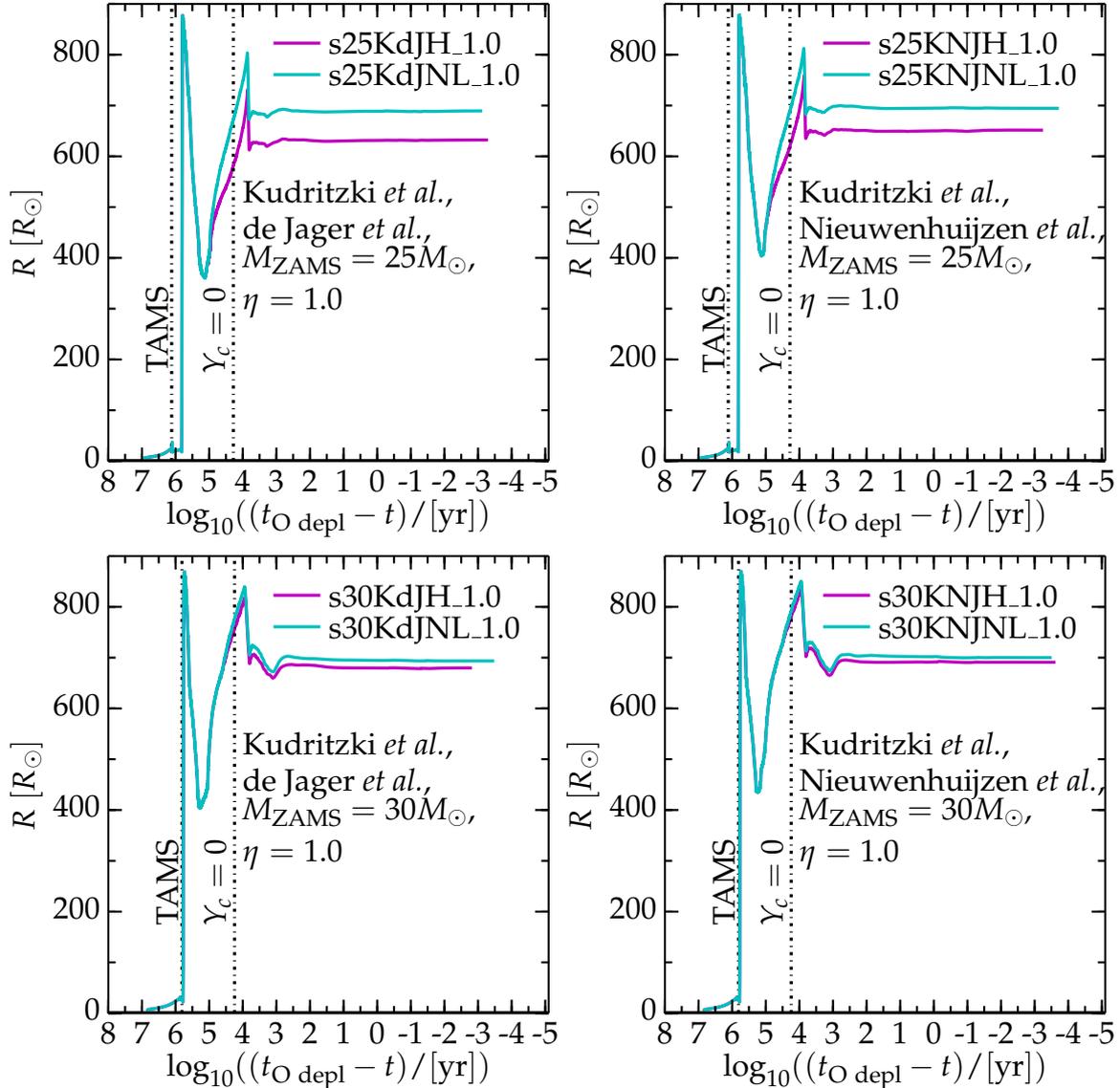


Figure 3.20: Radius for the WR models. Cyan curves use the Nugis & Lamers WR mass loss rate, magenta curves use the Hamann *et al.* rate. The top panels are for $M_{\text{ZAMS}} = 25M_{\odot}$, the bottom panels are for $M_{\text{ZAMS}} = 30M_{\odot}$. The left panels use the de Jager *et al.* cool mass loss rate, right panels use the Nieuwenhuijzen *et al.*, while all models shown use the Kudritzki *et al.* hot mass loss rate. Dot-dashed vertical lines indicate the TAMS and the end of helium core burning ($Y_c = 0$). Note the radius decrease during core helium burning, and the subsequent re-expansion.

scheme only if sufficient helium (and, to a lesser extent, metals) is brought up by convection (and/or other mixing processes), i.e. if $Y_s + Z_s \geq 0.6$.

Figure 3.20 shows the evolution of the photospheric radius (evaluated by MESA) in time. During core helium burning, the radius decreases, but it never reaches the typical radius of WR stars. At the end of core helium burning the radius inflates again, reaching a maximum of $\sim 800R_{\odot}$, and then settles to its final value. Fig. 3.20 also shows that, if the Kudritzki *et al.* wind scheme is used during the hot phase, the final radii when using the Hamann *et al.* and the Nugis & Lamers rates are different by about $\sim 12\%$. In particular, models computed with

the Hamann *et al.* rate are slightly more compact. Note that the final radii of the simulated WR ($\sim 700R_{\odot}$) stars are much larger than those usually cited for WR (e.g. [101]), which are more than an order of magnitude smaller. The presence of strong winds, with a very strong stratification complicates the definition of the stellar radius in WR stars, [90]. The optically thick surface of these stars is likely not in hydrostatic equilibrium. MESA does not compute the spectral distribution of the luminosity, and the radius is computed using the bolometric luminosity. However, such a striking discrepancy remains perplexing and casts even more doubt on the computational definition of WR stars. All WR models presented here barely reach the hydrogen depletion required for the switching to the WR wind scheme, and it is questionable whether they would be observationally classified as WR, but, in the modeling context, the differences presented here between the Hamann *et al.* and the Nugis & Lamers rate are relevant results.

The possibility of “envelope-inflation” for stars approaching the (classical) Eddington limit,

$$L \simeq L_{\text{Edd}}^{\text{cl}} \stackrel{\text{def}}{=} \frac{L_{\text{rad}}\kappa_e}{4\pi cGM} , \quad (3.6)$$

where κ_e is the opacity from electron scattering (instead of the local opacity $\kappa(r)$ used in Eq. 1.14, where I defined the “modified” Eddington luminosity), is an active research topic (see, e.g., [101] and references therein). However, even this envelope inflation can hardly reconcile the stars obtained in this grid with what would be called a WR stars from observations.

CHAPTER 4

Results: Simplified Envelope Shedding Mass Loss

It's better to burn out, than to fade away

[N. Young, "Hey Hey, My My (Into the Black)", Rust Never Sleeps]

4.1 Introduction

In this chapter, I present the results of the simulation of the evolution of a solar metallicity, $M_{\text{ZAMS}} = 15M_{\odot}$ star, with varying levels of mass stripping artificially activated at different evolutionary stages. These are an attempt to explore the effects on the history and final structure of a massive star caused by an impulsive, brief, dynamical mass loss event, that could remove part or all of the hydrogen rich envelope. These events might happen on shorter timescales than the thermal and/or dynamical timescale of the star, resulting in a non-thermal equilibrium and/or non static readjustment of the structure. The procedure (see §2.4) bypasses the limitations of presently available stellar evolution codes that cannot properly account for these events. Possible theoretical examples (pulsational instabilities, RLOF, LBVs eruptions¹, etc., see e.g. [1, 7, 50]) and observational evidence of envelope shedding events are available (see also §1.4.9). The set up for these simulations, stripping process, and choices for when to strip away several portions of the envelope are described thoroughly in §2.4, along with their advantages and shortcomings.

4.2 Stellar Structures after the Stripping Process

Figure 4.1 shows the evolutionary tracks of the stripped models on the HR diagram. The evolution follows the curve of the unstripped reference model (blue curve) until the point of stripping, and then jumps to the point from where the curve restarts with a different color. Each color corresponds to a different amount of mass removed from the envelope (indicated in units of M_{\odot} by the number in the model name). The tracks for the pseudo-evolution during which the stripping occurs are not plotted. Note that I remove the entire convective portion of the envelope (and nothing more) in the model hMR.5 M_{\odot} . The amount of mass lost in the scenario of an unstable RLOF event triggered at half maximum radius is exactly 5 M_{\odot} .

¹These, however, are not addressed in this work, since the M_{ZAMS} of the reference model is too small to reach the LBV instability region of the HR diagram.

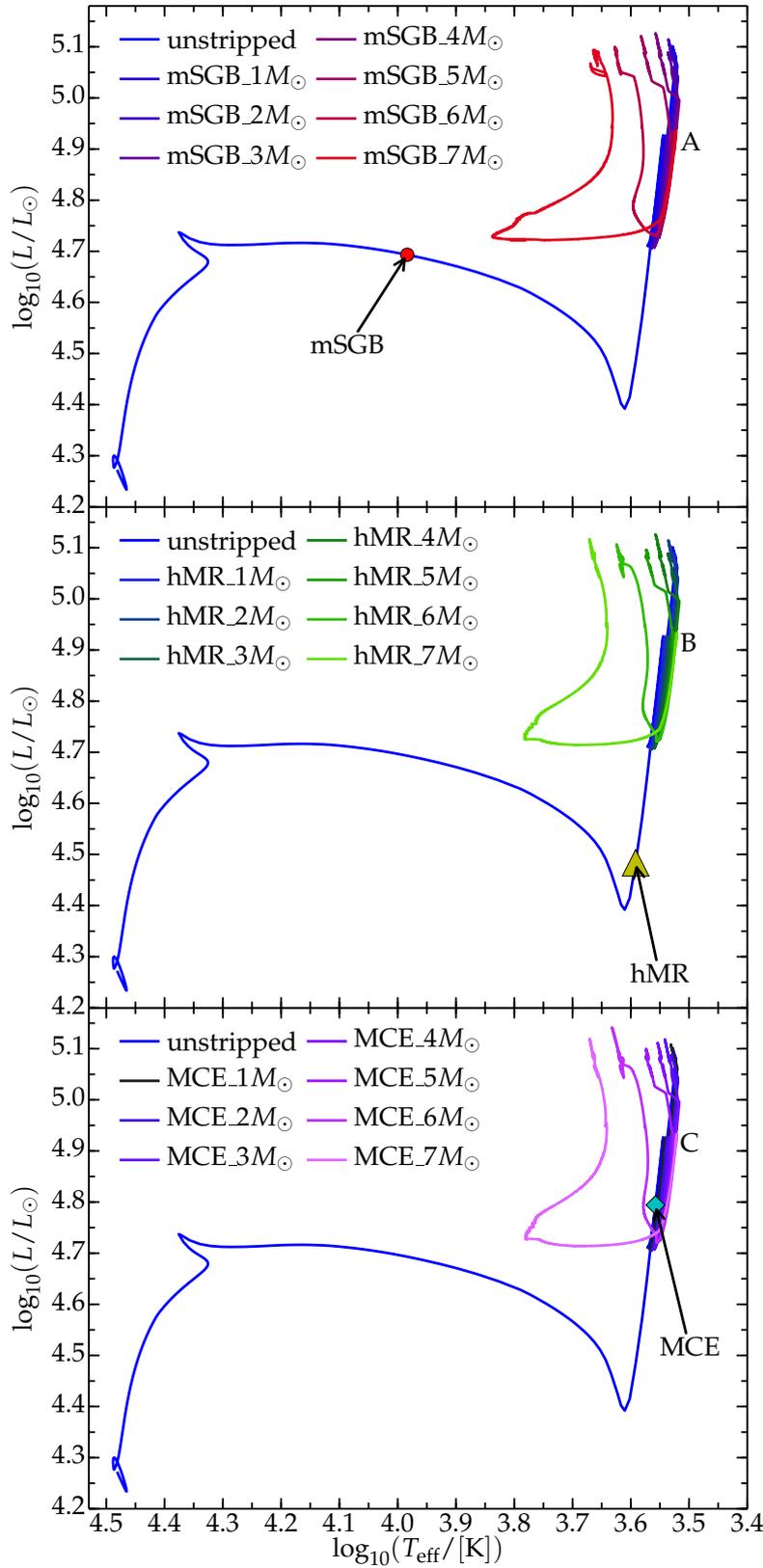


Figure 4.1: HR diagrams for the models stripped at $T_{\text{eff}} = 10^4$ K (mSGB series, top panel), $R \gtrsim 375R_{\odot}$ (hMR series, central panel), and at maximum extent of the convective envelope (MCE series, bottom panel). The point of stripping is indicated by symbols in each panel. The point from which the regular evolution resumes are marked by the letters **A**, **B**, **C** in each panel. From these points, the stripped models initially move downward, and then up again, either at roughly the same T_{eff} or higher, depending on the amount of mass stripped.

Note that using a coarser spatial resolution for these models eliminates the blue loop seen in the $15M_{\odot}$ models using the Vink *et al.* and de Jager *et al.* rates with $\eta = 1.0$. Such blue loops would be expected for the unstripped reference model, cf. Fig. 2.9. Using a coarser resolution saves some computational time, since the fine details of the models presented here are expected to be inaccurate, because of the crude approximations for the stripping procedure.

All stripped models jump from the stripping point to the cool side of the HR diagram, roughly all to the same $\log_{10}(T_{\text{eff}}/[\text{K}]) \sim 3.55$ and $\log_{10}(L/L_{\odot}) \sim 4.9$ (see Tab. 4.1). Tab. 4.2 shows the relative variation of T_{eff} , L , and of the net energy release from hydrogen burning,

$$W_{\text{H}} \stackrel{\text{def}}{=} \int_0^{M_{\text{tot}}} (\varepsilon_{\text{PP}} + \varepsilon_{\text{CNO}}) dm , \quad (4.1)$$

during the stripping process. The relative variations of W_{H} are always smaller than those of L . The relative variations of T_{eff} are small for the hMR and MCE series, since their stripping points are already on the red side of the HR diagram, i.e. a significant portion of the envelope is convective and the evolutionary track is almost as near as it can be to the Hayashi track for fully convective models (see also §2.4 for a description of the stellar structure at each stripping point).

The reason for the differences in the relative variation of L and W_{H} is that ε_{nuc} is a function of the thermal state of the burning region, $\varepsilon_{\text{nuc}} \equiv \varepsilon_{\text{nuc}}(\rho, T, \mu)$, as does W_{H} . Stripping is very fast² compared to the nuclear timescale, therefore μ does not vary significantly during the episodic mass loss event. T and ρ in the hydrogen shell readjust on the *local* thermal timescale of the shell, which does not depend on the total mass of the star. Therefore, the mass removal influences the hydrogen shell only weakly, and the envelope expands (see Tab. 4.1) until the energy flux from the shell can be carried outward. Note that the envelope expansion strongly decreases the outer density, and, therefore, it is easier for photons to stream out. This enforces an increase of the radiative losses, that can be seen in Tab. 4.2 as an increase of L much larger than the (relatively small) increase of ε_{nuc} . When the regular evolution resumes after the stripping, L decreases again, indicating that the radiative losses during the stripping process are too high compared to the power from the nuclear burning happening in the shell. The post-stripping T_{eff} , and L are similar for all models because the stripping points are very close in time (the difference is of $\mathcal{O}(10^4)$ yr) and the shell energy generation rates at each stripping point are very similar.

As mentioned in §2.4, regular hydrostatic stellar evolution works exactly in the opposite way: *the energy loss from the surface determines the energy generation rate in the inner regions* (burning shells and core), while here the energy generation rate cannot vary much (cf. Tab. 4.2) during the mass loss event, and the outer portion of the star must readjust to it. This is because the stripping process mimics events that are *not* hydrostatic and thermal equilibrium processes. Moreover, the reader must keep in mind that in these simulated events, the total amount of mass removed (or, equivalently, the new desired mass) is artificially imposed, while in reality it may depend on the structure of the inner regions of the star.

Figure 4.1 shows that even if the stars become colder than the reference model just after stripping, they all evolve toward the hot side of the diagram afterward. The most stripped models reach $\log_{10}(T_{\text{eff}}/[\text{K}]) \gtrsim 3.8$ before moving again slightly to the cool side. All stripped

²During the stripping process time is held fixed – but in order to find a solution to the stellar structure equations, all the quantities are allowed to change, see §2.4.

Table 4.1: Total mass (M), hydrogen-rich mass (M_{H}), radius (R), luminosity (L), effective temperature (T_{eff}), surface abundance of hydrogen (X_{s}) and hydrogen burning power (W_{H}) in solar units for the stripped models right after the stripping (i.e. at the first iteration outside the pseudo-evolution loop), and for the reference model at the stripping point.

Model	M [M_{\odot}]	M_{H} [M_{\odot}]	R [R_{\odot}]	L [L_{\odot}]	T_{eff} [K]	X_{s}	W_{H} [L_{\odot}]
unstripped_mSGB	14.48	10.67	80	49343	9632	0.71	58782
mSGB_1 M_{\odot}	13.48	9.61	746	76089	3512	0.68	57875
mSGB_2 M_{\odot}	12.48	8.61	772	77824	3473	0.68	59396
mSGB_3 M_{\odot}	11.48	7.61	793	78958	3438	0.67	60446
mSGB_4 M_{\odot}	10.48	6.61	817	80299	3401	0.67	62519
mSGB_5 M_{\odot}	9.48	5.67	796	76210	3401	0.66	59774
mSGB_6 M_{\odot}	8.48	4.67	819	77584	3369	0.66	60304
mSGB_7 M_{\odot}	7.48	3.67	835	78882	3349	0.65	61303
unstripped_hMR	14.48	10.61	382	30447	3906	0.71	50994
hMR_1 M_{\odot}	13.48	9.61	746	76089	3512	0.68	57875
hMR_2 M_{\odot}	12.48	8.61	772	77824	3473	0.68	59396
hMR_3 M_{\odot}	11.48	7.61	793	78958	3438	0.67	60446
hMR_4 M_{\odot}	10.48	6.61	817	80299	3401	0.67	62519
hMR_5 M_{\odot}	9.48	5.61	835	80747	3370	0.67	63437
hMR_6 M_{\odot}	8.48	4.61	849	81012	3344	0.67	64148
hMR_7 M_{\odot}	7.48	3.61	856	81082	3333	0.67	64480
unstripped_MCE	14.48	10.60	638	62253	3612	0.71	51306
MCE_1 M_{\odot}	13.48	9.60	755	77322	3506	0.68	58531
MCE_2 M_{\odot}	12.48	8.60	776	78384	3470	0.68	59869
MCE_3 M_{\odot}	11.48	7.60	797	79464	3435	0.67	60971
MCE_4 M_{\odot}	10.48	6.60	817	80274	3401	0.67	63087
MCE_5 M_{\odot}	9.48	5.60	836	80925	3369	0.67	63771
MCE_6 M_{\odot}	8.48	4.60	851	81234	3342	0.67	64513
MCE_7 M_{\odot}	7.48	3.60	858	81247	3329	0.67	64918

models are hotter than the reference model on reaching the pre-SN stage. In other words, the (simplistic) introduction of an extreme mass loss event in the otherwise regular evolution of the reference model populates the hot and bright portion of the HR diagram.

To explain the physical motivation for the post-stripping blueward evolution, I ran one of the models stripped when the convective envelope reaches its maximum extent (MCE_7 M_{\odot}), but this time without any wind mass loss after the stripping (i.e. $\eta = 0$). The evolutionary tracks on the HR diagram for the two models stripped by 7 M_{\odot} at MCE (with and without the post-stripping wind) are shown in Fig. 4.2, together with the reference model. The model without post-stripping wind mass loss remains cooler than model MCE_7 M_{\odot} , but, is hotter than the reference model when reaches the onset of core collapse. However, its temperature rises above the temperature of the reference model only during very late stages of the evolution. Therefore, the episodic mass loss event itself is not causing the blueward evolution of the stripped models; the combination of the episodic mass loss event plus the post-stripping wind mass loss drives it. The stripping event actually provokes an expansion of the envelope (during the pseudo-evolution, not shown, cf. Tab. 4.1), and the stellar

Table 4.2: Relative variations of the hydrogen burning power (W_{H}), effective temperature (T_{eff}) and luminosity (L) during the stripping process. The relative variations are $\Delta x = \frac{|x^{\text{pre stripping}} - x^{\text{post stripping}}|}{x^{\text{pre stripping}}}$, for $x = W_{\text{H}}, T_{\text{eff}},$ and L , respectively.

Model	$\Delta W_{\text{H}}^{\text{mSGB}}$	$\Delta T_{\text{eff}}^{\text{mSGB}}$	ΔL^{mSGB}
mSGB_1M $_{\odot}$	1.54%	63.54%	54.20%
mSGB_2M $_{\odot}$	1.04%	63.94%	57.72%
mSGB_3M $_{\odot}$	2.83%	64.31%	60.02%
mSGB_4M $_{\odot}$	6.36%	64.69%	62.74%
mSGB_5M $_{\odot}$	1.69%	64.69%	54.45%
mSGB_6M $_{\odot}$	2.59%	65.02%	57.23%
mSGB_7M $_{\odot}$	4.29%	65.23%	59.86%
Model	$\Delta W_{\text{H}}^{\text{hMR}}$	$\Delta T_{\text{eff}}^{\text{hMR}}$	ΔL^{hMR}
hMR_1M $_{\odot}$	11.71%	10.09%	149.91%
hMR_2M $_{\odot}$	14.29%	11.09%	155.60%
hMR_3M $_{\odot}$	16.08%	11.98%	159.33%
hMR_4M $_{\odot}$	19.61%	12.93%	163.73%
hMR_5M $_{\odot}$	21.17%	13.72%	165.21%
hMR_6M $_{\odot}$	22.38%	14.39%	166.08%
hMR_7M $_{\odot}$	22.94%	14.67%	166.31%
Model	$\Delta W_{\text{H}}^{\text{MCE}}$	$\Delta T_{\text{eff}}^{\text{MCE}}$	ΔL^{MCE}
MCE_1M $_{\odot}$	12.29%	2.93%	24.21%
MCE_2M $_{\odot}$	14.57%	3.93%	25.91%
MCE_3M $_{\odot}$	16.44%	4.90%	27.65%
MCE_4M $_{\odot}$	20.04%	5.84%	28.95%
MCE_5M $_{\odot}$	21.21%	6.73%	29.99%
MCE_6M $_{\odot}$	22.47%	7.48%	30.49%
MCE_7M $_{\odot}$	23.16%	7.83%	30.51%

envelope reaches low T_{eff} and high L . The wind algorithm used after the episodic event is the de Jager *et al.* rate, so after stripping, the wind is suddenly much stronger than before (L increases, and T_{eff} is either roughly the same or lower after the stripping), and it removes more mass (compare M_{stripped} in Tab. 4.1 and $M_{\text{pre-SN}}$ in Tab. 4.3), unveiling the inner (and thus hotter) regions of the star, and (partially) driving the blueward evolution. The impulsive mass loss significantly reduces $g \propto M/R^2$ and thus strongly enhances the wind mass loss³.

The post-stripping surface abundance of hydrogen X_{s} is another quantity of interest. Before the stripping event, the chemical composition in the outermost region of the star is homogeneous. Since the mass is removed from the surface and never from as deep as the hydrogen burning shell, the chemical composition of the surface is almost unaltered by internal mixing during the stripping, and a significant amount of hydrogen is left ($X_{\text{s}} \sim 0.7$, see Tab. 4.1). Small variations of X_{s} can be caused by the formation of deep convective layers in the region above the hydrogen shell, where the chemical composition gradually changes in a series of steps (cf. Fig. 3.13). Note that, until the stripping event, the reference model of

³However, the mass loss rate used does not depend explicitly on g .

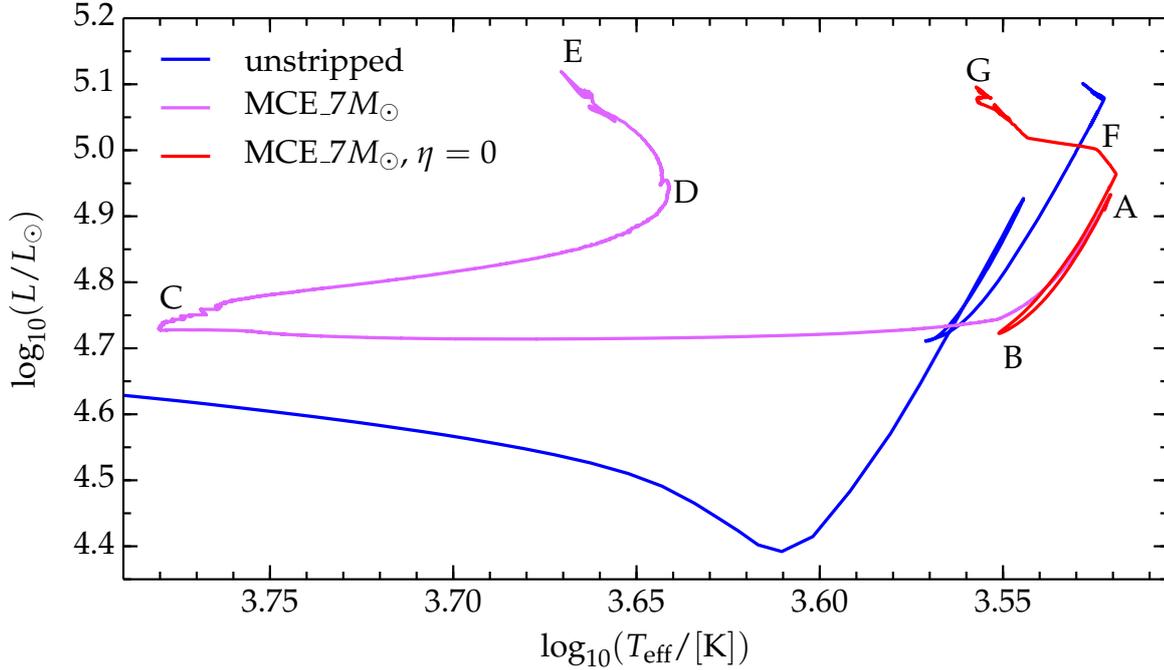


Figure 4.2: Evolutionary tracks on the HR diagram for the unstripped reference model (blue curve), model MCE_7M_{\odot} (pink curve), and the same stripped model re-run without the wind mass loss (i.e. $\eta = 0$, red curve) after the stripping. Both stripped models start from point **A**, that is the arrival point of the jump caused by the stripping. Initially both evolve toward point **B**, i.e. toward lower luminosity. The pink curve (with wind mass loss), then, proceeds toward **C** moving at constant luminosity toward higher T_{eff} , before increasing L and decreasing again T_{eff} to reach **D** and ending its evolution in **E**. In contrast, the red curve (without wind mass loss) moves from **B** to **F** increasing again L at roughly constant T_{eff} , and reach the onset of collapse in **G**. Summarizing, the stripped model with the post-stripping wind (pink curve) follows the evolutionary path **ABCDE**, while the model without post stripping wind follows the path **ABFG**.

the stripped series is identical to s15VdJNL.1.0 of the wind comparison study, whose Y profile at TAMS and at the end of the hot evolutionary phase is plotted in Fig. 3.13. Therefore, the mass loss events simulated here do not directly produce WR stars – which is not the aim of the study – nor do they completely remove the hydrogen envelope of the star. To obtain stars without hydrogen requires additional wind mass loss.

To summarize, the picture for the formation of completely hydrogen-free pre-SN structures investigated in this work is the following: an impulsive envelope-shedding event removes the bulk (in terms of mass) of the envelope, and then the regular line-driven wind finishes the job and (possibly) removes the surface hydrogen. This is not a result, but rather the *hypothesis* adopted for these simulations, which are strongly limited both by hydrostatic requirement of MESA and by our poor knowledge of the physics of these events.

Table 4.3: Final mass $M_{\text{pre-SN}}$, hydrogen-rich mass, helium, carbon/oxygen and iron core masses M_{H} , M_{CO} , and M_{Fe} , respectively, in units of M_{\odot} ; compactness $\zeta_{2.5}^{\text{pre-SN}}$, radius R , luminosity L , in solar units, T_{eff} in units of 10^3 K at the onset of core collapse. The helium (carbon/oxygen) core boundary is defined by the location where the hydrogen (helium) abundance as a function of mass is the steepest. The iron core boundary is the first location going inward where $Y_e < 0.49$, see [102].

Model	$M_{\text{pre-SN}}$ [M_{\odot}]	M_{H} [M_{\odot}]	M_{He} [M_{\odot}]	M_{CO} [M_{\odot}]	M_{Fe} [M_{\odot}]	$\zeta_{2.5}^{\text{pre-SN}}$	R [R_{\odot}]	L [L_{\odot}]	T_{eff} [10^3 K]
unstripped	12.28	7.18	5.10	3.27	1.51	0.103	1039	120309	3.337
mSGB_1 M_{\odot}	11.27	6.18	5.09	3.28	1.49	0.125	1031	121084	3.355
mSGB_2 M_{\odot}	10.25	5.16	5.09	3.26	1.49	0.142	1013	119370	3.373
mSGB_3 M_{\odot}	9.17	4.06	5.11	3.27	1.49	0.127	991	121536	3.425
mSGB_4 M_{\odot}	7.87	2.67	5.20	3.32	1.58	0.138	932	122270	3.537
mSGB_5 M_{\odot}	6.82	1.61	5.21	3.33	1.56	0.171	828	122984	3.759
mSGB_6 M_{\odot}	5.94	0.74	5.20	3.32	1.54	0.114	663	123258	4.204
mSGB_7 M_{\odot}	5.59	0.38	5.21	3.33	1.50	0.089	555	118763	4.553
hMR_1 M_{\odot}	11.27	6.18	5.09	3.28	1.49	0.125	1031	121084	3.355
hMR_2 M_{\odot}	10.25	5.16	5.09	3.26	1.49	0.142	1013	119370	3.373
hMR_3 M_{\odot}	9.17	4.06	5.11	3.27	1.49	0.127	991	121536	3.425
hMR_4 M_{\odot}	7.87	2.67	5.20	3.32	1.58	0.138	932	122270	3.537
hMR_5 M_{\odot}	6.87	1.68	5.19	3.32	1.53	0.118	843	122179	3.719
hMR_6M	5.96	0.77	5.18	3.31	1.58	0.122	676	122065	4.153
hMR_7M	5.52	0.32	5.21	3.30	1.60	0.110	551	122645	4.604
MCE_1 M_{\odot}	11.27	6.17	5.10	3.27	1.58	0.102	1032	118857	3.339
MCE_2 M_{\odot}	10.25	5.16	5.09	3.27	1.54	0.134	1016	120982	3.379
MCE_3 M_{\odot}	9.17	4.06	5.11	3.27	1.53	0.159	989	119197	3.413
MCE_4 M_{\odot}	7.88	2.69	5.19	3.32	1.58	0.130	932	122808	3.541
MCE_5 M_{\odot}	6.87	1.68	5.19	3.32	1.52	0.131	843	121709	3.715
MCE_6 M_{\odot}	5.96	0.78	5.18	3.31	1.55	0.153	675	122791	4.162
MCE_7 M_{\odot}	5.52	0.31	5.21	3.31	1.56	0.123	552	122631	4.602

4.3 Stripped Models at the Onset of Core Collapse

Table 4.3 summarizes some properties of the stripped models at the onset of core collapse (see Eq. 1.9). These have been used as initial condition for the simulation of light curves from stripped pre-SN stars in Morozova *et al.*, submitted to ApJ, arXiv:1505.06746.

Figure 4.3 shows the density profiles of the stripped models. Not surprisingly, that of the inner $\sim 1M_{\odot}$ does not change when different amounts of mass are stripped at different stages of the evolution. For all of the models, the central density at the onset of core collapse is $\log_{10}(\rho/[g \text{ cm}^{-3}]) \simeq 10$. This can be explained by considering the structure of the core when the envelope is removed (see Tab. 2.3). All the stripping points are during post-hydrogen-core-burning epoch, so either the core is radiative (because it is inert and there is no nuclear burning is below the hydrogen shell), or a radiative region exists between the core and the removed outer envelope. The structure of the core is, thus, determined by the

radiative temperature gradient,

$$\left. \frac{dT}{dr} \right|_{\text{rad}} = -\frac{3}{4acT^3} \frac{\kappa\rho L}{4\pi r^2}, \quad (4.2)$$

or there is a region outside the core where this is true. This gradient does not depend on the pressure at the outer boundary of the core, which is effectively the only thing changing when parts of the envelope are removed (chemical mixing in the envelope is not strong enough to significantly change κ). Thus, there is no significant thermal response of the core to the mass loss event. The presence of a radiative layer “protects” the core from the mass stripping, because all readjustments happens in the radiative layer. Moreover, the iron core of a star at the pre-SN stage is always composed of relativistically degenerate electrons that provide the bulk of the gas pressure, therefore, its EOS is a polytropic of index $\Gamma = 4/3$,

$$P \propto \rho^\Gamma, \quad (4.3)$$

where the proportionality constant depends only on the (Fermi-Dirac) statistical distribution that the electrons obey. There is no reason to expect variations in the structure of the innermost core because of the stripping.

Furthermore, Fig. 4.3 shows that the amount of mass stripped strongly affects the density profile in the mass shell $1.5 \lesssim m/M_\odot \lesssim 3.5$, corresponding roughly to the carbon-oxygen-rich shell. The differences between models stripped by the same amount but at different times are less pronounced. The differences in the density profiles arise during the post-stripping evolution. The amount of mass that the shell(s) must support during the post-stripping evolution is determined by the amount of mass removed in the stripping process. Therefore the energy generation rate in shells required to sustain the remaining envelope, the density structure, and final chemical composition of these shells, depends on the amount of mass removed during the stripping event.

Interestingly, the mass interval where the models vary the most is also where the compactness parameter $\xi_{2.5}$ (see Eq. 3.1) is determined. Whether a star successfully explodes as a result of core collapse, and (perhaps) whether a NS or BH is formed, is sensitive to the value of $\xi_{2.5}$ [16, 36, 89]. The removal of the hydrogen-rich envelope by some mechanism during the life time of a massive star could influence the outcome of core collapse.

Finally, the structure of the partially stripped hydrogen-envelope at the time of core collapse is *insensitive* to when the stripping occurred. As can be seen in Fig. 4.3, stars for which the same amount of mass was removed have nearly identical density profiles in their outer regions, irrespective of when the stripping occurs. Furthermore, the outermost (i.e. at larger mass coordinate) density, $\rho_{\text{surf}} \simeq 10^{-9} \text{ g cm}^{-3}$, and the helium core mass, $M_{\text{He}} \simeq 5.1M_\odot$, are similar for all the models (see Tab. 4.3). The density gradient in the envelope outside the helium core is steeper for models with less massive envelopes (i.e. the most stripped models).

Note that none of the stripped models reach the onset of core-collapse as WR stars. This is because of the choice of the reference model M_{ZAMS} . The wind, for $M_{\text{tot}} \lesssim 15M_\odot$, is not strong enough to remove the remaining layer of hydrogen left after stripping. However, the aim of this study is not to produce WR pre-SN structures, but rather to show that it is possible to explore the effects on stellar evolution of events that strip a large fraction of the hydrogen-rich envelope, even without a detailed, dynamical and self-consistent treatment of the mass loss event itself. The procedure, described in §2.4.1, allows one to evolve the

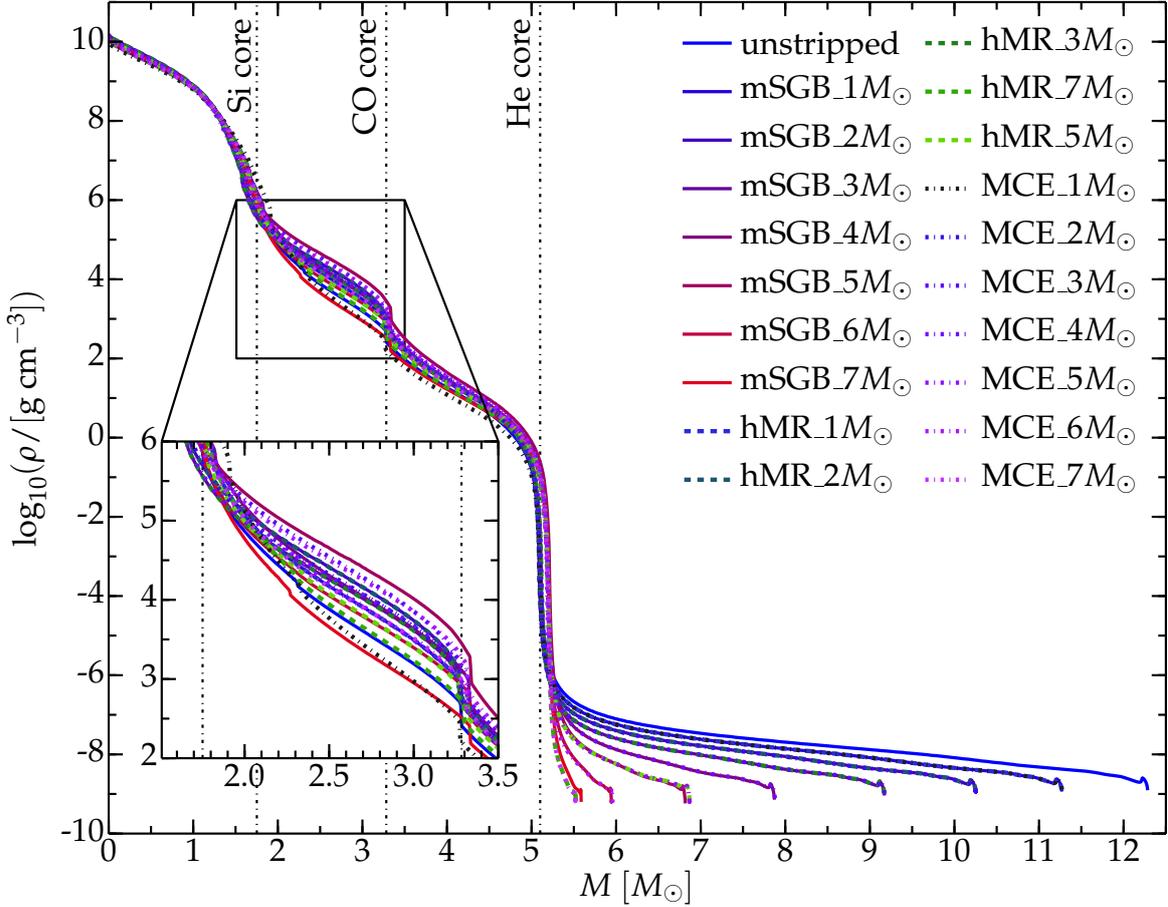


Figure 4.3: Density profiles of the stripped models at the pre-SN stage. Solid, dashed and dot dashed curves represent the mSGB, hMR and MCE series respectively. The zoomed panel focuses on the region where the most significant differences between the density profiles are, i.e. $1.5M_{\odot} \lesssim m \lesssim 3.5M_{\odot}$ and $2 \lesssim \log_{10}(\rho/[g\text{ cm}^{-3}]) \lesssim 6$. The vertical dot dashed lines mark the edges of the (going inward) helium, carbon-oxygen and silicon core (the latter is defined in analogy with Eq. 3.2, substituting the abundance of hydrogen X with that of oxygen, $X_{16\text{O}}$). The iron core mass, whose edge is defined where $Y_e < 0.495$, [102], is $M_{\text{Fe}} = 1.64M_{\odot}$.

star with the same parameters before and after the mass stripping event, resulting in a more realistic simulation than an arbitrary increase of the wind efficiency η .

Therefore, the formation of WR pre-SN structures, i.e. the progenitors of type Ib/Ic core-collapse SNe, could be explored with this procedure with a (initially) more massive reference model⁴.

⁴However, for $M_{\text{ZAMS}} \gtrsim 20M_{\odot}$ the treatment of near-to-super-Eddington radiation dominated envelopes may be crucial, see also §1.3.1 and §2.1.3.

CHAPTER 5

Discussion and Conclusion

Non esiste un piano che possa prevedere tutto. Altri solleveranno il capo, altri diserteranno. Il tempo non cesserà di elargire sconfitte e vittorie a chi proseguirà la lotta. [...] Possano i giorni trascorrere senza meta. Non si prosegua l'azione secondo un piano.
[L. Blisset, Q]

5.1 Context Summary

Computational studies in astrophysics require a large number of free parameters and simplified – perhaps too much so – algorithms to represent specific phenomena occurring in cosmic bodies. This reflects the intrinsic difficulties of their physical and self-consistent modeling. In many cases, as studies of stellar evolution, the sensitivity to variations of the unconstrained parameters, or to the use of different algorithms is not explored.

This thesis presents an extensive analysis of the treatment of mass loss in numerical simulations of massive star evolution using the open-source stellar evolution code MESA. It aims at understanding systematic differences caused by the different algorithms available in the literature. Mass loss remains the most uncertain element in the theoretical description of the evolution of single¹ massive stars [7], together with the uncertainties in the treatment of convection and mixing that are present in the numerical modeling of stars of any initial mass.

Massive stars, have three broad mass loss channels available in nature: (i) line (or dust) driven stellar winds; (ii) impulsive events capable of removing a large fraction of the mass in a very short time; and (iii) Roche lobe overflow (RLOF) in binary systems. Any mass loss event happening on a timescale shorter than the thermal timescale or the dynamical timescale is “impulsive”, i.e. the re-adjustment of the stellar structure cannot maintain the thermal and hydrostatic equilibrium.

The high non-linearity of the dynamical equations of wind-like flows is an obstacle to the finding of a self-consistent solution. A number of observational and theoretical clues points to the existence of envelope shedding, impulsive, dynamical events (such as pulsational instabilities, catastrophic eruptions, etc.). However, in stellar evolution modeling, little work has yet been done to explore these phenomena. This is because of the technical and/or computational difficulties (e.g. the complex geometry of these events) and our poor knowledge of the physics underlying these envelope shedding events.

¹i.e. that do not interact with the possibly present companion(s).

This study is divided into two separate parts. One deals with the wind mass loss, comparing and contrasting the effects of different wind models on the stellar evolution. The other part attempts a simplified, but astrophysically motivated, *numerical experiment* to simulate the effects of a very short timescale stripping of large fractions of the stellar envelope.

5.2 Wind Mass Loss

Mass loss is usually included in models using parametric algorithms. These are either semi-empirical, or theoretical predictions based on stellar wind models. They cannot yet be derived self-consistently from first principles. Most of these algorithms rely on an assumed velocity structure $v(r)$ to obtain the mass loss rate from the continuity equation, $\dot{M} = 4\pi r^2 \rho v$. A breakdown of the assumptions behind this equation (e.g. because of the wind “clumpiness” or non-sphericity) may invalidate the entire framework adopted to derive the mass loss rate.

Virtually all massive stars live in multiple systems, and the effects of binarity on the observed spectra are not well known (e.g. modification of the conditions in the outer portion of the stars because of colliding winds, tidal-induced mixing phenomena, etc.) and this work treats only winds of single stars. However, all the wind mass loss algorithms derived from observations may suffer from systematic errors because of the effects of the companion(s).

Many commonly employed algorithms for massive stars have not been examined in detail regarding understanding and pinpointing in a systematic way the differences in the resulting stellar structures (see e.g. [45]).

The wind mass loss study aims to show and understand these systematic differences. I varied two aspects: the wind mass loss rate algorithms, which provide a mass loss rate as a function of some stellar parameters, $\dot{M} \equiv \dot{M}(L, R, T_{\text{eff}}, Z, \dots)$; and the efficiency factor η with which \dot{M} is scaled.

The choice of the stellar variables used to express \dot{M} has a certain degree of freedom: different algorithms chose different variables, and these may often not be independent (e.g. M and L) and/or they do not form a complete set that characterizes uniquely the outflow. Such a complete set may not even exist, since mass loss rate is determined by the micro-physics of process(es) driving the wind, that may not be uniquely determined even if all the macroscopic stellar properties are specified. In other words, there could be a degeneracy that prevents to express the wind mass loss rate determined by micro-physical processes only with macro-physical variables. The dependence $\dot{M} \equiv \dot{M}(L, R, T_{\text{eff}}, Z, \dots)$ is obtained by fitting observed or synthetic data, but there is no theoretical justification derived from first principles to insure that the coefficients obtained with these fit procedures are, or should be, constants. Studies that vary separately each single coefficient in the mass loss rate formulae should be encouraged.

The efficiency factor η is a completely free parameter. Many authors have used $\eta \gg 1$ to enhance mass loss, with the aim of reproducing the total mass lost (including mass lost during non-wind and possibly non-radiatively-driven phenomena), e.g. [46]. Others (e.g. [7, 49] and references therein) suggest using $\eta < 1$, because the commonly adopted wind rates assume homogeneity, that is likely not correct. Observations indicate that winds, instead, have over-dense clumps of matter, that provide the bulk of the opacity, absorb most of the photons and create the spectral features, from which the wind mass loss rate is inferred. The average density of the wind is likely much smaller than the density of these clumps, and this

can produce an overestimation of the mass loss rate by factors between 2 to 10. In principle, if the formation of clumps is very efficient and their opacity is very large, they could cause a larger average mass loss rate. The lack of a physical basis for η must be emphasized.

I studied twelve different algorithms, obtained combining two different mass loss rates for hot stars (Vink *et al.* and Kudritzki *et al.*), three for cool stars (de Jager *et al.*, Nieuwenhuijzen *et al.* and van Loon *et al.*) and two for WR stars (Hamann *et al.* and Nugis & Lamers). I used three efficiencies $\eta = 0.1, 0.33, 1.0$; following the suggestions of [7]. I compared these algorithms in the evolution of non-rotating stars of four different initial masses ($M_{\text{ZAMS}} = \{15, 20, 25, 30\}M_{\odot}$) at solar metallicity. The use of a hydrostatic stellar evolution code to carry out the comparison is justified as long as the characteristic time for mass loss does not exceed the timescale on which thermal and hydrostatic equilibrium are achieved. I carried out extensive work to properly resolve in both space and time these simulations, at least until mass loss is on in the simulations (see also §B.3.3). The details of the setup for these simulations are described in §2.3.

The comparison shows that the differences in the models increases with increasing M_{ZAMS} , and that they are particularly pronounced during the cool and giant phases, i.e. the mass loss rate is less certain when it is the largest. The use of different algorithms yields qualitative differences in the predicted evolutionary tracks, although hard to test with observations because of the large degeneracy with other free parameters involved in the simulation of massive stars (e.g. α_{mlt} , overshooting, etc.), see e.g. §3.2.

More specifically, the Vink *et al.* rate for the hot phase of evolution produces a lower total mass at the end of this phase (i.e. higher average mass loss rate), because of its explicit treatment of the wind driving when iron starts recombining in the envelope. This influences the dimensions of the helium core, and triggers blue-loops in the $M_{\text{ZAMS}} = 15M_{\odot}$ models. The Kudritzki *et al.* rate causes unusual and distinctive “loop-like” features toward the end of the main sequence of $M_{\text{ZAMS}} = 25M_{\odot}$ and $30M_{\odot}$ stars. I discuss these in detail in §3.2.

Two of the cool mass loss rates compared in this study, de Jager *et al.* and Nieuwenhuijzen *et al.* are nearly identical, yet, I found a large difference comparing them to the van Loon *et al.* algorithm. The van Loon *et al.* algorithm yields a much larger mass loss rate, producing stars with a very different outer structure and its mass loss rate can be so high that it drives late blueward evolution (by revealing hotter and deeper layers of the star), that does not happen when using the other cool wind algorithms. This is because the van Loon *et al.* rate assumes the formation of dust in the extended stellar envelopes, which is more effective than metal lines in driving the wind.

Out of the 144 models computed, only eight reach the WR stage: those using the combination of the Kudritzki *et al.* and de Jager *et al.* or Nieuwenhuijzen *et al.* rates with $\eta = 1.0$. The small number of WR stars in this grid does not rule out the wind overestimation: many channels for the formation of WR (other than wind mass loss) exists and may dominate in nature. The highest M_{ZAMS} considered here ($30M_{\odot}$) is still at the lower end of the mass range for the *in scenario* formation of WR stars from the modeling of single isolated progenitors. The Hamann *et al.* mass loss rate is much higher than the Nugis & Lamers rate, for the luminosity compared, but the amount of mass lost during the WR stage is only a few percent of the total mass in each case. Note also that the final radii of the WR models in my grid are extremely large ($R \sim 700R_{\odot}$), however, this is caused by the fact that MESA does not compute the spectral distribution of the luminosity.

Mass loss from stellar winds also influences the structure (e.g. compactness parameter) of the stellar core at the pre-SN stage, thus playing a role in determining the “explodability”

of the star at the end of its life. It creates small variations in the stellar core structure during earlier stages, that then may be amplified by the subsequent evolution. I was not able to disentangle the effects of mass loss from the effects of changing the spatial discretization on the compactness of the core, because of difficulties encountered in obtaining properly resolved core structures in late stages (namely, neon and oxygen burning, i.e. after mass loss is shut down for computational reasons, see also §B.3.3),

Note that different wind mass loss algorithms predict different ejecta mass, stellar radii and effective temperature. Therefore, each simulation produces a different environment for the micro-physical processes determining dust formation and composition, both during the stellar life and at its explosive death. This could in principle provide observational signatures pointing to certain mass loss algorithms or ruling out others. The caveat is the degeneracy of wind signatures with other (poorly constrained) phenomena in the evolution of massive stars. Comparing stellar models computed with different algorithms to observations might also provide insight on when the parameters entering in the mass loss algorithms are not constant as assumed.

The main point to keep in mind is that the commonly used wind algorithms are *not* equivalent to each other, neither from the theoretical point of view because of the different choices of the stellar parameters to express \dot{M} , nor from the stellar evolution point of view because they cause significantly different evolutionary tracks. Although very challenging, the observational signature of different mass loss rates could be detected. Different algorithms produce different conditions in the outer regions of the stars in each evolutionary stage, which in turn determine the characteristics of the circumstellar mass, chemical composition, dust fraction and properties.

5.3 Envelope Shedding Events

Including non-wind mass loss in evolutionary computations is both very challenging (because of technical limitations and uncertainties surrounding the physical scenario) and highly needed. This is especially true if the winds are observationally overestimated; then non-wind mass loss mechanisms are required during the evolution of most of the progenitors of hydrogen-poor core-collapse SNe (e.g. type IIb/Ib/Ic). Among the proposed non-wind phenomena are pulsational instabilities, eruptive events of yet unknown trigger, such as LBVs outflows. Another, perhaps dominant, mechanism is RLOF in binaries, that may become dynamically unstable under certain circumstances. It is possible that impulsive phenomena strip away most but not all the hydrogen-rich envelope, and regular line driven winds remove the remaining portion.

Given the lacunae regarding the physics of any non-wind mass loss event, I used a very simplified approach to perform a sensitivity study by removing (stripping away) several portions of the envelope at different times of the RSG evolution of a $15M_{\odot}$ star. This approach is still strongly limited by the (global) hydrostatic equilibrium condition of the MESA code, and the timing and amount of mass lost in the stripping are specified *by hand*.

I chose three different epochs as the stripping stages (middle of the subgiant branch, when half of maximum radius is reached, and when the convective envelope reaches its maximum extent), to explore the sensitivity of the remaining structure to the moment of the stripping. I chose these three different moments since the envelope structure is different at each one of these, however, they are quite close in time ($\sim 10^4$ years) compared to the stellar

lifetime for the unstripped reference model ($\sim 14 \times 10^6$ years), so no significant difference is found changing the stripping point.

The innermost core is unaffected by the envelope stripping process, while the onion-skin structure of the shells changes, removing different portions of the envelope, but independent of when stripping. The helium core mass and structure is roughly the same for all stripped models (since it is almost determined before the stripping event), and the outermost density value is roughly $\rho \sim 10^9$ g cm³ for all of them. However, the amount of mass between the outer boundary of the helium core and the surface is very different in models stripped by different amounts, therefore the density gradient $d\rho/dm$ depends on the amount of mass removed. The effects on the light curve produced by the explosion of such models are studied in a forthcoming paper (Morozova *et al.*, submitted to ApJ, arXiv:1505.06746).

None of the stripped models is a WR star at the pre-SN stage. This is because of the combination of two reasons: (i) even in the most-stripped models, I left a layer of hydrogen-rich material above the hydrogen burning shell. This layer cannot be removed without violating the hydrostatic equilibrium requirement, since it is the one to re-adjust (on its local thermal timescale) to the mass stripping. (ii) the initial mass is only $15M_{\odot}$, therefore the post-stripping wind is not strong enough to remove the remaining hydrogen-rich mass.

The most important result of this sensitivity study is that *the combination* of impulsive mass loss events with the post-stripping stellar winds populates the hot and luminous portion of the HR diagram. This indicates that impulsive mass loss events might play a role in the solution of the so-called “Red Super Giant problem”, [29].

Although the mass stripping algorithm presented here is *not* physically self-consistent, and cannot capture the details of the envelope shedding event, nor allow its investigation, it is an improved way to simulate stripped pre-SN structures. Previous studies removed mass by enhancing the winds with $\eta \gg 1$, neglecting the fundamental physical differences between radiatively driven winds and impulsive events. The procedure presented here permits to evolve the star with the same set of parameters before and after the stripping, approaching more closely the evolution of real stars.

5.4 Directions for Further Work

The philosophy underlying this study is to show that widely used assumptions (and, therefore, algorithms) in stellar evolution calculations should not be completely trusted, as it is – maybe too often – done. The understanding of the physics in stellar environments (and of all other astrophysical phenomena depending on stellar astrophysics, such as galactic chemical evolution) can progress only if the weak points of the current paradigm are stressed, underlined, and carefully explored and understood. This requires the expansion of the kind of study performed here on the mass loss (and specifically on the wind algorithms) to the entire set of parameters and assumptions commonly made to carry out simulations of stellar evolution. In principle, this could provide larger sets of models to compare to observations and, hopefully, better constrain the uncertainties on the algorithms varied.

For these purposes, MESA is an efficient and easy to customize tool. However, it cannot be used naively: the MESA code rarely crashes, even when the parameters used produce unphysical situations. Quite often MESA finds solutions that do not have any physical meaning, because the settings adopted violate some of the assumptions on which the code (and more in general, stellar evolution theory) is founded. The careful and detailed analysis

of all the aspects of the solution found, including those considered secondary, is mandatory. More details of this aspect are discussed in §B.

The study presented here on the sensitivity of massive star evolution to different mass loss algorithms should be extended to a broader initial mass and metallicity range. The threshold to switch from one rate to another should also be varied, to check whether this significantly influences the results. Ultimately, such threshold should disappear from stellar evolution codes, as they do not correspond to true, physical discontinuities in the evolution of real stars.

On the other hand, a better connection with observations is needed, to use actual data for constraining the mass loss algorithms used and better understand the underlying physics. For this, it would be very helpful to produce synthetic spectra of the simulated stars in each phase of their evolution, to have quantities which may be directly compared with observed stars. The role of binarity deserves better attention in modeling, and its impact on the observation should be better constrained.

The number of free parameters available in stellar evolution codes (e.g. the wind efficiency η) permit the reproduction of a large variety of observed and/or inferred phenomena without improving the knowledge of the basic physical mechanisms involved. An example of this is the possibility of simulating the total amount of mass lost by a massive star during its entire evolution (possibly experiencing impulsive mass loss events) just by using $\eta \gg 1$. This approach should be avoided, and instead more careful and physically sound simulations should be carried out. In fact, a large enhancement of the wind mass loss could in principle reduce the mass loss timescale, and if this drops below the (local) thermal timescale, the assumption of thermal equilibrium used in stellar evolution calculations may break.

The stripping procedure adopted in this work is far from being realistic. However, even without the inclusion of hydrodynamics, and without a physical picture derived from first principles of what could trigger eruptive mass loss events, it allows one to study the change in the stellar structure resulting from an *artificial* short and impulsive mass loss event. Much more work is need in this direction: if the winds cannot remove the entire hydrogen envelope of massive stars, then there is a need to improve the understanding of how stripped pre-SN structures forms in nature. In the very end, this will require the inclusion of multi-dimensional hydrodynamics at least for the envelope shedding phase.

APPENDIX A

MESA parameters files and customized routines

An algorithm must be seen to be believed.

[D. Knuth, The Art of Computer Programming – Fundamental Algorithms]

A.1 The Inlist Parameters Files

To run a simulation with MESA the parameters can be set in multiple `inlist` files, the last loaded overwrites the parameters in common with the previous `inlists`. In a common situation, the directory in which the simulation is run contains the main `inlist`, which can redirect to (multiple) secondary `inlist(s)`, each dealing with a specific subset of parameters.

A.1.1 `inlists` for the Systematic Comparison of Wind Mass Loss

The MESA parameter file used for the systematic comparison of wind algorithms is the same for all runs. First the main `inlist` loads the default parameters¹ for massive stars (`inlist_massive_defaults`), and then it loads the customized set of parameters, overwriting the default in case of conflict. I set `Z_base=0.019` in the `inlist_massive_defaults`. The default set of parameters is listed in the following and it is the same for all runs. Only the **MASS** and **ETA** string must be substituted with the M_{ZAMS} value (in units of M_{\odot}) and the efficiency η wanted.

```
! inlist_project
&star_job

mesa_dir = '' ! empty string defaults to environment variable MESA_DIR
history_columns_file = './history_columns.list' ! if null string, use
    default.
profile_columns_file = './profile_columns.list' ! if null string, use
    default.

!     show_log_description_at_start = .true.
    show_net_species_info = .true.
```

¹These may depend on the MESA release version, the released used for this work is 6794.

```
show_net_reactions_info = .true.

create_pre_main_sequence_model = .false.

change_net = .true.
new_net_name = 'approx21.net'

set_rates_preference = .true. ! for use by net + rates modules
new_rates_preference = 2 ! only used if set_rates_preference is true
! 1 = NACRE rates -- this is the default
! 2 = jina reaclib rates

eos_file_prefix = 'mesa'
kappa_file_prefix = 'gs98'

set_initial_age = .true.
initial_age = 0 ! in years

set_initial_model_number = .true.
initial_model_number = 0

!SOLAR CHEMICAL COMPOSITION
change_initial_Z = .true.
new_Z = 0.019d0
change_initial_Y = .true.
new_Y=0.27

save_photo_when_terminate = .true.
pause_before_terminate = .false.
pgstar_flag = .false.

/ !end of star_job

&controls

!avoid MLT++ oscillations
which_atm_option = 'Eddington_grey'
which_atm_off_table_option = 'grey_and_kap'
Pextra_factor = 1

report_why_dt_limits = .true.
report_all_dt_limits = .true.

! starting specifications
initial_mass = MASS
!solar chemical composition
initial_Z = 0.02
!initial_Y = 0.27
```

```
!Mass Loss
use_other_wind = .true.
RGB_wind_scheme = 'other'
AGB_wind_scheme = 'other'
!hot wind efficiency
x_ctrl(1) = ETA
!cool wind efficiency
x_ctrl(2) = ETA
!WR wind efficiency
x_ctrl(3) = ETA

! when to stop
fe_core_infall_limit = 1d8
!the real stopping criterion is set in the run_star_extras.f

!Resolution
mesh_delta_coeff_for_highT = 1.0
logT_max_for_standard_mesh_delta_coeff = 9.0
logT_min_for_highT_mesh_delta_coeff = 9.5
max_allowed_nz=30000
varcontrol_target = 1d-4
mesh_delta_coeff = 1.0
mesh_max_allowed_ratio=2.5
min_timestep_limit = 1d-12 !in sec

!Mesh refinement (ready to push the evolution pass 0 depletion)

mesh_dlog_pp_dlogP_extra = 0.15
mesh_dlog_cno_dlogP_extra = 0.15
mesh_dlog_3alf_dlogP_extra = 0.15
mesh_dlog_burn_c_dlogP_extra = 0.15
mesh_dlog_burn_n_dlogP_extra = 0.15
mesh_dlog_burn_o_dlogP_extra = 0.15
mesh_dlog_burn_ne_dlogP_extra = 0.15
mesh_dlog_burn_na_dlogP_extra = 0.15
mesh_dlog_burn_mg_dlogP_extra = 0.15
mesh_dlog_cc_dlogP_extra = 0.15
mesh_dlog_co_dlogP_extra = 0.15
mesh_dlog_oo_dlogP_extra = 0.15
mesh_dlog_burn_si_dlogP_extra = 0.15
mesh_dlog_burn_s_dlogP_extra = 0.15
mesh_dlog_burn_ar_dlogP_extra = 0.15
mesh_dlog_burn_ca_dlogP_extra = 0.15
mesh_dlog_burn_ti_dlogP_extra = 0.15
mesh_dlog_burn_cr_dlogP_extra = 0.15
mesh_dlog_burn_fe_dlogP_extra = 0.15
mesh_dlog_pnhe4_dlogP_extra = 0.15
```

```
mesh_dlog_other_dlogP_extra = 0.15
mesh_dlog_photo_dlogP_extra = 0.15

!first 2 functions set in inlist_massive_defaults
xa_function_species(3) = 'c12'
xa_function_weight(3) = 20
xa_function_param(3) = 1d-2

xa_function_species(4) = 'o16'
xa_function_weight(4) = 20
xa_function_param(4) = 1d-2

xa_function_species(5) = 'ne20'
xa_function_weight(5) = 20
xa_function_param(5) = 1d-2

!The following will be useless for the
!wind study, I don't go that far in the evolution

xa_function_species(6) = 'si28'
xa_function_weight(6) = 20
xa_function_param(6) = 1d-2

xa_function_species(7) = 's32'
xa_function_weight(7) = 20
xa_function_param(7) = 1d-2

xa_function_species(8) = 'fe54'
xa_function_weight(8) = 20
xa_function_param(8) = 1d-2

xa_function_species(9) = 'fe56'
xa_function_weight(9) = 20
xa_function_param(9) = 1d-2

!increase the resolution at the surface (avoid "fake-WR")
T_function2_weight = 110

! output to files and terminal
photostep = 50
profile_interval = 10
history_interval = 1
terminal_interval = 10
write_header_frequency = 1
max_num_profile_models = 1000

mixing_length_alpha = 2.0
MLT_option = 'Henyey'
```

```
allow_semiconvective_mixing = .true.  
alpha_semiconvection = 0.1  
overshoot_f_above_nonburn = 0.025  
overshoot_f_above_burn_h = 0.025  
overshoot_f_above_burn_he = 0.025  
overshoot_f_above_burn_z = 0.025  
overshoot_f_below_nonburn = 0.025  
overshoot_f_below_burn_h = 0.025  
overshoot_f_below_burn_he = 0.025  
overshoot_f_below_burn_z = 0.025
```

/

A.1.2 inlist for the Simplified Envelope Shedding Mass Loss Events

The inlist parameter files for each stripped mass model are all identical except for **NEW_M**, which is set to a specific value in units of M_{\odot} for each run, and **MODEL_TO_LOAD** which contains the path to the model of the reference run saved at the stripping point.

```
! inlist_project  
  
! contains the parameters controlling this job  
  
! for the sake of future readers of this (yourself included),  
! ONLY include the controls you are actually using.  
! DO NOT include all of the other controls that  
! simply have their default values.  
  
!This inlist has been created merging "inlist_massive_defaults" and  
"inlist_massive_15M_z1.9m2"  
  
&star_job  
  
mesa_dir = '' ! empty string defaults to environment variable MESA_DIR  
history_columns_file = './history_columns.list' ! if null string, use  
default.  
profile_columns_file = './profile_columns.list' ! if null string, use  
default.  
  
show_log_description_at_start = .true.  
show_net_species_info = .true.  
! show_net_reactions_info = .true.  
  
create_pre_main_sequence_model = .false.
```

```

change_net = .true.
new_net_name = 'approx21.net'

set_rates_preference = .true. ! for use by net + rates modules
new_rates_preference = 1 ! only used if set_rates_preference is true
! 1 = NACRE rates -- this is the default
! 2 = jina reaclib rates

eos_file_prefix = 'mesa'
kappa_file_prefix = 'gs98'

set_initial_age = .false.
set_initial_model_number=.false.

load_saved_model = .true.
saved_model_name = MODEL_TO_LOAD

relax_initial_mass = .true. ! gradually change total mass by a wind
new_mass = NEW_M !mass of the He core of the progenitor
!lg_max_abs_mdot = 2

write_profile_when_terminate = .true. ! write to a given name
filename_for_profile_when_terminate = 'onset_of_cc_stripped_1M.data'
save_photo_when_terminate = .true.
pause_before_terminate = .false.
!   pgstar_flag =.true.
pgstar_flag = .false.

/ !end of star_job

&controls

! starting specifications
initial_mass = 15 ! in Msun units
!solar chemical composition
initial_Z = 0.02
!initial_Y = 0.27

! high center T limit to avoid negative mass fractions
sig_min_factor_for_high_Tcenter = 0.01
! inactive when >= 1d0
! if Tcenter >= Tcenter_min_for_sig_min_factor_full_on,
! then okay to reduce sig by as much as this factor
! as needed to prevent causing negative abundances
Tcenter_min_for_sig_min_factor_full_on = 2d9
! if Tcenter >= this, factor = sig_min_factor_for_neg_abundances
! this should be > Tcenter_max_for_sig_min_factor_full_off.
Tcenter_max_for_sig_min_factor_full_off = 1d9

```

```

! if Tcenter <= this, factor = 1, so has no effect
! this should be < Tcenter_min_for_sig_min_factor_full_on.
! for T > full_off and < full_on, factor changes linearly with Tcenter

min_years_dt_for_redo_mdot = 1 ! experimental

num_trace_history_values = 0
trace_history_value_name(1) = 'surf_avg_v_div_v_crit'

! output to files and terminal
photostep = 10
profile_interval = 10
history_interval = 1
terminal_interval = 10
write_header_frequency = 1
max_num_profile_models = 100

velocity_logT_lower_bound = 8
max_dt_yrs_for_velocity_logT_lower_bound = 0.1

okay_to_reduce_gradT_excess = .true.

gradT_excess_f1 = 1d-4
gradT_excess_f2 = 1d-2
gradT_excess_age_fraction = 0.9d0

gradT_excess_lambda1 = 1.0
gradT_excess_beta1 = 0.35

gradT_excess_lambda2 = 0.5
gradT_excess_beta2 = 0.25

gradT_excess_dlambda = 0.1
gradT_excess_dbeta = 0.1

!Mass Loss
use_other_wind = .true.
RGB_wind_scheme = 'other'
AGB_wind_scheme = 'other'
Dutch_wind_eta = 1.0

use_Type2_opacities = .true.
Zbase = 0.02

xa_scale = 1d-5
newton_itermin = 2

mixing_length_alpha = 2.0

```

```

MLT_option = 'Henyey'

allow_semiconvective_mixing = .true.
alpha_semiconvection = 0.1
! allow_thermohaline_mixing = .false.
! thermo_haline_coeff = 2
! thermohaline_option = 'Kippenhahn'

overshoot_f_above_nonburn = 0.025
overshoot_f_above_burn_h = 0.025
overshoot_f_above_burn_he = 0.025
overshoot_f_above_burn_z = 0.025
overshoot_f_below_nonburn = 0.025
overshoot_f_below_burn_h = 0.025
overshoot_f_below_burn_he = 0.025
overshoot_f_below_burn_z = 0.025

set_min_D_mix = .true.      !D_mix is the diffusion coefficient for
    mixing of material
min_D_mix = 1d3            ! D_mix will be at least this large
min_center_Ye_for_min_D_mix = 0 !D_mix is used only for Ye>= than this

screening_mode = 'extended'

max_iter_for_resid_tol1 = 3 !in the first max_iter_for_resid_tol1
tol_residual_norm1 = 1d-5  !the maximum difference between lhs and rhs
    of the equation
tol_max_residual1 = 1d-2   !is determined by this parameters

max_iter_for_resid_tol2 = 12 !after the first max_iter_for_resid_tol1
    iterations
tol_residual_norm2 = 1d99  !use this instead
tol_max_residual2 = 1d99

delta_lgL_He_limit = 0.1  ! limit for magnitude of change in lgL_He

dX_nuc_drop_limit = 1d-3
delta_Ye_highT_limit = 1d-4

dX_nuc_drop_max_A_limit = 52
dX_nuc_drop_min_X_limit = 1d-3
dX_nuc_drop_hard_limit = 1d99

delta_lgTeff_limit = 0.5
delta_lgL_limit = 0.5
delta_lgRho_cntr_limit = 0.02

xa_function_species(1) = 'h1'

```

```

xa_function_weight(1) = 70
xa_function_param(1) = 1d-3

xa_function_species(2) = 'he4'
xa_function_weight(2) = 70
xa_function_param(2) = 1d-3

max_allowed_nz = 16000

mesh_delta_coeff_for_highT = 1.5
logT_max_for_standard_mesh_delta_coeff = 9.0
logT_min_for_highT_mesh_delta_coeff = 9.5

varcontrol_target = 1d-3
mesh_delta_coeff= 1.0
mesh_max_allowed_ratio=2.5
min_timestep_limit = 1d-8 !in sec
! when to stop
fe_core_infall_limit = 1d8

/

```

The reference run has an analogous inlist, except for the stopping criterion. The stopping criteria for the mSGB series,

```
Teff_lower_limit = 1d4
```

and for the hMR series,

```
photosphere_r_upper_limit = 375 !in units of Rsun
```

are set directly in the controls namelist of the unstripped reference model inlist. MESA does not provide settings to control the run depending on the extension of the convective envelope, therefore the stopping criterion for the MCE series is implemented in the `run_star_extras.f`, see §A.2.14.

Moreover, in the reference run `star_job` namelist, the following instructions are modified:

```

set_initial_age = .true.
initial_age = 0 ! in years

set_initial_model_number = .true.
initial_model_number = 0

!SOLAR CHEMICAL COMPOSITION
change_initial_Z = .true.
new_Z = 0.019d0
change_initial_Y = .true.
new_Y=0.27

```

Finally, the `relax_initial_mass` is set to `.false.` (i.e. it is omitted from the inlist to let MESA use the default).

A.2 Routines for the `run_star_extras.f`

In this section, I report the routines implemented for this study. §A.2.1–A.2.11 contain the mass loss routines used for the systematic comparison of wind algorithms, while §A.2.13 contains the timestep control routine, and the routines used to stop the models at the desired condition.

The `chem_def` “public” module is loaded outside the implemented routine, to be able to access the variables it defines (e.g. the index of each element in the nuclear reaction network, `ih1`, `ih4`, `ic12`, etc.) without reloading it module multiple times. This is done by including at the very beginning of the module `run_star_extras` the following line:

```
use chem_def           !to be able to call isotopes
```

These routines must be copied in the `run_star_extras.f` to be used, following the guidelines of http://mesa.sourceforge.net/run_star_extras.html#toc-3. Note in particular that to instruct MESA to use the custom mass loss routines, the ‘other’ wind scheme must be specified in the `inlist`, and the pointer `other_wind` must be directed to the mass loss routine name in the `run_star_extras.f` subroutine `extras_controls`, i.e. the latter must contain the line

```
s% other_wind => Dutch_tanh_intrp_wind
```

where `Dutch_tanh_intrp_wind` must be substituted by the appropriate routine name. Note also that any modification to the `run_star_extras.f` requires a compilation (of the local work directory) to be effective.

A.2.1 Vink *et al.*, de Jager *et al.*, Nugis & Lamers – VdJNL

```
subroutine Dutch_tanh_intrp_wind(id, Lsurf, Msurf, Rsurf, Tsurf, w,
    ierr)
use cstdlib_lib
type (star_info), pointer :: s
integer, intent(in) :: id
real(dp), intent(in) :: Lsurf, Msurf, Rsurf, Tsurf ! surface values
    (cgs)
real(dp), intent(out) :: w ! wind in units of Msun/year (value is >= 0)
integer, intent(out) :: ierr
real(dp) :: L1, M1, R1, T1, xsurf, etaWR, etaHOT, etaCOOL, Zsolar, Y, Z
real(dp) :: log10w, w1, w2, T_high, T_low, alfa
call get_star_ptr(id,s,ierr)
w = 0
ierr = 0
L1 = Lsurf
M1 = Msurf
R1 = Rsurf
T1 = Tsurf
Zsolar = 0.019
!I use the same factor everywhere
etaHOT = s% x_ctrl(1)
```

```
etaCOOL = s% x_ctrl(2)
etaWR = s% x_ctrl(3)
!Assumes the use of approx21.net
xsurf = s%xa(s% net_iso(iprot), 1)+ s%xa(s% net_iso(ih1), 1) !H mass
      fraction of the outermost cell, species 1 being neutrons
Y = s% xa(s%net_iso(ihe3),1) + s% xa(s%net_iso(ihe4),1)
Z = 1 - Y -xsurf
T_high = 11000d0
T_low = 10000d0
if (xsurf .GE. 0.4d0) then
  if (T1 <= T_low) then
    print *, 'using: de_Jager, eta=', etaCOOL
    call eval_de_Jager_wind(w)
    w = w * etaCOOL
  else if (T1 >= T_high) then
    print *, 'using: Vink_tanh, eta=', etaHOT
    call eval_tanh_intrp_vink_wind(w)
    w = w * etaHOT
  else
    ! transition
    print *, 'interpolating de Jager and Vink_tanh, eta=', etaHOT
    call eval_de_Jager_wind(w1)
    call eval_tanh_intrp_vink_wind(w2)
    alfa = (T1 - T_low)/(T_high - T_low)
    w = (1-alfa)*w1 + alfa*w2
    w = w * etaHOT
  end if
else
  !means it's a WR star
  print *, 'using: N&L, eta=', etaWR
  w = 1d-11 * pow_cr(L1/Lsun,1.29d0) * pow_cr(Y,1.7d0) * sqrt(Z)
  w = w * etaWR
end if

contains

subroutine eval_de_Jager_wind(w)
! de Jager, C., Nieuwenhuijzen, H., & van der Hucht, K. A. 1988, A&AS, 72,
259.
  real(dp), intent(out) :: w
  real(dp) :: log10w
  include 'formats'
  log10w = 1.769d0*log10_cr(L1/Lsun) - 1.676d0*log10_cr(T1) - 8.158d0
  w = exp10_cr(log10w)
end subroutine eval_de_Jager_wind

subroutine eval_tanh_intrp_vink_wind(w)
  real(dp), intent(inout) :: w
  real(dp) :: alfa, w1, w2, w_c, w_h, Teff_jump, logMdot, dT,
  vinf_div_vesc
```

```

w = 0

if (T1 > 27500d0) then
  vinf_div_vesc = 2.6d0 ! this is the hot side galactic value
  vinf_div_vesc = vinf_div_vesc*pow_cr(Z/Zsolar,0.13d0) !
    corrected for Z
  logMdot = &
    - 6.697d0 &
    + 2.194d0*log10_cr(L1/Lsun/1d5) &
    - 1.313d0*log10_cr(M1/Msun/30) &
    - 1.226d0*log10_cr(vinf_div_vesc/2d0) &
    + 0.933d0*log10_cr(T1/4d4) &
    - 10.92d0*pow2(log10_cr(T1/4d4)) &
    + 0.85d0*log10_cr(Z/Zsolar)
  w1 = exp10_cr(logMdot)
  w = w1
  else if (T1 < 22500d0) then
    vinf_div_vesc = 1.3d0 ! this is the cool side galactic value
    vinf_div_vesc = vinf_div_vesc*pow_cr(Z/Zsolar,0.13d0) ! corrected
      for Z
    logMdot = &
      - 6.688d0 &
      + 2.210d0*log10_cr(L1/Lsun/1d5) &
      - 1.339d0*log10_cr(M1/Msun/30) &
      - 1.601d0*log10_cr(vinf_div_vesc/2d0) &
      + 1.07d0*log10_cr(T1/2d4) &
      + 0.85d0*log10_cr(Z/Zsolar)
    w2 = exp10_cr(logMdot)
    w = w2
  else ! use Vink et al 2001, eqns 14 and 15 to set "jump"
    temperature
    Teff_jump = 1d3*(61.2d0 + 2.59d0*(-13.636d0 +
      0.889d0*log10_cr(Z/Zsolar)))
    dT = 100d0
    print *, 'Teff_jump[K]', Teff_jump
!   first evaluate the mass loss rate one would have if T = 27500, all
rest held constant
  vinf_div_vesc = 2.6d0 ! this is the hot side galactic value
  vinf_div_vesc = vinf_div_vesc*pow_cr(Z/Zsolar,0.13d0) !
    corrected for Z
  w_h = &
    - 6.697d0 &
    + 2.194d0*log10_cr(L1/Lsun/1d5) &
    - 1.313d0*log10_cr(M1/Msun/30) &
    - 1.226d0*log10_cr(vinf_div_vesc/2d0) &
    + 0.933d0*log10_cr(27500/4d4) &
    - 10.92d0*pow2(log10_cr(27500/4d4)) &
    + 0.85d0*log10_cr(Z/Zsolar)

```

```
w_h = exp10_cr(w_h)
!then evaluate the mass loss rate one would have if T= 22500,
  all rest held constant
vinf_div_vesc = 1.3d0 ! this is the cool side galactic value
vinf_div_vesc = vinf_div_vesc*pow_cr(Z/Zsolar,0.13d0) !
  corrected for Z
w_c = &
  - 6.688d0 &
  + 2.210d0*log10_cr(L1/Lsun/1d5) &
  - 1.339d0*log10_cr(M1/Msun/30) &
  - 1.601d0*log10_cr(vinf_div_vesc/2d0) &
  + 1.07d0*log10_cr(22500/2d4) &
  + 0.85d0*log10_cr(Z/Zsolar)
w_c = exp10_cr(w_c)
!then interpolate between this two values
w = ((w_h - w_c)/2)*TANH((T1-Teff_jump)/(1d0*dT)) &
  +(w_c+w_h)/2
end if

end subroutine eval_tanh_intrp_vink_wind
```

```
end subroutine Dutch_tanh_intrp_wind
```

A.2.2 Vink *et al.*, de Jager *et al.*, Hamann *et al.* – VdJH

```
subroutine VdJNL_wind(id, Lsurf, Msurf, Rsurf, Tsurf, w, ierr)
use cstdlibm_lib
type (star_info), pointer :: s
integer, intent(in) :: id
real(dp), intent(in) :: Lsurf, Msurf, Rsurf, Tsurf ! surface values
  (cgs)
! NOTE: surface is outermost cell. not necessarily at photosphere.
! NOTE: don't assume that vars are set at this point.
! so if you want values other than those given as args,
! you should use values from s% xh(:,.) and s% xa(:,.) only.
! rather than things like s% Teff or s% lnT(:) which have not been
  set yet.
real(dp), intent(out) :: w ! wind in units of Msun/year (value is >=
  0)
integer, intent(out) :: ierr
real(dp) :: L1, M1, R1, T1, xsurf, etaWR, Zsolar, Y, Z, etaCOOL,
  etaHOT
real(dp) :: log10w
call get_star_ptr(id,s,ierr)
w = 0
ierr = 0
L1 = Lsurf
```

```
M1 = Msurf
R1 = Rsurf
T1 = Tsurf
etaHOT = s% x_ctrl(1)
etaCOOL = s% x_ctrl(2)
etaWR = s% x_ctrl(3)
Zsolar = 0.019
xsurf = s%xa(s% net_iso(iprot), 1)+ s%xa(s% net_iso(ih1), 1) !H
      mass fraction of the outermost cell, species 1 being neutrons
Y = s% xa(s%net_iso(ihe3),1) + s% xa(s%net_iso(ihe4),1)
Z = 1 - Y -xsurf
if (xsurf .GE. 0.4d0) then
  if (T1 .LT. 15000d0) then
    print *, 'Lsurf=', L1, 'Tsurf=', Tsurf , 'Xsurf=', xsurf,
      'using:de Jager', etaCOOL
    call eval_de_Jager_wind(w)
    w = w *etaCOOL
  else
    print *, 'Lsurf=', L1, 'Tsurf=', Tsurf , 'Xsurf=', xsurf,
      'using: vink_tanh', etaHOT
    call eval_tanh_intrp_vink_wind(w)
    w = w* etaHOT
  end if
else
      !the thing is a WR star
  print *, 'Lsurf=', L1, 'Tsurf=', Tsurf , 'Xsurf=', xsurf, 'using:
    Hamman', etaWR
  if (log10_cr(L1/Lsun) .LE. 4.5d0) then
    log10w = -35.8d0 + 6.8d0*log10_cr(L1/Lsun)
    w = exp10_cr(log10w)* etaWR
  else
    log10w = -11.95d0 + 1.5d0*log10_cr(L1/Lsun) - 2.85d0*(xsurf)
    w = exp10_cr(log10w)* etaWR
  end if
end if

contains

subroutine eval_de_Jager_wind(w)
  ! de Jager, C., Nieuwenhuijzen, H., & van der Hucht, K. A. 1988,
  A&AS, 72, 259.
  real(dp), intent(out) :: w
  real(dp) :: log10w
  include 'formats'
  log10w = 1.769d0*log10_cr(L1/Lsun) - 1.676d0*log10_cr(T1) -
    8.158d0
  w = exp10_cr(log10w)
end subroutine eval_de_Jager_wind
```

```

subroutine eval_tanh_intrp_vink_wind(w)
  real(dp), intent(inout) :: w
  real(dp) :: alfa, w1, w2, w_c, w_h, Teff_jump, logMdot, dT,
    vinf_div_vesc
  w = 0

  if (T1 > 27500d0) then
    vinf_div_vesc = 2.6d0 ! this is the hot side galactic value
    vinf_div_vesc = vinf_div_vesc*pow_cr(Z/Zsolar,0.13d0) !
      corrected for Z
    logMdot = &
      - 6.697d0 &
      + 2.194d0*log10_cr(L1/Lsun/1d5) &
      - 1.313d0*log10_cr(M1/Msun/30) &
      - 1.226d0*log10_cr(vinf_div_vesc/2d0) &
      + 0.933d0*log10_cr(T1/4d4) &
      - 10.92d0*pow2(log10_cr(T1/4d4)) &
      + 0.85d0*log10_cr(Z/Zsolar)
    w1 = exp10_cr(logMdot)
    w = w1
  else if (T1 < 22500d0) then
    vinf_div_vesc = 1.3d0 ! this is the cool side galactic value
    vinf_div_vesc = vinf_div_vesc*pow_cr(Z/Zsolar,0.13d0) ! corrected
      for Z
    logMdot = &
      - 6.688d0 &
      + 2.210d0*log10_cr(L1/Lsun/1d5) &
      - 1.339d0*log10_cr(M1/Msun/30) &
      - 1.601d0*log10_cr(vinf_div_vesc/2d0) &
      + 1.07d0*log10_cr(T1/2d4) &
      + 0.85d0*log10_cr(Z/Zsolar)
    w2 = exp10_cr(logMdot)
    w = w2
  else ! use Vink et al 2001, eqns 14 and 15 to set "jump"
    temperature
    Teff_jump = 1d3*(61.2d0 + 2.59d0*(-13.636d0 +
      0.889d0*log10_cr(Z/Zsolar)))
    dT = 100d0
    print *, 'Teff_jump[K]', Teff_jump
  ! first evaluate the mass loss rate one would have if T = 27500, all
  rest held constant
    vinf_div_vesc = 2.6d0 ! this is the hot side galactic value
    vinf_div_vesc = vinf_div_vesc*pow_cr(Z/Zsolar,0.13d0) !
      corrected for Z
    w_h = &
      - 6.697d0 &
      + 2.194d0*log10_cr(L1/Lsun/1d5) &
      - 1.313d0*log10_cr(M1/Msun/30) &

```

```

- 1.226d0*log10_cr(vinf_div_vesc/2d0) &
+ 0.933d0*log10_cr(27500/4d4) &
- 10.92d0*pow2(log10_cr(27500/4d4)) &
+ 0.85d0*log10_cr(Z/Zsolar)
w_h = exp10_cr(w_h)
!then evaluate the mass loss rate one would have if T= 22500,
  all rest held constant
vinf_div_vesc = 1.3d0 ! this is the cool side galactic value
vinf_div_vesc = vinf_div_vesc*pow_cr(Z/Zsolar,0.13d0) !
  corrected for Z
w_c = &
- 6.688d0 &
+ 2.210d0*log10_cr(L1/Lsun/1d5) &
- 1.339d0*log10_cr(M1/Msun/30) &
- 1.601d0*log10_cr(vinf_div_vesc/2d0) &
+ 1.07d0*log10_cr(22500/2d4) &
+ 0.85d0*log10_cr(Z/Zsolar)
w_c = exp10_cr(w_c)
!then interpolate between this two values
w = ((w_h - w_c)/2)*TANH((T1-Teff_jump)/(1d0*dT)) &
  +(w_c+w_h)/2
!print *, 'USING TANH(',(T1-Teff_jump)/(10d0*dT), &
!      ')=',&
!      TANH(T1-Teff_jump)/(10d0*dT), 'w_c=', &
!      w_c,&
!      'w_h=', w_h, 'w=',w
end if
end subroutine eval_tanh_intrp_vink_wind

```

```
end subroutine VdJNL_wind
```

A.2.3 Vink *et al.*, Nieuwenhuijzen *et al.*, Nugis & Lamers – VNJNL

```

subroutine VNJNL_wind(id, Lsurf, Msurf, Rsurf, Tsurf, w, ierr)
use cstdlib_lib
type (star_info), pointer :: s
integer, intent(in) :: id
real(dp), intent(in) :: Lsurf, Msurf, Rsurf, Tsurf ! surface values
  (cgs)
! NOTE: surface is outermost cell. not necessarily at photosphere.
! NOTE: don't assume that vars are set at this point.
! so if you want values other than those given as args,
! you should use values from s% xh(:, :) and s% xa(:, :) only.
! rather than things like s% Teff or s% lnT(:) which have not been
  set yet.

```

```

real(dp), intent(out) :: w ! wind in units of Msun/year (value is >=
0)
integer, intent(out) :: ierr
real(dp) :: L1, M1, R1, T1, xsurf, etaWR, Zsolar, Y, Z, etaCOOL,
etaHOT
real(dp) :: log10w, w1, w2, T_high, T_low, alfa
call get_star_ptr(id,s,ierr)
w = 0
ierr = 0
L1 = Lsurf
M1 = Msurf
R1 = Rsurf
T1 = Tsurf
Zsolar = 0.019

!I use the same factor everywhere
etaHOT = s% x_ctrl(1)
etaCOOL = s% x_ctrl(2)
etaWR = s% x_ctrl(3) !eta is the factor for WR winds
!Assumes the use of approx21.net
xsurf = s%xa(s% net_iso(iprot), 1)+ s%xa(s% net_iso(ih1), 1) !H
mass fraction of the outermost cell, species 1 being neutrons
Y = s% xa(s%net_iso(ihe3),1) + s% xa(s%net_iso(ihe4),1)
Z = 1 - Y -xsurf

T_high = 11000d0
T_low = 10000d0
if (xsurf .GE. 0.4d0) then
  if (T1 <= T_low) then
    print *, 'using: NJ, eta=', etaCOOL
    call eval_Nieuwenhuijzen_wind(w)
    w = w * etaCOOL
  else if (T1 >= T_high) then
    print *, 'using: Vink_tanh, eta=', etaHOT
    call eval_tanh_intrp_vink_wind(w)
    w = w * etaHOT
  else
    ! transition
    print *, 'interpolating NJ and Vink_tanh, eta=', etaHOT
    call eval_Nieuwenhuijzen_wind(w1)
    call eval_tanh_intrp_vink_wind(w2)
    alfa = (T1 - T_low)/(T_high - T_low)
    w = (1-alfa)*w1 + alfa*w2
    w = w * etaHOT
  end if
else !means it's a WR star
  print *, 'using: N&L, eta=', etaWR
  w = 1d-11 * pow_cr(L1/Lsun,1.29d0) * pow_cr(Y,1.7d0) * sqrt(Z)
  w = w * etaWR
end if

```

contains

```

subroutine eval_Nieuwenhuijzen_wind(w)
  ! Nieuwenhuijzen, H.; de Jager, C. 1990, A&A, 231, 134 (eqn 2)
  real(dp), intent(out) :: w
  real(dp) :: log10w
  include 'formats'
  log10w = -14.02d0 + &
           1.24d0*log10_cr(L1/Lsun) + &
           0.16d0*log10_cr(M1/Msun) + &
           0.81d0*log10_cr(R1/Rsun)
  w = exp10_cr(log10w)
end subroutine eval_Nieuwenhuijzen_wind

```

```

subroutine eval_tanh_intrp_vink_wind(w)
  real(dp), intent(inout) :: w
  real(dp) :: alfa, w1, w2, w_c, w_h, Teff_jump, logMdot, dT,
    vinf_div_vesc
  w = 0

  if (T1 > 27500d0) then
    vinf_div_vesc = 2.6d0 ! this is the hot side galactic value
    vinf_div_vesc = vinf_div_vesc*pow_cr(Z/Zsolar,0.13d0) !
      corrected for Z
    logMdot = &
      - 6.697d0 &
      + 2.194d0*log10_cr(L1/Lsun/1d5) &
      - 1.313d0*log10_cr(M1/Msun/30) &
      - 1.226d0*log10_cr(vinf_div_vesc/2d0) &
      + 0.933d0*log10_cr(T1/4d4) &
      - 10.92d0*pow2(log10_cr(T1/4d4)) &
      + 0.85d0*log10_cr(Z/Zsolar)
    w1 = exp10_cr(logMdot)
    w = w1
  else if (T1 < 22500d0) then
    vinf_div_vesc = 1.3d0 ! this is the cool side galactic value
    vinf_div_vesc = vinf_div_vesc*pow_cr(Z/Zsolar,0.13d0) ! corrected
      for Z
    logMdot = &
      - 6.688d0 &
      + 2.210d0*log10_cr(L1/Lsun/1d5) &
      - 1.339d0*log10_cr(M1/Msun/30) &
      - 1.601d0*log10_cr(vinf_div_vesc/2d0) &
      + 1.07d0*log10_cr(T1/2d4) &
      + 0.85d0*log10_cr(Z/Zsolar)

```

```

        w2 = exp10_cr(logMdot)
        w = w2
    else ! use Vink et al 2001, eqns 14 and 15 to set "jump"
        temperature
        Teff_jump = 1d3*(61.2d0 + 2.59d0*(-13.636d0 +
            0.889d0*log10_cr(Z/Zsolar)))
        dT = 100d0
        print *, 'Teff_jump[K]', Teff_jump
!    first evaluate the mass loss rate one would have if T = 27500, all
rest held constant
        vinf_div_vesc = 2.6d0 ! this is the hot side galactic value
        vinf_div_vesc = vinf_div_vesc*pow_cr(Z/Zsolar,0.13d0) !
            corrected for Z
        w_h = &
            - 6.697d0 &
            + 2.194d0*log10_cr(L1/Lsun/1d5) &
            - 1.313d0*log10_cr(M1/Msun/30) &
            - 1.226d0*log10_cr(vinf_div_vesc/2d0) &
            + 0.933d0*log10_cr(27500/4d4) &
            - 10.92d0*pow2(log10_cr(27500/4d4)) &
            + 0.85d0*log10_cr(Z/Zsolar)
        w_h = exp10_cr(w_h)
        !then evaluate the mass loss rate one would have if T= 22500,
            all rest held constant
        vinf_div_vesc = 1.3d0 ! this is the cool side galactic value
        vinf_div_vesc = vinf_div_vesc*pow_cr(Z/Zsolar,0.13d0) !
            corrected for Z
        w_c = &
            - 6.688d0 &
            + 2.210d0*log10_cr(L1/Lsun/1d5) &
                - 1.339d0*log10_cr(M1/Msun/30) &
            - 1.601d0*log10_cr(vinf_div_vesc/2d0) &
            + 1.07d0*log10_cr(22500/2d4) &
            + 0.85d0*log10_cr(Z/Zsolar)
        w_c = exp10_cr(w_c)
        !then interpolate between this two values
        w = ((w_h - w_c)/2)*TANH((T1-Teff_jump)/(1d0*dT)) &
            +(w_c+w_h)/2
    end if

end subroutine eval_tanh_intrp_vink_wind

end subroutine VNJNL_wind

```

A.2.4 Vink et al., Nieuwenhuijzen et al., Hamann et al. – VNJH

```

subroutine VNJH_wind(id, Lsurf, Msurf, Rsurf, Tsurf, w, ierr)
use cstdlib_lib

```

```
type (star_info), pointer :: s
integer, intent(in) :: id
real(dp), intent(in) :: Lsurf, Msurf, Rsurf, Tsurf ! surface values
    (cgs)
! NOTE: surface is outermost cell. not necessarily at photosphere.
! NOTE: don't assume that vars are set at this point.
! so if you want values other than those given as args,
! you should use values from s% xh(:, :) and s% xa(:, :) only.
! rather than things like s% Teff or s% lnT(:) which have not been
    set yet.
real(dp), intent(out) :: w ! wind in units of Msun/year (value is >=
    0)
integer, intent(out) :: ierr
real(dp) :: L1, M1, R1, T1, xsurf, etaWR, Zsolar, Y, Z, etaCOOL,
    etaHOT
real(dp) :: log10w, w1, w2, T_high, T_low, alfa
call get_star_ptr(id,s,ierr)
w = 0
ierr = 0
L1 = Lsurf
M1 = Msurf
R1 = Rsurf
T1 = Tsurf
Zsolar = 0.019

!I use the same factor everywhere
etaHOT = s% x_ctrl(1)
etaCOOL = s% x_ctrl(2)
etaWR = s% x_ctrl(3) !eta is the factor for WR winds
xsurf = s%xa(s% net_iso(iprot), 1)+ s%xa(s% net_iso(ih1), 1) !H
    mass fraction of the outermost cell, species 1 being neutrons
Y = s% xa(s%net_iso(ihe3),1) + s% xa(s%net_iso(ihe4),1)
Z = 1 - Y -xsurf
T_high = 11000d0
T_low = 10000d0
if (xsurf .GE. 0.4d0) then
    if (T1 <= T_low) then
        print *, 'using: NJ, eta=', etaCOOL
        call eval_Nieuwenhuijzen_wind(w)
        w = w * etaCOOL
    else if (T1 >= T_high) then
        print *, 'using: Vink_tanh, eta=', etaHOT
        call eval_tanh_intrp_vink_wind(w)
        w = w * etaHOT
    else
        ! transition
        print *, 'interpolating NJ and Vink_tanh, eta=', etaHOT
        call eval_Nieuwenhuijzen_wind(w1)
        call eval_tanh_intrp_vink_wind(w2)
```

```
        alfa = (T1 - T_low)/(T_high - T_low)
        w = (1-alfa)*w1 + alfa*w2
        w = w * etaHOT
    end if
else
    !the thing is a WR star
    print *, 'Lsurf=', L1, 'Tsurf=', Tsurf, 'Xsurf=', xsurf, 'using:
        manual Hamann', etaWR
    if (log10_cr(L1/Lsun) .LE. 4.5d0) then
        log10w = -35.8d0 + 6.8d0*log10_cr(L1/Lsun)
        w = exp10_cr(log10w)* etaWR
    else
        log10w = -11.95d0 + 1.5d0*log10_cr(L1/Lsun) - 2.85d0*(xsurf)
        w = exp10_cr(log10w)* etaWR
    end if
end if

contains

subroutine eval_Nieuwenhuijzen_wind(w)
    ! Nieuwenhuijzen, H.; de Jager, C. 1990, A&A, 231, 134 (eqn 2)
    real(dp), intent(out) :: w
    real(dp) :: log10w
    include 'formats'
    log10w = -14.02d0 + &
        1.24d0*log10_cr(L1/Lsun) + &
        0.16d0*log10_cr(M1/Msun) + &
        0.81d0*log10_cr(R1/Rsun)
    w = exp10_cr(log10w)
end subroutine eval_Nieuwenhuijzen_wind

subroutine eval_tanh_intrp_vink_wind(w)
    real(dp), intent(inout) :: w
    real(dp) :: alfa, w1, w2, w_c, w_h, Teff_jump, logMdot, dT,
        vinf_div_vesc
    w = 0

    if (T1 > 27500d0) then
        vinf_div_vesc = 2.6d0 ! this is the hot side galactic value
        vinf_div_vesc = vinf_div_vesc*pow_cr(Z/Zsolar,0.13d0) !
            corrected for Z
        logMdot = &
            - 6.697d0 &
            + 2.194d0*log10_cr(L1/Lsun/1d5) &
            - 1.313d0*log10_cr(M1/Msun/30) &
            - 1.226d0*log10_cr(vinf_div_vesc/2d0) &
            + 0.933d0*log10_cr(T1/4d4) &
            - 10.92d0*pow2(log10_cr(T1/4d4)) &
            + 0.85d0*log10_cr(Z/Zsolar)
```

```
w1 = exp10_cr(logMdot)
w = w1
else if (T1 < 22500d0) then
vinf_div_vesc = 1.3d0 ! this is the cool side galactic value
vinf_div_vesc = vinf_div_vesc*pow_cr(Z/Zsolar,0.13d0) ! corrected
for Z
logMdot = &
- 6.688d0 &
+ 2.210d0*log10_cr(L1/Lsun/1d5) &
- 1.339d0*log10_cr(M1/Msun/30) &
- 1.601d0*log10_cr(vinf_div_vesc/2d0) &
+ 1.07d0*log10_cr(T1/2d4) &
+ 0.85d0*log10_cr(Z/Zsolar)
w2 = exp10_cr(logMdot)
w = w2
else ! use Vink et al 2001, eqns 14 and 15 to set "jump"
temperature
Teff_jump = 1d3*(61.2d0 + 2.59d0*(-13.636d0 +
0.889d0*log10_cr(Z/Zsolar)))
dT = 100d0
print *, 'Teff_jump[K]', Teff_jump
! first evaluate the mass loss rate one would have if T = 27500, all
rest held constant
vinf_div_vesc = 2.6d0 ! this is the hot side galactic value
vinf_div_vesc = vinf_div_vesc*pow_cr(Z/Zsolar,0.13d0) !
corrected for Z
w_h = &
- 6.697d0 &
+ 2.194d0*log10_cr(L1/Lsun/1d5) &
- 1.313d0*log10_cr(M1/Msun/30) &
- 1.226d0*log10_cr(vinf_div_vesc/2d0) &
+ 0.933d0*log10_cr(27500/4d4) &
- 10.92d0*pow2(log10_cr(27500/4d4)) &
+ 0.85d0*log10_cr(Z/Zsolar)
w_h = exp10_cr(w_h)
!then evaluate the mass loss rate one would have if T= 22500,
all rest held constant
vinf_div_vesc = 1.3d0 ! this is the cool side galactic value
vinf_div_vesc = vinf_div_vesc*pow_cr(Z/Zsolar,0.13d0) !
corrected for Z
w_c = &
- 6.688d0 &
+ 2.210d0*log10_cr(L1/Lsun/1d5) &
- 1.339d0*log10_cr(M1/Msun/30) &
- 1.601d0*log10_cr(vinf_div_vesc/2d0) &
+ 1.07d0*log10_cr(22500/2d4) &
+ 0.85d0*log10_cr(Z/Zsolar)
w_c = exp10_cr(w_c)
```

```

!then interpolate between this two values
w = ((w_h - w_c)/2)*TANH((T1-Teff_jump)/(1d0*dT)) &
    +(w_c+w_h)/2
!print *, 'USING TANH(',(T1-Teff_jump)/(10d0*dT), &
!      ')=',&
!      TANH(T1-Teff_jump)/(10d0*dT), 'w_c=', &
!      w_c,&
!      'w_h=', w_h, 'w=' ,w
end if

end subroutine eval_tanh_intrp_vink_wind

end subroutine VNJH_wind

```

A.2.5 Vink *et al.*, van Loon *et al.*, Hamann – VvLH

```

subroutine VvLH_wind(id, Lsurf, Msurf, Rsurf, Tsurf, w, ierr)
use cstdlib_lib
type (star_info), pointer :: s
integer, intent(in) :: id
real(dp), intent(in) :: Lsurf, Msurf, Rsurf, Tsurf ! surface values
    (cgs)
! NOTE: surface is outermost cell. not necessarily at photosphere.
! NOTE: don't assume that vars are set at this point.
! so if you want values other than those given as args,
! you should use values from s% xh(:, :) and s% xa(:, :) only.
! rather than things like s% Teff or s% lnT(:) which have not been
    set yet.
real(dp), intent(out) :: w ! wind in units of Msun/year (value is >=
    0)
integer, intent(out) :: ierr
real(dp) :: L1, M1, R1, T1, xsurf, etaWR, etaHOT, etaCOOL, Zsolar,
    Y, Z
real(dp) :: log10w, w1, w2, T_high, T_low, alfa
call get_star_ptr(id,s,ierr)
w = 0
ierr = 0
L1 = Lsurf
M1 = Msurf
R1 = Rsurf
T1 = Tsurf
Zsolar = 0.019
!I use the same factor everywhere
etaHOT = s% x_ctrl(1)
etaCOOL = s% x_ctrl(2)
etaWR = s% x_ctrl(3) !eta is the factor for WR winds
xsurf = s%xa(s% net_iso(iprot), 1)+ s%xa(s% net_iso(ih1), 1) !H
    mass fraction of the outermost cell, species 1 being neutrons

```

```
Y = s% xa(s%net_iso(ihe3),1) + s% xa(s%net_iso(ihe4),1)
Z = 1 - Y -xsurf
T_high = 11000d0
T_low = 10000d0
if (xsurf .GE. 0.4d0) then
  if (T1 <= T_low) then
    print *, 'using: van loon, eta=', etaCOOL
    call eval_van_Loon_wind(w)
    w = w * etaCOOL
  else if (T1 >= T_high) then
    print *, 'using: Vink_tanh, eta=', etaHOT
    call eval_tanh_intrp_vink_wind(w)
    w = w * etaHOT
  else
    ! transition
    print *, 'interpolating de Jager and Vink_tanh, eta=', etaHOT
    call eval_van_Loon_wind(w1)
    call eval_tanh_intrp_vink_wind(w2)
    alfa = (T1 - T_low)/(T_high - T_low)
    w = (1-alfa)*w1 + alfa*w2
    w = w * etaHOT
  end if
else !means it's a WR star
  print *, 'Lsurf=', L1, 'Tsurf=', Tsurf , 'Xsurf=', xsurf, 'using:
    Hamman', etaWR
  if (log10_cr(L1/Lsun) .LE. 4.5d0) then
    log10w = -35.8d0 + 6.8d0*log10_cr(L1/Lsun)
    w = exp10_cr(log10w)* etaWR
  else
    log10w = -11.95d0 + 1.5d0*log10_cr(L1/Lsun) - 2.85d0*(xsurf)
    w = exp10_cr(log10w)* etaWR
  end if
end if

contains

subroutine eval_van_Loon_wind(w)
  ! van Loon et al. 2005, A&A, 438, 273
  real(dp), intent(out) :: w
  real(dp) :: log10w
  include 'formats'
  log10w = -5.65d0 + 1.05*log10_cr(L1/(1d4*Lsun)) -
    6.3d0*log10_cr(T1/35d2)
  w = exp10_cr(log10w)
end subroutine eval_van_Loon_wind

subroutine eval_tanh_intrp_vink_wind(w)
  real(dp), intent(inout) :: w
  real(dp) :: alfa, w1, w2, w_c, w_h, Teff_jump, logMdot, dT,
```

```

    vinf_div_vesc
    w = 0

if (T1 > 27500d0) then
    vinf_div_vesc = 2.6d0 ! this is the hot side galactic value
    vinf_div_vesc = vinf_div_vesc*pow_cr(Z/Zsolar,0.13d0) !
        corrected for Z
    logMdot = &
        - 6.697d0 &
        + 2.194d0*log10_cr(L1/Lsun/1d5) &
        - 1.313d0*log10_cr(M1/Msun/30) &
        - 1.226d0*log10_cr(vinf_div_vesc/2d0) &
        + 0.933d0*log10_cr(T1/4d4) &
        - 10.92d0*pow2(log10_cr(T1/4d4)) &
        + 0.85d0*log10_cr(Z/Zsolar)
    w1 = exp10_cr(logMdot)
    w = w1
    else if (T1 < 22500d0) then
    vinf_div_vesc = 1.3d0 ! this is the cool side galactic value
    vinf_div_vesc = vinf_div_vesc*pow_cr(Z/Zsolar,0.13d0) ! corrected
        for Z
    logMdot = &
        - 6.688d0 &
        + 2.210d0*log10_cr(L1/Lsun/1d5) &
        - 1.339d0*log10_cr(M1/Msun/30) &
        - 1.601d0*log10_cr(vinf_div_vesc/2d0) &
        + 1.07d0*log10_cr(T1/2d4) &
        + 0.85d0*log10_cr(Z/Zsolar)
    w2 = exp10_cr(logMdot)
    w = w2
    else ! use Vink et al 2001, eqns 14 and 15 to set "jump"
        temperature
    Teff_jump = 1d3*(61.2d0 + 2.59d0*(-13.636d0 +
        0.889d0*log10_cr(Z/Zsolar)))
    dT = 100d0
    print *, 'Teff_jump[K]', Teff_jump
! first evaluate the mass loss rate one would have if T = 27500, all
rest held constant
    vinf_div_vesc = 2.6d0 ! this is the hot side galactic value
    vinf_div_vesc = vinf_div_vesc*pow_cr(Z/Zsolar,0.13d0) !
        corrected for Z
    w_h = &
        - 6.697d0 &
        + 2.194d0*log10_cr(L1/Lsun/1d5) &
        - 1.313d0*log10_cr(M1/Msun/30) &
        - 1.226d0*log10_cr(vinf_div_vesc/2d0) &
        + 0.933d0*log10_cr(27500/4d4) &
        - 10.92d0*pow2(log10_cr(27500/4d4)) &

```

```

      + 0.85d0*log10_cr(Z/Zsolar)
    w_h = exp10_cr(w_h)
    !then evaluate the mass loss rate one would have if T= 22500,
      all rest held constant
    vinf_div_vesc = 1.3d0 ! this is the cool side galactic value
    vinf_div_vesc = vinf_div_vesc*pow_cr(Z/Zsolar,0.13d0) !
      corrected for Z
    w_c = &
      - 6.688d0 &
      + 2.210d0*log10_cr(L1/Lsun/1d5) &
      - 1.339d0*log10_cr(M1/Msun/30) &
      - 1.601d0*log10_cr(vinf_div_vesc/2d0) &
      + 1.07d0*log10_cr(22500/2d4) &
      + 0.85d0*log10_cr(Z/Zsolar)
    w_c = exp10_cr(w_c)
    !then interpolate between this two values
    w = ((w_h - w_c)/2)*TANH((T1-Teff_jump)/(1d0*dT)) &
      +(w_c+w_h)/2
    !print *, 'USING TANH(',(T1-Teff_jump)/(10d0*dT), &
    !      ')=',&
    !      TANH(T1-Teff_jump)/(10d0*dT), 'w_c=', &
    !      w_c,&
    !      'w_h=', w_h, 'w=',w
  end if

end subroutine eval_tanh_intrp_vink_wind

```

```
end subroutine VvLH_wind
```

A.2.6 Vink *et al.*, van Loon *et al.*, Nugis & Lamers – VvLNL

```

subroutine VvLNL_wind(id, Lsurf, Msurf, Rsurf, Tsurf, w, ierr)
  use cstdlib_lib
  type (star_info), pointer :: s
  integer, intent(in) :: id
  real(dp), intent(in) :: Lsurf, Msurf, Rsurf, Tsurf ! surface values
    (cgs)
  ! NOTE: surface is outermost cell. not necessarily at photosphere.
  ! NOTE: don't assume that vars are set at this point.
  ! so if you want values other than those given as args,
  ! you should use values from s% xh(:, :) and s% xa(:, :) only.
  ! rather than things like s% Teff or s% lnT(:) which have not been
    set yet.
  real(dp), intent(out) :: w ! wind in units of Msun/year (value is >=
    0)
  integer, intent(out) :: ierr
  real(dp) :: L1, M1, R1, T1, xsurf, etaWR, etaHOT, etaCOOL, Zsolar,

```

```

    Y, Z
real(dp) :: log10w, w1, w2, T_high, T_low, alfa
call get_star_ptr(id,s,ierr)
w = 0
ierr = 0
L1 = Lsurf
M1 = Msurf
R1 = Rsurf
T1 = Tsurf
Zsolar = 0.019
!I use the same factor everywhere
etaHOT = s% x_ctrl(1)
etaCOOL = s% x_ctrl(2)
etaWR = s% x_ctrl(3) !eta is the factor for WR winds
xsurf = s%xa(s% net_iso(iprot), 1)+ s%xa(s% net_iso(ih1), 1) !H
    mass fraction of the outermost cell, species 1 being neutrons
Y = s% xa(s%net_iso(ihe3),1) + s% xa(s%net_iso(ihe4),1)
Z = 1 - Y -xsurf
T_high = 11000d0
T_low = 10000d0
if (xsurf .GE. 0.4d0) then
    if (T1 <= T_low) then
        print *, 'using: de_Jager, eta=', etaCOOL
        call eval_van_Loon_wind(w)
        w = w * etaCOOL
    else if (T1 >= T_high) then
        print *, 'using: Vink_tanh, eta=', etaHOT
        call eval_tanh_intrp_vink_wind(w)
        w = w * etaHOT
    else
        ! transition
        print *, 'interpolating de Jager and Vink_tanh, eta=', etaHOT
        call eval_van_Loon_wind(w1)
        call eval_tanh_intrp_vink_wind(w2)
        alfa = (T1 - T_low)/(T_high - T_low)
        w = (1-alfa)*w1 + alfa*w2
        w = w * etaHOT
    end if
else !means it's a WR star
    print *, 'using: N&L, eta=', etaWR
    w = 1d-11 * pow_cr(L1/Lsun,1.29d0) * pow_cr(Y,1.7d0) * sqrt(Z)
    w = w * etaWR
end if

contains

subroutine eval_van_Loon_wind(w)
    ! van Loon et al. 2005, A&A, 438, 273
    real(dp), intent(out) :: w

```

```
    real(dp) :: log10w
    include 'formats'
    log10w = -5.65d0 + 1.05*log10_cr(L1/(1d4*Lsun)) -
            6.3d0*log10_cr(T1/35d2)
    w = exp10_cr(log10w)
end subroutine eval_van_Loon_wind

subroutine eval_tanh_intrp_vink_wind(w)
    real(dp), intent(inout) :: w
    real(dp) :: alfa, w1, w2, w_c, w_h, Teff_jump, logMdot, dT,
            vinf_div_vesc
    w = 0

    if (T1 > 27500d0) then
        vinf_div_vesc = 2.6d0 ! this is the hot side galactic value
        vinf_div_vesc = vinf_div_vesc*pow_cr(Z/Zsolar,0.13d0) !
            corrected for Z
        logMdot = &
            - 6.697d0 &
            + 2.194d0*log10_cr(L1/Lsun/1d5) &
            - 1.313d0*log10_cr(M1/Msun/30) &
            - 1.226d0*log10_cr(vinf_div_vesc/2d0) &
            + 0.933d0*log10_cr(T1/4d4) &
            - 10.92d0*pow2(log10_cr(T1/4d4)) &
            + 0.85d0*log10_cr(Z/Zsolar)
        w1 = exp10_cr(logMdot)
        w = w1
    else if (T1 < 22500d0) then
        vinf_div_vesc = 1.3d0 ! this is the cool side galactic value
        vinf_div_vesc = vinf_div_vesc*pow_cr(Z/Zsolar,0.13d0) ! corrected
            for Z
        logMdot = &
            - 6.688d0 &
            + 2.210d0*log10_cr(L1/Lsun/1d5) &
            - 1.339d0*log10_cr(M1/Msun/30) &
            - 1.601d0*log10_cr(vinf_div_vesc/2d0) &
            + 1.07d0*log10_cr(T1/2d4) &
            + 0.85d0*log10_cr(Z/Zsolar)
        w2 = exp10_cr(logMdot)
        w = w2
    else ! use Vink et al 2001, eqns 14 and 15 to set "jump"
            temperature
        Teff_jump = 1d3*(61.2d0 + 2.59d0*(-13.636d0 +
            0.889d0*log10_cr(Z/Zsolar)))
        dT = 100d0
        print *, 'Teff_jump[K]', Teff_jump
    ! first evaluate the mass loss rate one would have if T = 27500, all
        rest held constant
```

```

vinf_div_vesc = 2.6d0 ! this is the hot side galactic value
vinf_div_vesc = vinf_div_vesc*pow_cr(Z/Zsolar,0.13d0) !
    corrected for Z
w_h = &
    - 6.697d0 &
    + 2.194d0*log10_cr(L1/Lsun/1d5) &
    - 1.313d0*log10_cr(M1/Msun/30) &
    - 1.226d0*log10_cr(vinf_div_vesc/2d0) &
    + 0.933d0*log10_cr(27500/4d4) &
    - 10.92d0*pow2(log10_cr(27500/4d4)) &
    + 0.85d0*log10_cr(Z/Zsolar)
w_h = exp10_cr(w_h)
!then evaluate the mass loss rate one would have if T= 22500,
    all rest held constant
vinf_div_vesc = 1.3d0 ! this is the cool side galactic value
vinf_div_vesc = vinf_div_vesc*pow_cr(Z/Zsolar,0.13d0) !
    corrected for Z
w_c = &
    - 6.688d0 &
    + 2.210d0*log10_cr(L1/Lsun/1d5) &
    - 1.339d0*log10_cr(M1/Msun/30) &
    - 1.601d0*log10_cr(vinf_div_vesc/2d0) &
    + 1.07d0*log10_cr(22500/2d4) &
    + 0.85d0*log10_cr(Z/Zsolar)
w_c = exp10_cr(w_c)
!then interpolate between this two values
w = ((w_h - w_c)/2)*TANH((T1-Teff_jump)/(1d0*dT)) &
    +(w_c+w_h)/2
!print *, 'USING TANH(',(T1-Teff_jump)/(10d0*dT), &
!      ')=',&
!      TANH(T1-Teff_jump)/(10d0*dT), 'w_c=', &
!      w_c,&
!      'w_h=', w_h, 'w=',w
end if

end subroutine eval_tanh_intrp_vink_wind

```

```
end subroutine VvLNL_wind
```

A.2.7 Kudritzki *et al.*, de Jager *et al.*, Nugis & Lamers – KdJNL

```

subroutine KdJNL_wind(id, Lsurf, Msurf, Rsurf, Tsurf, w, ierr)
    use cstdlib_lib
    type (star_info), pointer :: s
    integer, intent(in) :: id
    real(dp), intent(in) :: Lsurf, Msurf, Rsurf, Tsurf ! surface values
    (cgs)

```

```

! NOTE: surface is outermost cell. not necessarily at photosphere.
! NOTE: don't assume that vars are set at this point.
! so if you want values other than those given as args,
! you should use values from s% xh(:, :) and s% xa(:, :) only.
! rather than things like s% Teff or s% lnT(:) which have not been
  set yet.
real(dp), intent(out) :: w ! wind in units of Msun/year (value is >=
  0)
integer, intent(out) :: ierr
real(dp) :: L1, M1, R1, T1, xsurf, Y, Z, etaHOT, etaCOOL, etaWR
real(dp) :: log10w
call get_star_ptr(id,s,ierr)
w = 0
ierr = 0
L1 = Lsurf
M1 = Msurf
R1 = Rsurf
T1 = Tsurf
etaHOT = s% x_ctrl(1)
s% Kudritzki_wind_eta = etaHOT
etaCOOL = s% x_ctrl(2)
etaWR = s% x_ctrl(3)
xsurf = s%xa(s% net_iso(iprot), 1)+ s%xa(s% net_iso(ih1), 1) !H
  mass fraction of the outermost cell, species 1 being neutrons
Y = s% xa(s%net_iso(ihe3),1) + s% xa(s%net_iso(ihe4),1)
Z = 1- xsurf - Y
if (xsurf .GE. 0.4d0) then
  if (T1 .LT. 15000d0) then
    print *, 'Lsurf=', L1, 'Tsurf=', Tsurf , 'Xsurf=', xsurf,
      'using: Nieuwenhuijzen', etaCOOL
    call eval_de_Jager_wind(w)
    w = w * etaCOOL
  else
    !hot star
    print *, 'Lsurf=', L1, 'Tsurf=', Tsurf , 'Xsurf=', xsurf,
      'using: Kudritzki', s% Kudritzki_wind_eta
    call eval_Kudritzki_wind(w)
  end if
else
  !the thing is a WR star
  print *, 'Lsurf=', L1, 'Tsurf=', Tsurf , 'Xsurf=', xsurf, 'using:
    Nugis & Lamers', etaWR
  w = 1d-11 * pow_cr(L1/Lsun,1.29d0) * pow_cr(Y,1.7d0) * sqrt(Z)
  w = w * etaWR
end if

contains

subroutine eval_Kudritzki_wind(w)
  use mod_kuma, only: kuma

```

```

real(dp), intent(inout) :: w
real(dp) :: xlogl, xlteff, stmass, xsurf, xmdfic, vinfkm
xlogl = log10_cr(L1/Lsun)
xlteff = log10_cr(T1)
    stmass = M1/Msun
xsurf = s% xa(2,1)
! output xmdfic = mass loss rate [Msun/yr]
! output vinfkm = v_infinity
call kuma(xlogl,xlteff,stmass,xsurf,xmdfic,vinfkm)
w = xmdfic*s% Kudritzki_wind_eta
! if (dbg) write(*,*) 'lg eval_Kudritzki_wind', log10_cr(w)

```

```
end subroutine eval_Kudritzki_wind
```

```

subroutine eval_de_Jager_wind(w)
! de Jager, C., Nieuwenhuijzen, H., & van der Hucht, K. A. 1988,
! A&AS, 72, 259.

```

\$

```

real(dp), intent(out) :: w
real(dp) :: log10w
include 'formats'
log10w = 1.769d0*log10_cr(L1/Lsun) - 1.676d0*log10_cr(T1) -
    8.158d0

$
w = exp10_cr(log10w)
end subroutine eval_de_Jager_wind

```

```
end subroutine KdJNL_wind
```

A.2.8 Kudritzki *et al.*, de Jager *et al.*, Hamann *et al.* – KdJH

```

subroutine KdJH_wind(id, Lsurf, Msurf, Rsurf, Tsurf, w, ierr)
    use cstdlib_lib
type (star_info), pointer :: s
integer, intent(in) :: id
real(dp), intent(in) :: Lsurf, Msurf, Rsurf, Tsurf ! surface values
    (cgs)
! NOTE: surface is outermost cell. not necessarily at photosphere.
! NOTE: don't assume that vars are set at this point.
! so if you want values other than those given as args,
! you should use values from s% xh(:, :) and s% xa(:, :) only.
! rather than things like s% Teff or s% lnT(:) which have not been
    set yet.
real(dp), intent(out) :: w ! wind in units of Msun/year (value is >=
    0)
integer, intent(out) :: ierr

```

```

real(dp) :: L1, M1, R1, T1, xsurf, etaWR, etaHOT, etaCOOL
real(dp) :: log10w
call get_star_ptr(id,s,ierr)
w = 0
ierr = 0
L1 = Lsurf
M1 = Msurf
R1 = Rsurf
T1 = Tsurf
etaHOT = s% x_ctrl(1)
s% Kudritzki_wind_eta = etaHOT
etaCOOL = s% x_ctrl(2)
etaWR = s% x_ctrl(3) !eta is the factor for WR winds
xsurf = s%xa(s% net_iso(iprot), 1)+ s%xa(s% net_iso(ih1), 1) !H
mass fraction of the outermost cell, species 1 being neutrons
if (xsurf .GE. 0.4d0) then
  if (T1 .LT. 15000d0) then
    print *, 'Lsurf=', L1, 'Tsurf=', Tsurf , 'Xsurf=', xsurf,
      'using: Nieuwenhuijzen', etaCOOL
    call eval_de_Jager_wind(w)
    w = w * etaCOOL
  else !hot star
    print *, 'Lsurf=', L1, 'Tsurf=', Tsurf , 'Xsurf=', xsurf,
      'using: Kudritzki', s% Kudritzki_wind_eta
    call eval_Kudritzki_wind(w)
  end if
else !the thing is a WR star
  print *, 'Lsurf=', L1, 'Tsurf=', Tsurf , 'Xsurf=', xsurf, 'using:
    manual Hamann', etaWR
  if (log10_cr(L1/Lsun) .LE. 4.5d0) then
    log10w = -35.8d0 + 6.8d0*log10_cr(L1/Lsun)
    w = exp10_cr(log10w)* etaWR
  else
    log10w = -11.95d0 + 1.5d0*log10_cr(L1/Lsun) - 2.85d0*(xsurf)
    w = exp10_cr(log10w)* etaWR
  end if
end if

contains

subroutine eval_de_Jager_wind(w)
  ! de Jager, C., Nieuwenhuijzen, H., & van der Hucht, K. A. 1988,
  A&AS, 72, 259.
  !
  real(dp), intent(out) :: w
  real(dp) :: log10w
  include 'formats'
  log10w = 1.769d0*log10_cr(L1/Lsun) - 1.676d0*log10_cr(T1) -

```

8.158d0

\$

w = exp10_cr(log10w)

end subroutine eval_de_Jager_wind

subroutine eval_Kudritzki_wind(w)

use mod_kuma, **only**: kuma

real(dp), **intent**(inout) :: w

real(dp) :: xlogl, xlteff, stmass, xsurf, xmdfic, vinfkm

xlogl = log10_cr(L1/Lsun)

xlteff = log10_cr(T1)

stmass = M1/Msun

xsurf = s% xa(2,1)

! output xmdfic = mass loss rate [Msun/yr]

! output vinfkm = v_infinity

call kuma(xlogl,xlteff,stmass,xsurf,xmdfic,vinfkm)

w = xmdfic*s% Kudritzki_wind_eta

! if (dbg) write(*,*) 'lg eval_Kudritzki_wind', log10_cr(w)

end subroutine eval_Kudritzki_wind

end subroutine KdJH_wind

A.2.9 Kudritzki *et al.*, Nieuwenhuijzen *et al.*, Nugis & Lamers – KNJNL

subroutine KNJNL_wind(id, Lsurf, Msurf, Rsurf, Tsurf, w, ierr)

use cstdlib_lib

type (star_info), **pointer** :: s

integer, **intent**(in) :: id

real(dp), **intent**(in) :: Lsurf, Msurf, Rsurf, Tsurf ! surface values
(cgs)

! NOTE: surface is outermost cell. not necessarily at photosphere.

! NOTE: don't assume that vars are set at this point.

! so if you want values other than those given as args,

! you should use values from s% xh(:,.) and s% xa(:,.) only.

! rather than things like s% Teff or s% lnT(.) which have not been
set yet.

real(dp), **intent**(out) :: w ! wind in units of Msun/year (value is >=
0)

integer, **intent**(out) :: ierr

real(dp) :: L1, M1, R1, T1, xsurf, Y, Z, etaHOT, etaCOOL, etaWR

real(dp) :: log10w

call get_star_ptr(id,s,ierr)

w = 0

ierr = 0

L1 = Lsurf

M1 = Msurf

```
R1 = Rsurf
T1 = Tsurf
etaHOT = s% x_ctrl(1)
s% Kudritzki_wind_eta = etaHOT
etaCOOL = s% x_ctrl(2)
etaWR = s% x_ctrl(3)
xsurf = s%xa(s% net_iso(iprot), 1)+ s%xa(s% net_iso(ih1), 1)
      !H mass fraction of the outermost cell, species 1 being
      neutrons
Y = s% xa(s%net_iso(ihe3),1) + s% xa(s%net_iso(ihe4),1)
Z = 1- xsurf - Y
if (xsurf .GE. 0.4d0) then
  if (T1 .LT. 15000d0) then
    print *, 'Lsurf=', L1, 'Tsurf=', Tsurf , 'Xsurf=',
      xsurf, 'using: Nieuwenhuijzen', etaCOOL
    call eval_Nieuwenhuijzen_wind(w)
    w = w * etaCOOL
  else !hot star
    print *, 'Lsurf=', L1, 'Tsurf=', Tsurf , 'Xsurf=',
      xsurf, 'using: Kudritzki', s% Kudritzki_wind_eta
    call eval_Kudritzki_wind(w)
  end if
else !the thing is a WR star
  print *, 'Lsurf=', L1, 'Tsurf=', Tsurf , 'Xsurf=',
    xsurf, 'using: Nugis & Lamers', etaWR
  w = 1d-11 * pow_cr(L1/Lsun,1.29d0) * pow_cr(Y,1.7d0)
    * sqrt(Z)
  w = w * etaWR
end if
```

contains

```
subroutine eval_Kudritzki_wind(w)
  use mod_kuma, only: kuma
  real(dp), intent(inout) :: w
  real(dp) :: xlogl, xlteff, stmass, xsurf, xmdfic, vinfkm
  xlogl = log10_cr(L1/Lsun)
  xlteff = log10_cr(T1)
  stmass = M1/Msun
  xsurf = s% xa(2,1)
  ! output xmdfic = mass loss rate [Msun/yr]
  ! output vinfkm = v_infinity
  call kuma(xlogl,xlteff,stmass,xsurf,xmdfic,vinfkm)
  w = xmdfic*s% Kudritzki_wind_eta
  ! if (dbg) write(*,*) 'lg eval_Kudritzki_wind', log10_cr(w)
```

end subroutine eval_Kudritzki_wind

```
subroutine eval_Nieuwenhuijzen_wind(w)
  ! Nieuwenhuijzen, H.; de Jager, C. 1990, A&A, 231, 134 (eqn 2)
  real(dp), intent(out) :: w
  real(dp) :: log10w
  include 'formats'
  log10w = -14.02d0 + 1.24d0*log10_cr(L1/Lsun) +
    0.16d0*log10_cr(M1/Msun) &
  + 0.81d0*log10_cr(R1/Rsun)
  w = exp10_cr(log10w)

end subroutine eval_Nieuwenhuijzen_wind
```

```
end subroutine KNJNL_wind
```

A.2.10 Kudritzki *et al.*, Nieuwenhuijzen *et al.*, Hamann *et al.* – KNJH

```
subroutine wellstein_langer_wind(id, Lsurf, Msurf, Rsurf, Tsurf, w,
  ierr)
  use cstdlib_lib
  type (star_info), pointer :: s
  integer, intent(in) :: id
  real(dp), intent(in) :: Lsurf, Msurf, Rsurf, Tsurf ! surface values
    (cgs)
  ! NOTE: surface is outermost cell. not necessarily at photosphere.
  ! NOTE: don't assume that vars are set at this point.
  ! so if you want values other than those given as args,
  ! you should use values from s% xh(:, :) and s% xa(:, :) only.
  ! rather than things like s% Teff or s% lnT(:) which have not been
    set yet.
  real(dp), intent(out) :: w ! wind in units of Msun/year (value is >=
    0)
  integer, intent(out) :: ierr
  real(dp) :: L1, M1, R1, T1, xsurf, Y, Z, etaWR, etaCOOL, etaHOT
  real(dp) :: log10w
  call get_star_ptr(id,s,ierr)
  w = 0
  ierr = 0
  L1 = Lsurf
  M1 = Msurf
  R1 = Rsurf
  T1 = Tsurf
  etaHOT = s% x_ctrl(1)
  s% Kudritzki_wind_eta = etaHOT
  etaCOOL = s% x_ctrl(2)
  etaWR = s% x_ctrl(3) !eta is the factor for WR winds
  xsurf = s%xa(s% net_iso(iprot), 1)+ s%xa(s% net_iso(ih1), 1) !H
    mass fraction of the outermost cell, species 1 being neutrons
```

```
Y = s% xa(s%net_iso(ihe3),1) + s% xa(s%net_iso(ihe4),1)
if (xsurf .GE. 0.4d0) then
  if (T1 .LT. 15000d0) then
    print *, 'Lsurf=', L1, 'Tsurf=', Tsurf , 'Xsurf=', xsurf,
      'using: Nieuwenhuijzen', etaCOOL
    call eval_Nieuwenhuijzen_wind(w)
    w = w * etaCOOL
  else
    print *, 'Lsurf=', L1, 'Tsurf=', Tsurf , 'Xsurf=', xsurf,
      'using: Kudritzki', s% Kudritzki_wind_eta
    call eval_Kudritzki_wind(w)
  end if
else
  !the thing is a WR star
  print *, 'Lsurf=', L1, 'Tsurf=', Tsurf , 'Xsurf=', xsurf, 'using:
    manual Hamann', etaWR
  if (log10_cr(L1/Lsun) .LE. 4.5d0) then
    log10w = -35.8d0 + 6.8d0*log10_cr(L1/Lsun)
    w = exp10_cr(log10w)* etaWR
  else
    log10w = -11.95d0 + 1.5d0*log10_cr(L1/Lsun) - 2.85d0*(xsurf)
    w = exp10_cr(log10w)* etaWR
  end if
end if
```

contains

```
subroutine eval_Nieuwenhuijzen_wind(w)
  ! Nieuwenhuijzen, H.; de Jager, C. 1990, A&A, 231, 134 (eqn 2)
  real(dp), intent(out) :: w
  real(dp) :: log10w
  include 'formats'
  log10w = -14.02d0 + 1.24d0*log10_cr(L1/Lsun) + &
    0.16d0*log10_cr(M1/Msun) + &
    0.81d0*log10_cr(R1/Rsun)
  w = exp10_cr(log10w)
  ! if (dbg) then
  !   write(*,1) 'Nieuwenhuijzen log10 wind', log10w
  ! end if
end subroutine eval_Nieuwenhuijzen_wind
subroutine eval_Kudritzki_wind(w)
  use mod_kuma, only: kuma
  real(dp), intent(inout) :: w
  real(dp) :: xlogl, xlteff, stmass, xsurf, xmdfic, vinfkm
  xlogl = log10_cr(L1/Lsun)
  xlteff = log10_cr(T1)
  stmass = M1/Msun
```

```
xsurf = s% xa(2,1)
! output xmdfic = mass loss rate [Msun/yr]
! output vinfkm = v_infinity
call kuma(xlogl,xlteff,stmass,xsurf,xmdfic,vinfkm)
w = xmdfic*s% Kudritzki_wind_eta
! if (dbg) write(*,*) 'lg eval_Kudritzki_wind', log10_cr(w)

end subroutine eval_Kudritzki_wind
end subroutine wellstein_langer_wind
```

A.2.11 Kudritzki *et al.*, van Loon *et al.*, Hamann *et al.* – KvLH

```
subroutine KvLH_wind(id, Lsurf, Msurf, Rsurf, Tsurf, w, ierr)
  use cstdlib_lib
  type (star_info), pointer :: s
  integer, intent(in) :: id
  real(dp), intent(in) :: Lsurf, Msurf, Rsurf, Tsurf ! surface values
    (cgs)
  ! NOTE: surface is outermost cell. not necessarily at photosphere.
  ! NOTE: don't assume that vars are set at this point.
  ! so if you want values other than those given as args,
  ! you should use values from s% xh(:,) and s% xa(:,) only.
  ! rather than things like s% Teff or s% lnT(:) which have not been
    set yet.
  real(dp), intent(out) :: w ! wind in units of Msun/year (value is >=
    0)
  integer, intent(out) :: ierr
  real(dp) :: L1, M1, R1, T1, xsurf, Y, Z, etaHOT, etaCOOL, etaWR
  real(dp) :: log10w
  call get_star_ptr(id,s,ierr)
  w = 0
  ierr = 0
  L1 = Lsurf
  M1 = Msurf
  R1 = Rsurf
  T1 = Tsurf
  etaHOT = s% x_ctrl(1)
  s% Kudritzki_wind_eta = etaHOT
  etaCOOL = s% x_ctrl(2)
  etaWR = s% x_ctrl(3)
  xsurf = s%xa(s% net_iso(iprot), 1)+ s%xa(s% net_iso(ih1), 1) !H
    mass fraction of the outermost cell, species 1 being neutrons
  Y = s% xa(s%net_iso(ihe3),1) + s% xa(s%net_iso(ihe4),1)
  Z = 1- xsurf - Y
  if (xsurf .GE. 0.4d0) then
    if (T1 .LT. 15000d0) then
      print *, 'Lsurf=', L1, 'Tsurf=', Tsurf , 'Xsurf=', xsurf,
        'using: van loon', etaCOOL
```

```
    call eval_van_Loon_wind(w)
    w = w * etaCOOL
else
    !hot star
    print *, 'Lsurf=', L1, 'Tsurf=', Tsurf, 'Xsurf=', xsurf,
        'using: Kudritzki', s% Kudritzki_wind_eta
    call eval_Kudritzki_wind(w)
end if
else
    !the thing is a WR star
    print *, 'Lsurf=', L1, 'Tsurf=', Tsurf, 'Xsurf=', xsurf, 'using:
        Hamman', etaWR
    if (log10_cr(L1/Lsun) .LE. 4.5d0) then
        log10w = -35.8d0 + 6.8d0*log10_cr(L1/Lsun)
        w = exp10_cr(log10w)* etaWR
    else
        log10w = -11.95d0 + 1.5d0*log10_cr(L1/Lsun) - 2.85d0*(xsurf)
        w = exp10_cr(log10w)* etaWR
    end if
end if

contains

subroutine eval_Kudritzki_wind(w)
    use mod_kuma, only: kuma
    real(dp), intent(inout) :: w
    real(dp) :: xlogl, xlteff, stmass, xsurf, xmdfic, vinfkm
    xlogl = log10_cr(L1/Lsun)
    xlteff = log10_cr(T1)
    stmass = M1/Msun
    xsurf = s% xa(2,1)
        ! output xmdfic = mass loss rate [Msun/yr]
        ! output vinfkm = v_infinity
    call kuma(xlogl,xlteff,stmass,xsurf,xmdfic,vinfkm)
    w = xmdfic*s% Kudritzki_wind_eta
        ! if (dbg) write(*,*) 'lg
        eval_Kudritzki_wind', log10_cr(w)

end subroutine eval_Kudritzki_wind

subroutine eval_van_Loon_wind(w)
        ! van Loon et al. 2005, A&A, 438, 273
    real(dp), intent(out) :: w
    real(dp) :: log10w
    include 'formats'
    log10w = -5.65d0 + 1.05*log10_cr(L1/(1d4*Lsun)) -
        6.3d0*log10_cr(T1/35d2)
    w = exp10_cr(log10w)
end subroutine eval_van_Loon_wind
```

```
end subroutine KvLH_wind
```

A.2.12 Kudritzki *et al.*, van Loon *et al.*, Nugis & Lamers – KvLNL

```
subroutine KvLNL_wind(id, Lsurf, Msurf, Rsurf, Tsurf, w, ierr)
  use cstdlib_lib
  type (star_info), pointer :: s
  integer, intent(in) :: id
  real(dp), intent(in) :: Lsurf, Msurf, Rsurf, Tsurf ! surface values
    (cgs)
  ! NOTE: surface is outermost cell. not necessarily at photosphere.
  ! NOTE: don't assume that vars are set at this point.
  ! so if you want values other than those given as args,
  ! you should use values from s% xh(:, :) and s% xa(:, :) only.
  ! rather than things like s% Teff or s% lnT(:) which have not been
    set yet.
  real(dp), intent(out) :: w ! wind in units of Msun/year (value is >=
    0)
  integer, intent(out) :: ierr
  real(dp) :: L1, M1, R1, T1, xsurf, Y, Z, etaHOT, etaCOOL, etaWR
  real(dp) :: log10w
  call get_star_ptr(id,s,ierr)
  w = 0
  ierr = 0
  L1 = Lsurf
  M1 = Msurf
  R1 = Rsurf
  T1 = Tsurf
  etaHOT = s% x_ctrl(1)
  s% Kudritzki_wind_eta = etaHOT
  etaCOOL = s% x_ctrl(2)
  etaWR = s% x_ctrl(3)
  xsurf = s%xa(s% net_iso(iprot), 1)+ s%xa(s% net_iso(ih1), 1) !H
    mass fraction of the outermost cell, species 1 being neutrons
  Y = s% xa(s%net_iso(ihe3),1) + s% xa(s%net_iso(ihe4),1)
  Z = 1 - Y - xsurf
  if (xsurf .GE. 0.4d0) then
    if (T1 .LT. 15000d0) then
      print *, 'Lsurf=', L1, 'Tsurf=', Tsurf , 'Xsurf=',
        xsurf, 'using: van loon', etaCOOL
      call eval_van_Loon_wind(w)
      w = w * etaCOOL
    else !hot star
      print *, 'Lsurf=', L1, 'Tsurf=', Tsurf , 'Xsurf=',
        xsurf, 'using: Kudritzki', s% Kudritzki_wind_eta
      call eval_Kudritzki_wind(w)
```

```
        end if
    else !the thing is a WR star
        print *, 'Lsurf=', L1, 'Tsurf=', Tsurf, 'Xsurf=',
            xsurf, 'using: Nugis & Lamers', etaWR
        w = 1d-11 * pow_cr(L1/Lsun,1.29d0) * pow_cr(Y,1.7d0)
            * sqrt(Z)
        w = w * etaWR
    end if

contains

subroutine eval_Kudritzki_wind(w)
    use mod_kuma, only: kuma
    real(dp), intent(inout) :: w
    real(dp) :: xlogl, xlteff, stmass, xsurf, xmdfic, vinfkm
    xlogl = log10_cr(L1/Lsun)
    xlteff = log10_cr(T1)
    stmass = M1/Msun
    xsurf = s% xa(2,1)
    ! output xmdfic = mass loss rate [Msun/yr]
    ! output vinfkm = v_infinity
    call kuma(xlogl,xlteff,stmass,xsurf,xmdfic,vinfkm)
    w = xmdfic*s% Kudritzki_wind_eta
    ! if (dbg) write(*,*) 'lg eval_Kudritzki_wind', log10_cr(w)

end subroutine eval_Kudritzki_wind

subroutine eval_van_Loon_wind(w)
    ! van Loon et al. 2005, A&A, 438, 273
    real(dp), intent(out) :: w
    real(dp) :: log10w
    include 'formats'
    log10w = -5.65d0 + 1.05*log10_cr(L1/(1d4*Lsun)) -
        6.3d0*log10_cr(T1/35d2)
    w = exp10_cr(log10w)
end subroutine eval_van_Loon_wind

end subroutine KvLNL_wind
```

A.2.13 Timestep Controls

```
integer function extras_finish_step(s, id, id_extra)
    use cstdlib_lib
    type (star_info), pointer :: s
    integer, intent(in) :: id, id_extra
    integer :: ierr, k
    real(dp) :: time, mdot_timescale, rad_diff_env,
```

```

rad_process_timescale, random_walk_num_step
real(dp) :: env_R, env_mfp, R_core, deltaR, env_M, M_core, tmp
real(dp) :: mean_rho, mean_op

call system_clock(time1,clock_rate)
time = double(time1-time0)/ clock_rate / 60 ! elapsed time in minutes
extras_finish_step = keep_going
call store_extra_info(s)
!Uncomment this to run on the Zwicky Cluster when the run time is
  limited to 12 hours
!if (time .GT. 715) then !715 min = 11h:55min
!  call output(id, ierr) !to save a photo
!  extras_finish_step = terminate
!  print *, 'Terminated because maximum time reached'
!end if
call get_star_ptr(id,s,ierr)
s% lxtra1 = false.
if (s% xa(s%net_iso(io16),s%nz) .LT. 0.04) then
  if (s%xa(s%net_iso(isi28), s%nz) .GT. 0.01) then !0
    depletion: X_o16 < 0.04 in the center and si28 has been
    produced
    extras_finish_step = terminate
    s%lxtra1 = true.
    if ((s%lxtra1_old .eqv. false.) .AND. (s% lxtra1 .eqv.
      true.) then
      call output(id, ierr) !to save a photo
    end if
  end if
end if

!control that the next timestep is smaller than the relevant
  timescales!

rad_process_timescale = 1d-8 !in sec = (avg mfp)/c =
  1/(opacity*rho*c) ~ 1/(0.1*c) ~ 1d-8

! find the envelope boundary
k = s% nz
do while (s% m(k) / Msun .LT. s% mass_conv_core) !in Msun
  k = k-1
end do
R_core = s% r(k) / Rsun !in Rsun
M_core = s% m(k) / Msun
R_core = max(R_core, s% he_core_radius) !in Rsun
M_core = max(M_core, s% he_core_mass) !in Msun
!he_core_radius is the radius where H drops below 0.01
!print *, "R_core=", R_core, s% he_core_radius

```

```
!print *, "M_core=", M_core
env_R = s% r(1) / Rsun - R_core !in Rsun
env_M = s% mstar /Msun - M_core !in Msun
env_M = env_M * Msun !in grams
! r(1) is the radius of the surface
!print *, "env_R", env_R
!print *, "should be 1", (env_R + R_core) / (s% r(1) / Rsun)
if (env_R /= 0) then
  k = s% nz
  do while ((s% r(k) / Rsun .LT. R_core))
    k = k-1 !move outward
  end do !at the end of the loop k is the index of the
        !first cell of the envelope
  !print *, s% r(k)
  mean_rho = 0
  mean_op = 0
  tmp = 0
  !print *, k, R_core
  do while (k .GE. 1)
    !print *, s% opacity(k)
    if (k == s% nz) then
      deltaR = s%r(k) !in cm
      tmp = tmp + deltaR / s% csound(k) !in sec
      ! evaluate the mean opacity and density in the envelope
      ! weighting by the radius
      mean_op = mean_op + s% opacity(k) * s% r(k)
      mean_rho = mean_rho + s% rho(k) * s% r(k)
    else !means we are not in the central cell
      deltaR = s% r(k) / Rsun - s% r(k+1) / Rsun
      deltaR = deltaR * Rsun !in cm
      !print *, deltaR, deltaR / s% csound(k)
      ! evaluate the mean opacity and density in the envelope
      ! weighting by the radius
      tmp = tmp + deltaR / s% csound(k) !in sec
      mean_op = mean_op + s% opacity(k) * ((s% r(k)/ Rsun &
        - s% r(k+1) / Rsun )) * Rsun
      mean_rho = mean_rho + s% rho(k) * ((s% r(k)/ Rsun &
        - s% r(k+1) / Rsun )) * Rsun
    end if
    k = k-1
    !print *, tmp
  end do
  mean_rho = mean_rho / (env_R * Rsun) !normalization
  mean_op = mean_op / (env_R * Rsun) !normalization
  !print *, "mean_rho_cgs =", mean_rho, "mean_op_cgs =", mean_op
  env_R = env_R * Rsun !in cm
  env_mfp = 1d0 / (mean_rho * mean_op) !in cm
  random_walk_num_step = (env_R/env_mfp)**2 !pure number
```

```

!print *, "random_walk_num_step=", random_walk_num_step
rad_diff_env = rad_process_timescale * random_walk_num_step !in
    sec
!print *, 'env_mfp=', env_mfp
!print *, 'rad_diff_env=', rad_diff_env
else !there isn't an envelope anymore (unlikely), set this to
    infinite so that they don't limit the timestep
rad_diff_env = 1d99
!print *, "where is my envelope?"
end if

mdot_timescale = s% mstar / (1d0 * abs(s% mstar_dot)) ! in sec
!print *, "proposed dt_next [yr]", s% dt_next / secyer
!dt_next must be in seconds, but for numerical stability the min
! must be evaluated with stuff in years
s% dt_next = min(s% dt_next / secyer , s% kh_timescale, &
    mdot_timescale / secyer, &
    s% nuc_timescale, &
    s% chem_timescale , &
    rad_diff_env / secyer)

!print *, s% dt_next
s% dt_next = s% dt_next * secyer
if (min(s% dt_next / secyer , s% kh_timescale, &
    mdot_timescale / secyer, &
    s% nuc_timescale, &
    s% chem_timescale , &
    rad_diff_env / secyer) &
    == s% kh_timescale ) then
    print *, "dt_next limited by kh_timescale"
else if (min(s% dt_next / secyer , s% kh_timescale, &
    mdot_timescale / secyer, &
    s% nuc_timescale, &
    s% chem_timescale , &
    rad_diff_env / secyer) &
    == mdot_timescale / secyer ) then
    print *, "dt_next limited by mdot_timescale"
else if (min(s% dt_next / secyer , s% kh_timescale, &
    mdot_timescale / secyer, &
    s% nuc_timescale, &
    s% chem_timescale , &
    rad_diff_env / secyer) &
    == s% dynamic_timescale / secyer ) then
    print *, "dt_next limited by dynamic_timescale"
else if (min(s% dt_next / secyer , s% kh_timescale, &
    mdot_timescale / secyer, &
    s% nuc_timescale, &
    s% chem_timescale , &
    rad_diff_env / secyer) &

```

```
        == s% nuc_timescale ) then
    print *, "dt_next limited by nuc_timescale"
else if (min(s% dt_next / secyer , s% kh_timescale, &
            mdot_timescale / secyer, &
            s% nuc_timescale, &
            s% chem_timescale , &
            rad_diff_env / secyer) &
        == s% chem_timescale * secyer) then
    print *, "dt_next limited by chem_timescale"
else if (min(s% dt_next / secyer , s% kh_timescale, &
            mdot_timescale / secyer, &
            s% nuc_timescale, &
            s% chem_timescale , &
            rad_diff_env / secyer) &
        == rad_diff_env / secyer) then
    print *, "dt_next limited by rad_diff_env"
end if

!print *, "next timestep in years:", s% dt_next / secyer
!print *, "log_dt_next", log10_cr(s% dt_next / secyer)

contains

subroutine output(id, ierr)
    ! use star_utils, only: get_name_for_restart_file
use cstdlib_lib
interface
    subroutine save_restart_info(iounit, id, ierr)
        integer, intent(in) :: iounit
        integer, intent(in) :: id
        integer, intent(out) :: ierr
    end subroutine save_restart_info
end interface
integer, intent(in) :: id
integer, intent(out) :: ierr

character (len=256) :: filename, num_str, fstring, modname
type (star_info), pointer :: s
integer :: num_digits

call get_star_ptr(id, s, ierr)
if (ierr /= 0) return
num_str = 'photo_11h55min'
if (ierr /= 0) return
if ((s% xa(8,s%nz) .LT. 0.04) .AND. (s%xa(11, s%nz) .GT. 0.01))
    then !0 depletion: X_o16 < 0.04 in the center and si28 has
        been produced
    num_str = 'photo_0_depl'
```

```
    call star_write_model(id, '0_depl.mod' , ierr)
end if

filename = trim(s% photo_directory) // '/' // trim(num_str)
s% need_to_save_profiles_now = .true.
s% save_profiles_model_priority = 10
call output_to_file(filename, id, ierr)

end subroutine output

subroutine output_to_file(filename, id, ierr)
use utils_lib, only:alloc_iounit, free_iounit
use model_out, only:output_star_model
character (len=*) :: filename
integer, intent(in) :: id
integer, intent(out) :: ierr

integer :: iounit, k
type (star_info), pointer :: s
character(len=256) :: iomsg

include 'formats'

call get_star_ptr(id, s, ierr)
if (ierr /= 0) return

iounit = alloc_iounit(ierr); if (ierr /= 0) return
open(iounit, file=trim(filename), action='write', &
status='replace', iostat=ierr, iomsg=iomsg, form='unformatted')
if (ierr == 0) then
    s% most_recent_photo_name = trim(filename)
    call output_star_model(s, iounit)
    close(iounit)
else
    write(*,*) trim(iomsg)
endif
call free_iounit(iounit)

end subroutine output_to_file
end function extras_finish_step
```

A.2.14 run_star_extras.f for the Simplified Envelope Shedding Events

Stopping Criterion

The run_star_extras.f used for the study of impulsive mass loss events do not use customized routines different from those of the wind study. The only exception is the stopping criterion for the MCE serie, c.f Tab. 2.3. MESA does not include settings to control the

run based on the extent of the convective envelope, therefore, the MCE condition must be checked using the `extras_check_model` routine in the `run_star_extras.f`.

```
integer function extras_check_model(s, id, id_extra)
  type (star_info), pointer :: s
  integer, intent(in) :: id, id_extra
  integer :: k, j
  real(dp) :: relv,surf_v,loc_v
  logical set_surf
  set_surf = .true.
  extras_check_model = keep_going
  if(s%center_h1 == 0.0) then
    do k=1, 150
      surf_v = surf_v + s% conv_vel(k)
    end do
    surf_v=surf_v/150 !surf_v is the average convective velocity in
      the outermost 150 cells
    do k=2, s% nz
      if ((s% conv_vel(k)==0.0).and.(s% mixing_type(k-1)==3)) then
        !k is the cell of the lower boundary of a convective region
        !mixing type = 3 => semiconvection
        do j=1,150
          loc_v= loc_v+ s% conv_vel(k-j)
        end do
        loc_v = loc_v/150
        relv = abs((loc_v - surf_v)/surf_v)
        if (relv.LT.0.1) then
          extras_check_model=terminate
          s% termination_code = t_xtra1
          termination_code_str(t_xtra1)="made convective envelope"
          return
        end if
      end if
    end do
  end if
  ! by default, indicate where (in the code) MESA terminated
  if (extras_check_model == terminate) s% termination_code =
    t_extras_check_model
end function extras_check_model
```

Mass Loss and Timestep Controls

The mass loss routine is the one for the Vink *et al.*, de Jager *et al.*, Nugis & Lamers – Vd-JNL, presented in §A.2.1. The customized timestep control is enforced using the routine of §A.2.13. The mass stripping is done by saving a model at the desired stripping point and restarting it using the built-in MESA routine `adjust_mass`, which is controlled using the `relax_initial_mass` in the `inlist` used when restarting the simulation – see §A.1.2.

APPENDIX B

From a naive use of MESA toward physically sound Results

We have a habit in writing articles published in scientific journals to make the work as finished as possible, to cover up all the tracks, to not worry about the blind alleys or describe how you had the wrong idea first, and so on. So there really isn't any place to publish, in a dignified manner, what you actually did in order to get to do the work.

[R. Feynman, Nobel Lecture, December 11, 1965]

B.1 Warning to the Naive MESA User

MESA is a very well designed code that rarely crashes, forcing the new user to look for the problem(s) to fix. Most of the time, MESA does *not* crash, but instead it is able to find a solution, which – too often – may not be physically sound. The burden of understanding the solutions MESA finds, and more importantly, of determining how realistic these are, is left to the user. “MESA is a tool, not a theory”.

The aim of this appendix is to summarize part of the work I did to improve my results. I focus on issues found in the evolution of non-rotating, solar metallicity, massive stars of $M_{\text{ZAMS}} \leq 40M_{\odot}$, however the careful analysis of the results is needed for any problem the code can handle. The take-home point is *to not believe any result the MESA code can and will produce, but analyze it very carefully each time*. In my experience, MESA often converges to unphysical solutions rather than crashing when something is wrong. Because of the non-linearity of the equations for a stellar structure, a small inaccuracy in an aspect believed to be secondary, might have rather significant effects. The aspect causing unexpected and undesired behaviors may be very hard to individuate. Most of the work is left to do once a numerical model has been found for the problem considered.

In §B.2 I describe the issue of unphysical oscillations of the stellar surface (see also §2.3), which can be solved by using a more realistic determination of the outer boundary condition. This provides an example of the effects that an inaccurate treatment of an aspect initially thought to be secondary can have on the results. In §B.3, I describe the ongoing work to improve the resolution of the stellar cores during late burning stages, emphasizing the need for large nuclear reaction networks.

B.2 Surface Oscillations

When using the default parameters of MESA for massive stars (see `$MESA_DIR/star/inlist_massive_defaults`), I find unphysical oscillations of the surface L , R , and T_{eff} during the late evolutionary stages of stars with $M_{\text{ZAMS}} \geq 20M_{\odot}$ (as mentioned briefly in §2.3). These oscillations in turn cause rapid and unphysical variations of the \dot{M} , which is evaluated as a function of some of the variables¹ characterizing the stellar surface.

Figure B.1 shows an example of such oscillations for the luminosity of $M_{\text{ZAMS}} = 30M_{\odot}$ models computed using the Vink *et al.*, Nieuwenhuijzen *et al.*, and Hamann *et al.* wind mass loss scheme with different efficiencies.

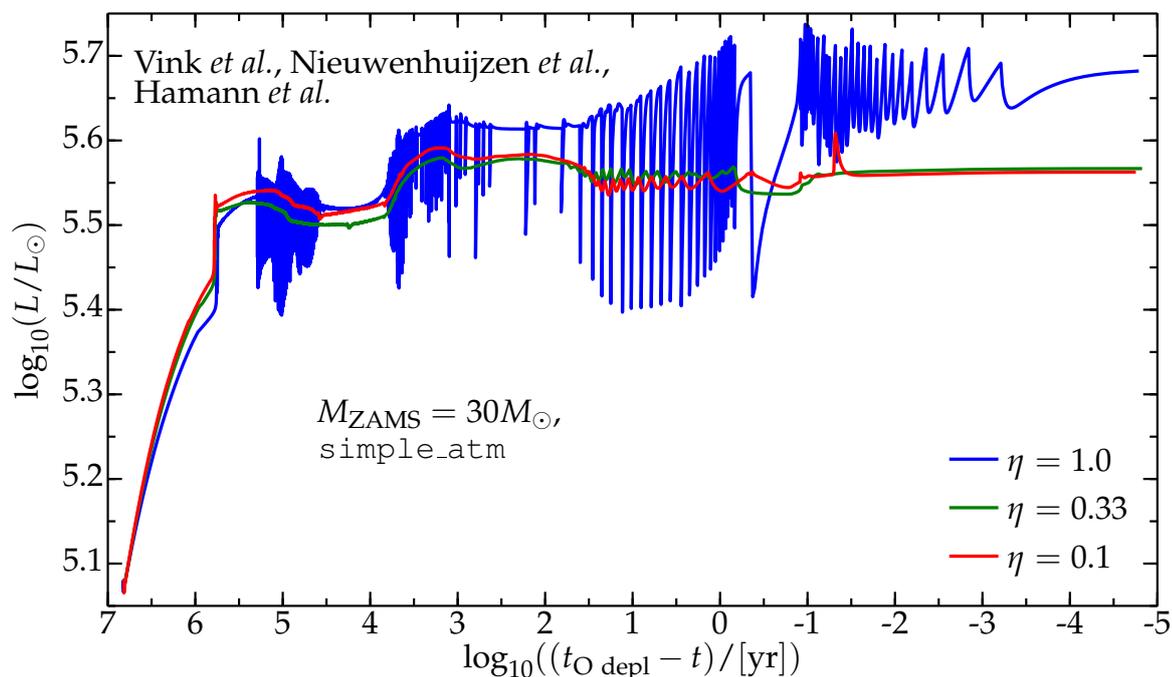


Figure B.1: Luminosity for $30M_{\odot}$ models computed with the default `simple_atm` boundary condition. The oscillation shown are an unphysical, numerical effect.

These oscillations are most likely caused by the interaction of the MLT++ routine (see §2.1.3) with the default outer boundary condition `simple_atm`. They cannot be a real physical effect because of their extreme amplitude (note the logarithmic scale on the y-axis of Fig. B.1), and their period, which does not match any of the relevant physical timescales of the star.

Lower mass models (e.g. $M_{\text{ZAMS}} = \{12, 15\}M_{\odot}$) that do not need MLT++, do not show the oscillations. Moreover, by changing the boundary condition they can be damped and, and if $M_{\text{ZAMS}} \leq 30M_{\odot}$ eliminated, see Fig. B.2. In particular, the boundary condition I used is the implementation of an Eddington-grey atmosphere, which is also more physically accurate than `simple_atm`, see §2.3 and the `atm` module in `$MESA_DIR/atm`.

The presence of these oscillations also causes a major slowdown in the computation: the code often struggles to find a solution, the timestep taken get increasingly smaller during the

¹Which specific variables are used depends on the mass loss algorithm, see §1.4.

evolution², and MESA often needs to go a few step back to restart with a smaller timestep. For example, the $\eta = 1.0$ model in Fig. B.2 (blue curve) reaches oxygen depletion after 252460 steps, of which 3032 are taken twice (i.e. they are MESA retry steps, re-taken with a lower timestep because the first solution found exceeds some limiting controls), and 1124 times MESA needs to come back by two steps and restart from there with a lower timestep (i.e. 1124 are MESA backups). For comparison, the corresponding model computed with the setup described in §2.3 (i.e. using the Eddington-grey boundary condition and the local refinements of the mesh instead of the `simple.atm` and no refinements; see also §A.2.4 and §A.2.13) reaches oxygen depletion in 20876 steps, of which only 38 are retry steps, and 13 are backup steps. The luminosity of this model is the blue curve in Fig. B.2.

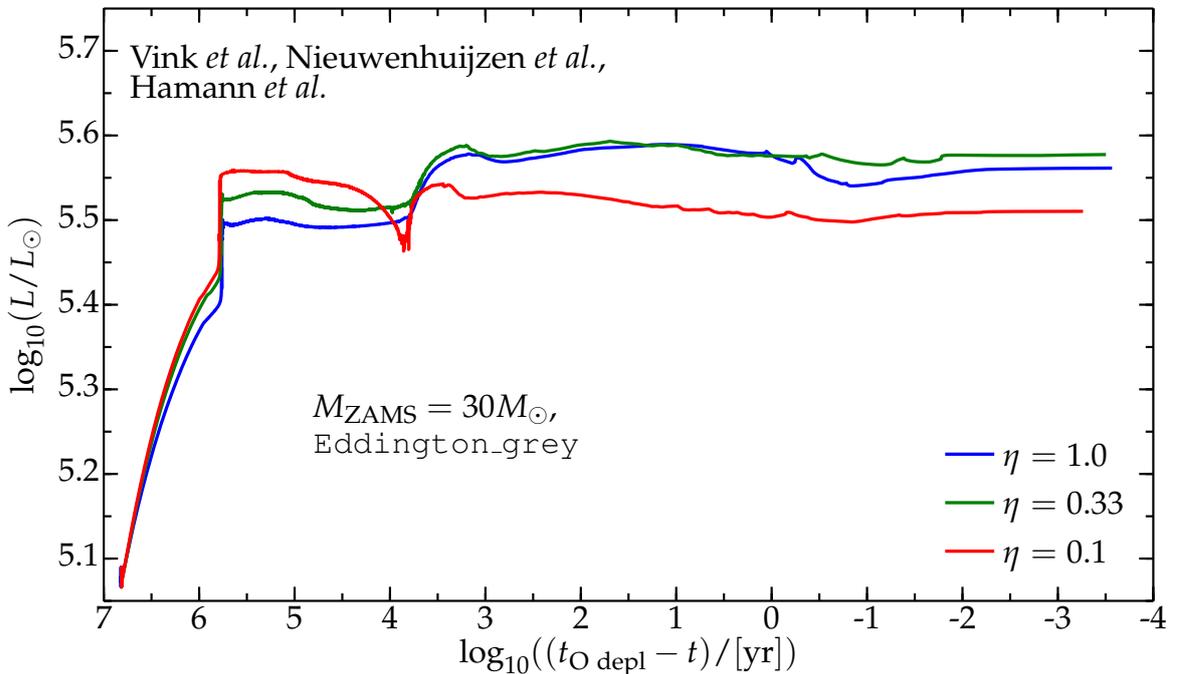


Figure B.2: Luminosity for $30M_{\odot}$ models computed with the `Eddington-grey` outer boundary condition, and with the “local mesh refinements” described in §2.3. The use of a better set of parameters, namely a better determination of the outer boundary condition, can eliminate non-physical features seen with the default set of parameters, cf. Fig. B.1.

This example illustrates that the default parameters suggested (see files in `$MESA_DIR/star/defaults`) are not always suitable for the study of a particular problem, and they may cause unphysical effects that are sometimes very hard to detect. The careful analysis of the parameters used, also of those that are not of primary interest for the target research topic, can often help in the detection and solution of (potential) numerical issues. Some unphysical effects may be visible only in plots of particular planes and using appropriate scales (e.g. (L,t) , using a logarithmic scale for t for this example), otherwise they may remain hidden. The bottom line is to never assume that the MESA default parameter are suitable.

²If $M_{\text{ZAMS}} \geq 35M_{\odot}$, they can be so small that it is impossible compute further than helium depletion.

B.3 On the Resolution at high Temperatures

B.3.1 Introduction to the Problem

The modeling of the late burning stages in the core of massive stars is particularly challenging, see also §1.3.2. The number of active isotopes in the nuclear processes and the burning rates increase with the temperature. Often forward and backward reactions nearly balance each others, justifying quasi-equilibrium approximations for the very late burning stage, e.g. Quasi Statistical Equilibrium (QSE) during silicon burning (see §1.3.2, and, e.g. [3, 11, 12]).

The MESA approach for silicon burning does not rely on the quasi-equilibrium approximation, since the code is designed to solve the fully coupled set of equations for the structure and composition of the star. Instead, the equation for the abundances (cf. Eq. 2.9) is solved dividing the timestep in many sub-steps and going to higher orders with the Bulirsch-Stoer algorithm [103]. This means that MESA does not assume that forward and backward reactions cancel out, and all the active isotopes which can possibly react must be followed individually in the nuclear reaction network³. This implies a major slowdown of the computation for these stages. The results of this different approach, not relying on quasi-equilibrium approximations, is still in a testing phase.

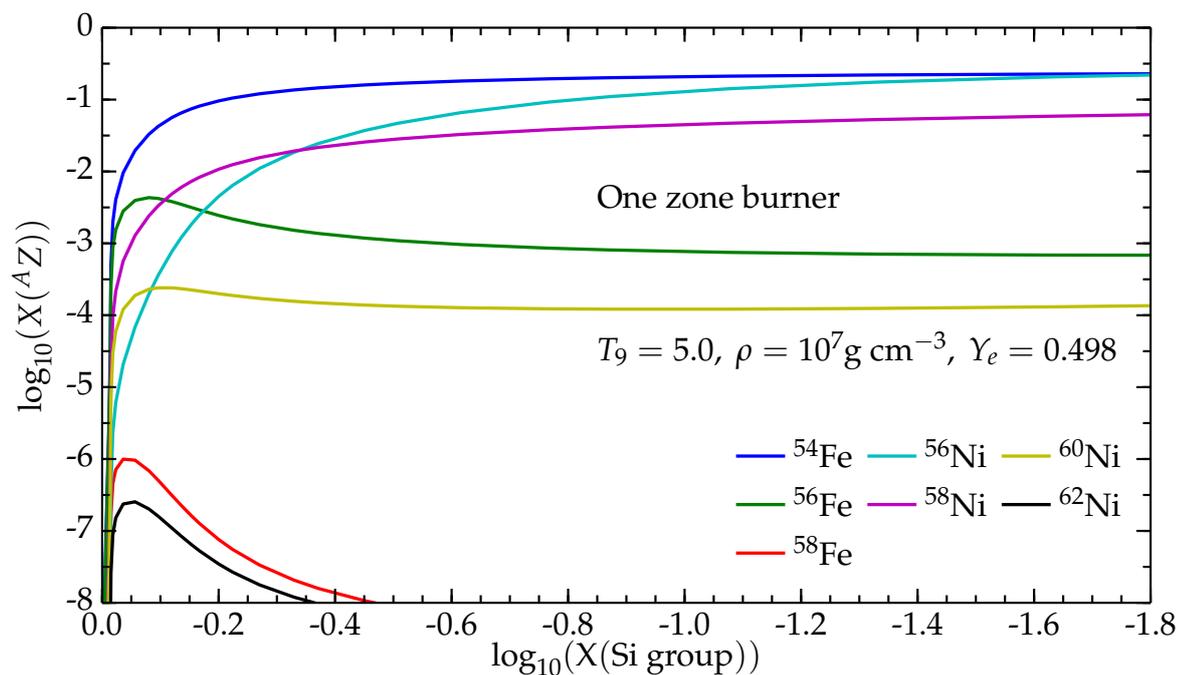


Figure B.3: One zone nuclear burning calculation with a 299-isotope non-QSE MESA network at constant T and ρ . Compare this figure to Fig. 3a of [12].

Figure B.3 shows the evolution of the abundance of some heavy nuclei as a function of the abundance of elements of the silicon group, $24 \leq A \leq 44$, [12], computed with a 299-isotope network. The computation is carried out following the work of Hix *et al.*, in a single zone model at constant temperature and density. The results of the large, non-QSE MESA network are in remarkable agreement with the QSE-network results of Hix *et al.*

³Of course, the conditions in the core of a massive star cannot be reproduced on Earth, therefore, the number of active isotopes (which may be very short-lived) must be determined theoretically.

Moreover, the details of neutrino cooling and convective mixing (including overshooting) are paramount to obtain realistic core structures – which, in turn, are paramount for the SN explosion mechanism studies. These rely on stellar evolution calculations for the initial conditions. The proper resolution of these phenomena, both in the spatial and temporal domain, is a difficult problem. Variations can be very fast, requiring timesteps that can reach ~ 1 second of physical time during silicon burning. The spatial grid used must be very fine to be sure of not under-resolving temperature gradients that regulate the convective mixing.

B.3.2 Need for a large Nuclear Reaction Network

The definition of “large” depends on the precise burning stage considered: realistic simulations of carbon, neon, oxygen or silicon burning require different minimum numbers of isotopes. This even if the details of the nucleosynthesis are not of direct concern for the research carried out: a large nuclear network is needed to obtain a realistic structure of the core – even if the details of the composition are not of primary interest.

There are two main reasons why large nuclear reaction networks are needed in the simulation of late burning stages. The first is to have a realistic energy generation rate, and the second is to obtain a realistic density structure, and particularly a realistic value of the effective Chandrasekhar mass.

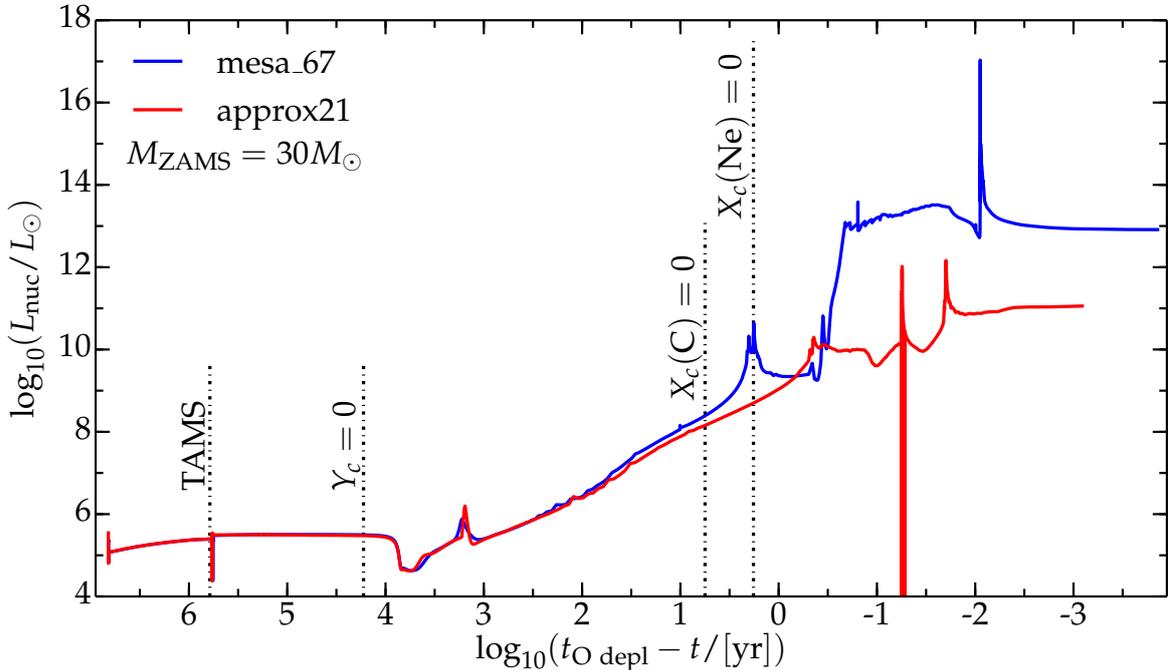


Figure B.4: Nuclear burning power in a $M_{\text{ZAMS}} = 30M_{\odot}$ model. The vertical dot dashed lines indicate approximately the end of hydrogen, helium, carbon and neon burning, respectively. The model is stopped at oxygen depletion, see Eq. 2.21.

Figure B.4 shows the power from nuclear reactions (excluding neutrinos),

$$L_{\text{nuc}} = \int_0^{M_{\text{tot}}} \varepsilon_{\text{nuc}} dm , \quad (\text{B.1})$$

in a non-rotating, solar metallicity, $M_{\text{ZAMS}} = 30M_{\odot}$ model⁴, with a 21-isotope nuclear network (`approx21.net`, red curve), and with a 67-isotope nuclear network (`mesa.67.net`, blue curve). The agreement between the two network is generally satisfying roughly until carbon depletion, however, the inclusion of more isotopes creates a large difference in the power released during neon burning, and later on during oxygen burning. Note that nothing ensures that the blue curve (i.e. the 67-isotope network) is itself realistic, even if it is generally accepted that ~ 50 isotopes are sufficient for obtaining the proper energy release from oxygen burning (Arnett, private communication).

The hypothesis of quasi-statistical equilibrium, commonly adopted for silicon burning, does not hold yet at carbon depletion. The approach⁵ of switching from a small nuclear reaction network to a large (QSE) network at oxygen depletion (e.g. [14, 18, 19]) can and should be improved, for example, by switching to progressively larger networks during and post-carbon burning.

The second reason to use a large nuclear reaction network, is to have a realistic density structure and effective Chandrasekhar mass of the iron core formed at the end of the evolution. This does not concern the simulations presented in §3, since these terminates at oxygen depletion, long before the formation of the iron core. However, this is relevant for the simulations presented in §4.

The isotopes present in the core of a massive star determine the fine details of the density structure, since the EOS depends on the mean molecular weight. Moreover, the neutronization of matter can be treated self-consistently only if the nuclear network adopted includes the many short-lived, neutron-rich isotopes that can form. Therefore the central Y_e can vary significantly changing the isotopes included in the nuclear reaction network, and this is of paramount importance for the determination of the effective Chandrasekhar mass, see Eq. 1.7.

For instance, `approx21.net` does not include the many neutron rich isotopes involved in silicon burning, and treats neutronization with a single compound reaction (Eq. 2.23), see §2.4. This produces a predetermined central $Y_e(r = 0) \equiv Y_e(^{56}\text{Cr}) = 0.428$ at the onset of core collapse. Note that this is the only value that `approx21.net` can yield⁶. Calculations with larger networks including hundreds of active isotopes may and do yield different (of a few percent) Y_e values. For example, a $M_{\text{ZAMS}} = 25M_{\odot}$, solar metallicity star computed with `approx21.net` until oxygen depletion, and then restarted from there with the 201-isotope network `mesa.201.net` reaches the onset of core collapse with $Y_e(r = 0) \simeq 0.423$. Although the difference between the `approx21.net` and the larger nuclear reaction network Y_e values may seem small, it is extremely relevant for the determination of the effective Chandrasekhar mass (Eq. 1.7), the compactness parameter (Eq. 3.1) and therefore “explosibility” of the star.

⁴This model uses the Vink *et al.*, de Jager *et al.*, and Nugis & Lamers (VdJNL) wind scheme with $\eta = 1.0$.

⁵This approach is also the original motivation to stop the simulations carried out in this work at oxygen depletion.

⁶MESA includes also modified version of `approx21.net`, that substitute the isotope produced by the compound electron-capture reaction, ^{56}Cr , with other isotopes, possibly of other elements. This allows to obtain different (but still pre-determined and fixed) values of the central Y_e at the onset of core collapse. For more details, see `$MESA_DIR/data/net_datanets`.

B.3.3 Spatial Resolution of the Core

Because of the high sensitivity of the core structure to the details of the physics included in the simulations, it is hard to find results that are robust against variations of the spatial and/or temporal discretization.

In §3.1.1, I present the compactness parameter at oxygen depletion, $\xi_{2.5}^{\text{O depl}}$, as an example of results that cannot yet be trusted. This because of their dependence on the numerical discretization used in the simulations. The dependence is of course not physical (nature does not need to discretize space and time to produce stars, as computers do), and must be eliminated to discuss the physics determining $\xi_{2.5}^{\text{O depl}}$.

Together with Fig. 3.8, Fig. B.5 illustrates the problem found (in the worse case scenario). Any variation of the parameters used for the simulation (e.g. nuclear reaction network, resolution coefficient `mesh_delta_coeff`, `timestep`, etc...) causes a variation of the spatial discretization, and consequently of the gradients in the stellar structure. Such variation can either be a direct (e.g. when I modify `mesh_delta_coeff`) or an indirect consequence (e.g. when the nuclear reaction network is changed, the amount of energy released and the location of the energy release may vary slightly, causing a displacement of the finer portions of the mesh) of the parameters choice.

The agreement between the evolutionary tracks of Fig. B.5 is satisfying until $T_c \simeq 1.3 \times 10^9$ K, i.e. until after MESA starts to artificially damp the mass loss (see §2.3). This ensures

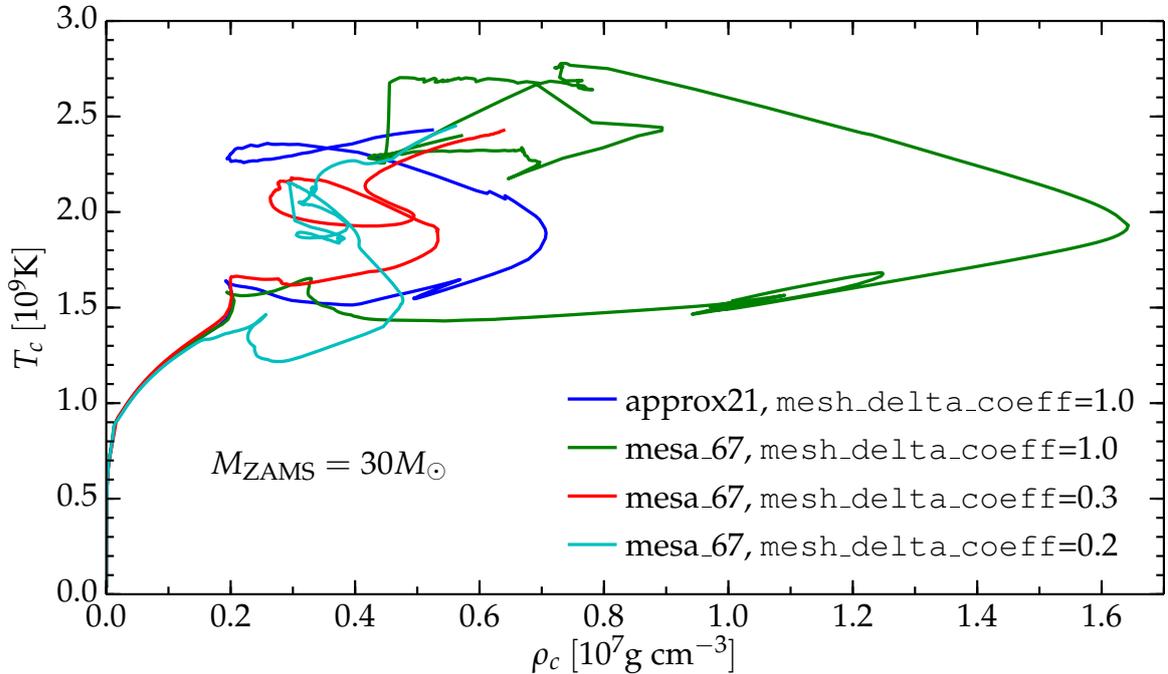


Figure B.5: Evolutionary tracks on the (ρ_c, T_c) plane for $M_{\text{ZAMS}} = 30M_{\odot}$ stars computed with the Vink *et al.*, de Jager *et al.*, Nugis & Lamers mass loss rate combination and $\eta = 1.0$. This is one of the worst cases computed among the many tests carried out. The blue curve corresponds to the $M_{\text{ZAMS}} = 30M_{\odot}$ model computed with the above algorithm presented in §3. To compute the green curve I changed the nuclear network to `mesa_67.net`. The red and cyan curves are computed with the same parameters as the green curve, except for the resolution, which is roughly 3 and 5 times higher, respectively.

that the results shown in §3 are not influenced by the poorly resolved core structure, except the values of the compactness parameter, as already underlined in §3.1.1. At very high temperatures, the evolutionary tracks depart from each other in an apparently chaotic way. The amplitude of these deviations is striking.

In principle, these non-converging deviations would be easy to eliminate, by forbidding any re-mesh procedure when $T \gtrsim 1.3 \times 10^9$ K somewhere in the star. However, this would not be a solution: given the sensitivity of the results to variations of some parameters, nothing assures that one of the curves in Fig. B.5 corresponds to a realistic evolutionary track. The structure of these simulated cores is *not* robust.

Moreover, this provides another example of untrustworthy results obtained without any crash and/or warning from the MESA code. If I had not tried to vary the resolution, and if I had not investigated the evolutionary tracks on the (ρ_c, T_c) plane, I would not have found this problem. Even if resolution studies are extremely time consuming, they are strongly advised for any problem analyzed.

The reason for the departure of the curves in Fig. B.5 is not known yet, and work to understand it and fix the problem is still in progress. The stubborn tightening of the local mesh refinements described in §2.3 has proven not to be sufficient to solve the problem.

Bibliography

- [1] N. Langer. *Ann. Rev. Astron. Astroph.*, **50**, 107, 2012.
- [2] Z. Haiman and A. Loeb. *Astrophys. J.*, **483**, 21, 1997.
- [3] J. José and C. Iliadis. *Reports on Progress in Physics*, **74**, 096901, 2011.
- [4] M. Arnould and K. Takahashi. *Reports on Progress in Physics*, **62**, 395, 1999.
- [5] H. J. G. L. M. Lamers. Mass Loss and Evolution of Massive Stars. In G. Pugliese, A. de Koter, and M. Wijnburg, editors, *370 Years of Astronomy in Utrecht*, volume 470 of *Astronomical Society of the Pacific Conference Series*, page 97, 2013.
- [6] J. Puls, J. O. Sundqvist, and N. Markova. *ArXiv:1409.3582*, 2014.
- [7] N. Smith. *Ann. Rev. Astron. Astroph.*, **52**, 487, 2014.
- [8] A. Heger, S. E. Woosley, G. Martínez-Pinedo, and K. Langanke. *Astrophys. J.*, **560**, 307, 2001.
- [9] W. D. Arnett. *Astrophys. J. Supp. Ser.*, **35**, 145, 1977.
- [10] I. R. Seitenzahl, D. M. Townsley, F. Peng, and J. W. Truran. *Atomic Data and Nuclear Data Tables*, **95**, 96, 2009.
- [11] W. R. Hix, S. T. Parete-Koon, C. Freiburghaus, and F.-K. Thielemann. *Astrophys. J.*, **667**, 476, 2007.
- [12] W. R. Hix and F.-K. Thielemann. *Astrophys. J.*, **460**, 869, 1996.
- [13] H.-T. Janka, K. Langanke, A. Marek, G. Martínez-Pinedo, and B. Müller. *Phys. Rep.*, **442**, 38, 2007.
- [14] S. E. Woosley, A. Heger, and T. A. Weaver. *Reviews of Modern Physics*, **74**, 1015, 2002.
- [15] E. O'Connor and C. D. Ott. *Astrophys. J.*, **762**, 126, 2013.
- [16] M. Ugliano, H.-T. Janka, A. Marek, and A. Arcones. *Astrophys. J.*, **757**, 69, 2012.
- [17] A. Heger, C. L. Fryer, S. E. Woosley, N. Langer, and D. H. Hartmann. *Astrophys. J.*, **591**, 288, 2003.
- [18] S. E. Woosley and A. Heger. *Phys. Rep.*, **442**, 269, 2007.
- [19] T. Sukhbold and S. E. Woosley. *Astrophys. J.*, **783**, 10, 2014.
- [20] H. Sana, S. E. de Mink, A. de Koter, N. Langer, C. J. Evans, M. Gieles, E. Gosset, R. G. Izzard, J.-B. Le Bouquin, and F. R. N. Schneider. *Science*, **337**, 444, 2012.
- [21] R. Kippenhahn, A. Weigert, and A. Weiss. *Stellar Structure and Evolution*. Springer-Verlag, 2013.
- [22] V. Castellani. *Astrofisica Stellare*. Zanichelli, 1985.
- [23] J. P. Cox and T. R. Giuli. *Principles of Stellar Structure - Vol. I & II*. Gordon & Breach, 1968.
- [24] S. N. Shore. *The Tapestry of Modern Astrophysics*. Wiley Interscience, 2003.
- [25] J. S. Vink, A. de Koter, and H. J. G. L. M. Lamers. *Astron. Astrophys.*, **369**, 574, 2001.
- [26] C. E. Rolfs and W. S. Rodney. *Cauldrons in the Cosmos*. The University of Chicago

BIBLIOGRAPHY

- Press, 1988.
- [27] M. B. Aufderheide. *Astrophys. J.*, **411**, 813, 1993.
- [28] J. J. Walmswell, C. A. Tout, and J. J. Eldridge. *Mon. Not. Roy. Astron. Soc.*, **447**, 2951, 2015.
- [29] S. J. Smartt, J. J. Eldridge, R. M. Crockett, and J. R. Maund. *Mon. Not. Roy. Astron. Soc.*, **395**, 1409–1437, 2009.
- [30] S. Horiuchi, K. Nakamura, T. Takiwaki, K. Kotake, and M. Tanaka. *Mon. Not. Roy. Astron. Soc.*, **445**, L99–L103, 2014.
- [31] A. Heger, S. E. Woosley, and H. C. Spruit. *Astrophys. J.*, **626**, 350, 2005.
- [32] E. O'Connor and C. D. Ott. *Classical and Quantum Gravity*, **27(11)**, 114103, 2010.
- [33] P. Goldreich and S. V. Weber. *Astrophys. J.*, **238**, 991, 1980.
- [34] A. Yahil and J. M. Lattimer. Supernovae for pedestrians. In M. J. Rees and R. J. Stoneham, editors, *NATO Advanced Science Institutes (ASI) Series C*, volume 90 of *NATO Advanced Science Institutes (ASI) Series C*, pages 53–70, 1982.
- [35] N. Smith and W. D. Arnett. *Astrophys. J.*, **785**, 82, 2014.
- [36] D. Clausen, A. L. Piro, and C. D. Ott. *Astrophys. J.*, **799**, 190, 2015.
- [37] E. Cappellaro and M. Turatto. *Invited review at the meeting: "The influence of binaries on stellar population studies"*, ed. D. Vanbeveren (Brussels 21-25 Aug. 2000), 2000.
- [38] W. D. Arnett. *ArXiv:1408.0326*, 2014.
- [39] B. Paxton, M. Cantiello, P. Arras, L. Bildsten, E. F. Brown, A. Dotter, C. Mankovich, M. H. Montgomery, D. Stello, F. X. Timmes, and R. Townsend. *Astrophys. J. Supp. Ser.*, **208**, 4, 2013.
- [40] P. C. Joss, E. E. Salpeter, and J. P. Ostriker. *Astrophys. J.*, **181**, 429–438, 1973.
- [41] K. H. Prendergast and E. A. Spiegel. *Comments on Astrophysics and Space Physics*, **5**, 43, 1973.
- [42] E. A. Spiegel. Phenomenological photofluidynamics. In M. Rieutord and B. Dubrulle, editors, *EAS Publications Series*, volume 21 of *EAS Publications Series*, pages 127–145, 2006.
- [43] G. Gräfenor and W.-R. Hamann. *Astron. Astrophys.*, **482**, 945–960, 2008.
- [44] S. Ekström, C. Georgy, P. Eggenberger, G. Meynet, N. Mowlavi, A. Wyttenbach, A. Granada, T. Decressin, R. Hirschi, U. Frischknecht, C. Charbonnel, and A. Maeder. *Astron. Astrophys.*, **537**, A146, 2012.
- [45] J. J. Eldridge and C. A. Tout. *Mon. Not. Roy. Astron. Soc.*, **353**, 87, 2004.
- [46] G. Meynet, V. Chomienne, S. Ekström, C. Georgy, A. Granada, J. Groh, A. Maeder, P. Eggenberger, E. Levesque, and P. Massey. *Submitted to Astron. Astrophys.*, *arXiv:1410.8721*, 2014.
- [47] C. Georgy. *Astron. Astrophys.*, **538**, L8, 2012.
- [48] J. H. Groh, G. Meynet, and S. Ekström. *Astron. Astrophys.*, **550**, L7, 2013.
- [49] J. Puls, J. S. Vink, and F. Najarro. *Astron. Astrophys.*, **16**, 209, 2008.
- [50] J. H. Shiode and E. Quataert. *Astrophys. J.*, **780**, 96, 2014.
- [51] H. J. G. L. M. Lamers and J. P. Cassinelli. *Introduction to stellar winds*. Cambridge University Press, 1999.
- [52] J. Krticka and J. Kubat. *Accepted for publication in Astron. Astrophys.*; *arXiv:1406.1288*, 2014.
- [53] I. Brott, S. E. de Mink, M. Cantiello, N. Langer, A. de Koter, C. J. Evans, I. Hunter, C. Trundle, and J. S. Vink. *Astron. Astrophys.*, **530**, A115, 2011.
- [54] C. de Jager, H. Nieuwenhuijzen, and K. A. van der Hucht. *Astron. Astrophys.*, **72**, 259,

- 1988.
- [55] J. S. Vink. *arXiv:1406.5357*, 2014.
- [56] S. N. Shore. *Astrophysical Hydrodynamics*. Wiley-VCH, 2007.
- [57] M. J. Reid and J. M. Moran. *Ann. Rev. Astron. Astrophys.*, **19**, 231, 1981.
- [58] H. Nieuwenhuijzen and C. de Jager. *Astron. Astrophys.*, **231**, 134, 1990.
- [59] A. Maeder and G. Meynet. *Astron. Astrophys.*, **76**, 411, 1988.
- [60] A. Maeder and G. Meynet. *Astron. Astrophys.*, **210**, 155, 1989.
- [61] N. Mauron and E. Josselin. *Astron. Astrophys.*, **526**, A156, 2011.
- [62] J. T. van Loon, M.-R. L. Cioni, A. A. Zijlstra, and C. Loup. *Astron. Astrophys.*, **438**, 273, 2005.
- [63] Z. Ivezić, M. Nenkova, and M. Elitzur. DUSTY: Radiation transport in a dusty environment. Astrophysics Source Code Library, 1999.
- [64] J. S. Vink, A. de Koter, and H. J. G. L. M. Lamers. *Astron. Astrophys.*, **362**, 295, 2000.
- [65] W. Schmutz. *Stellar Atmospheres: Beyond Classical Models*. Springer NATO ASI Science Series, 1991.
- [66] J. S. Vink, A. de Koter, and H. J. G. L. M. Lamers. *Astron. Astrophys.*, **350**, 181, 1999.
- [67] E. Glebbeek, E. Gaburov, S. E. de Mink, O. R. Pols, and S. F. Portegies Zwart. *Astron. Astrophys.*, **497**, 255, 2009.
- [68] R. P. Kudritzki, A. Pauldrach, J. Puls, and D. C. Abbott. *Astron. Astrophys.*, **219**, 205, 1989.
- [69] J. I. Castor, D. C. Abbott, and R. I. Klein. *Astrophys. J.*, **195**, 157, 1975.
- [70] A. Pauldrach, J. Puls, and R. P. Kudritzki. *Astron. Astrophys.*, **164**, 86, 1986.
- [71] D. C. Abbott. *Astrophys. J.*, **259**, 282, 1982.
- [72] S. Wellstein and N. Langer. *Astron. Astrophys.*, **350**, 148, 1999.
- [73] A. W. A. Pauldrach, R. P. Kudritzki, J. Puls, K. Butler, and J. Hunsinger. *Astron. Astrophys.*, **283**, 525, 1994.
- [74] T. Nugis and H. J. G. L. M. Lamers. *Astron. Astrophys.*, **360**, 227, 2000.
- [75] W.-R. Hamann, D. Schoenberner, and U. Heber. *Astron. Astrophys.*, **116**, 273, 1982.
- [76] W.-R. Hamann, L. Koesterke, and U. Wessolowski. *Astron. Astrophys.*, **299**, 151, 1995.
- [77] W. R. Hamann and L. Koesterke. *Astron. Astrophys.*, **335**, 1003, 1998.
- [78] N. Smith and R. Tombleson. *Mon. Not. Roy. Astron. Soc.*, **447**, 598–617, 2015.
- [79] N. Smith, W. Li, A. V. Filippenko, and R. Chornock. *Mon. Not. Roy. Astron. Soc.*, **412**, 1522–1538, 2011.
- [80] N. Smith, W. Li, J. M. Silverman, M. Ganeshalingam, and A. V. Filippenko. *Mon. Not. Roy. Astron. Soc.*, **415**, 773–810, 2011.
- [81] B. Paxton, L. Bildsten, A. Dotter, F. Herwig, P. Lesaffre, and F. Timmes. *Astrophys. J. Supp. Ser.*, **192**, 3, 2011.
- [82] C. Angulo, M. Arnould, M. Rayet, P. Descouvemont, D. Baye, C. Leclercq-Willain, A. Coc, S. Barhoumi, P. Aguer, C. Rolfs, R. Kunz, J. W. Hammer, A. Mayer, T. Paradelis, S. Kossionides, C. Chronidou, K. Spyrou, S. degl’Innocenti, G. Fiorentini, B. Ricci, S. Zavatarelli, C. Providencia, H. Wolters, J. Soares, C. Grama, J. Rahighi, A. Shotter, and M. Lamahi Rachti. *Nuclear Physics A*, **656**, 3, 1999.
- [83] F. J. Rogers and A. Nayfonov. *Astrophys. J.*, **576**, 1064, 2002.
- [84] F. X. Timmes and F. D. Swesty. *Astrophys. J. Supp. Ser.*, **126**, 501, 2000.
- [85] E. Ramirez-Ruiz, M. Trenti, L. F. Roberts, W. H. Lee, and M. I. Saladino-Rosas. *ArXiv:1410.3467*, 2014.
- [86] P. Eggenberger, G. Meynet, a. Maeder, R. Hirschi, C. Charbonnel, S. Talon, and S. Ek-

BIBLIOGRAPHY

- ström. *Astrophysics and Space Science*, **316(1)**, 43, 2007.
- [87] M. Limongi and A. Chieffi. *Astrophys. J.*, **647**, 483, 2006.
- [88] R. Kippenhahn, G. Ruschenplatt, and H.-C. Thomas. *Astron. Astrophys.*, **91**, 175–180, 1980.
- [89] E. O’Connor and C. D. Ott. *Astrophys. J.*, **730**, 70, 2011.
- [90] P. A. Crowther. *Ann. Rev. Astron. Astroph.*, **45**, 177–219, 2007.
- [91] A. Maeder and P. S. Conti. *Ann. Rev. Astron. Astroph.*, **32**, 227–275, 1994.
- [92] S.-C. Yoon, S. E. Woosley, and N. Langer. *ArXiv e-prints*, 2010.
- [93] S. E. de Mink, M. Cantiello, N. Langer, S.-C. Yoon, I. Brott, E. Glebbeek, M. Verkoulen, and O. R. Pols. Rotational mixing in close binaries. In L. Deng and K. L. Chan, editors, *IAU Symposium*, volume 252 of *IAU Symposium*, pages 365–370, 2008.
- [94] A. Maeder and G. Meynet. *Astron. Astrophys.*, **182**, 243–263, 1987.
- [95] S. Shore. The Luminous Blue Variables and the IUE Legacy. In W. Wamsteker, R. Gonzalez Riestra, and B. Harris, editors, *Ultraviolet Astrophysics Beyond the IUE Final Archive*, volume 413 of *ESA Special Publication*, page 139, 1998.
- [96] C. A. Tout and P. P. Eggleton. *Mon. Not. Roy. Astron. Soc.*, **231**, 823–831, 1988.
- [97] W. Schmutz and L. Drissen. Spectroscopic analyses of the Wolf-Rayet stars in NGC 3603. In N. I. Morrell, V. S. Niemela, and R. H. Barbá, editors, *Revista Mexicana de Astronomia y Astrofisica Conference Series*, volume 8 of *Revista Mexicana de Astronomia y Astrofisica*, vol. 27, pages 41–48, 1999.
- [98] K. A. van der Hucht. *New Astron. Rev.*, **45**, 135–232, 2001.
- [99] S. V. Marchenko, A. F. J. Moffat, and P. A. Crowther. *Astrophys. J. Lett.*, **724**, L90–L94, 2010.
- [100] K. R. Sokal, S. L. Skinner, S. A. Zhekov, M. Güdel, and W. Schmutz. *Astrophys. J.*, **715**, 1327–1337, 2010.
- [101] G. Gräfener, S. P. Owocki, and J. S. Vink. *Astron. Astrophys.*, **538**, A40, 2012.
- [102] L. Dessart, D. J. Hillier, R. Waldman, and E. Livne. *Mon. Not. Roy. Astron. Soc.*, **433**, 1745, 2013.
- [103] W.H. Press, S.A. Teukolsky, W. T. Vetterling, and B.P. Flannery. *Numerical Recipes - Third Edition*. Cambridge University Press, 2007.

Acknowledgements

Science is a way of trying not to fool yourself. The first principle is that you must not fool yourself, and you are the easiest person to fool.
[R. Feynman]

It is so easy to fool ourselves, this is why social interactions with people are so important in doing science. As any other, this coin has two sides: on the one hand it leads to competition, which is never a genuine driving factor. Far too often, competition causes stress, bad practices (to obstacle the “competitors” instead of improving ourselves) and all this things that one would naively think do not regard fundamental sciences. Many would love to see physics as a purely intellectual activity. Instead, it is important to recognize that scientists are people, and these negative aspects exist and should be counteracted. On the other hand, social interactions not only prevent self-fooling ourselves thanks to feedback on our work, they also enrich us not only as scientists, but as people.

While working on this thesis, and, more generally, during my years at the University of Pisa, I have had the huge chance to really and deeply enjoy the good side of that coin. I am really grateful to all of you for this. It is impossible to mention every single person that has “perturbed” my path in this years, creating small variations that may have changed the trajectory toward my personal strange attractor.

I cannot thank enough Christian and Steve for the help, guidance, and support provided during these months. You have been patient and responsive, and it has been (and I hope it will go on being) a great pleasure to work with you. I really think that you thought me much more than “just” science, and I feel I grew as a person, not just as a physicist, thanks to you.

I had the opportunity of interacting with many people that gave paramount contributions to this thesis, maybe even without noticing it. I should thank the entire MESA Community, and Bill Paxton for the invaluable help, but also for releasing publicly the code (I still do not understand how people can claim to do science with closed-source codes).

Among all those I had the chance to meet during my visits overseas, I want to thank especially Drew, Matteo, Jim, Sean, Philipp, Luke, Sebastiano, Viktoriya, JoAnn, of course my roommate Christian R., and can I legitimately mention Ushi?

I also want to thank Luc Dessart and Tuguldur Sukhbold for sharing their parameters file with me.

Most of the computations have been carried out on computers at Caltech, especially on

BIBLIOGRAPHY

Fermi, Bethe and the Zwicky cluster – which had to suffer through all the mistakes I managed to do. A smaller portion of the calculations has been ran on the NSF XSEDE network. I mention this to thank all the people maintaining these machines, that allow lucky people like me to play God with the laws of nature.

But are these laws really there, written *in questo grandissimo libro che continuamente ci sta aperto innanzi a gli occhi*[†]? Or does just nature do stuff, and we struggle to put some order in it? Do we find the laws of nature, or do we just impose them in our models? I have had the pleasure to discuss these questions with my friends during our studies, and I still think that these are more important than knowing how to solve some crazy-difficult integral. Some of you agree, some do not. But, still, it has been awesome to literally *live* together (*attenti al lupo!*[‡]) through University with you guys: Bob, Nino, Andrea R., Andrea G., Mimmo, Daniele, Gina, Raissa, Antonio, Vittoria, Giacomo, Giorgia, Davide, Paolo, Cosimo, Giannandrea, Leonardo, Umberto (with whom I shared 11 years of education and life, how can you still stand me?), Gaia, Luca, Enza. I cannot condensate in a few lines what you have been to me.

Alessandro, Federica, Ilaria, and Federico have walked with me the thesis path here in Pisa, *thanks for the all the fish*^{*}.

Agnese, Francesca, Tiziano, Valentina have hosted me so many times at the “Reggia Solferino”, where I always found a welcoming and friendly environment. Thanks for the support!

There are also two places in Pisa I want to thank: Piazza dei Cavalieri (especially late at night), and the AM1. Thanking a physical location may seem a weird thing to do, but the atmosphere I found there, the melting and mixing crowds that cross these two places are things that I will miss about this city. In these crowds, I met Cheikh. Your strength has been an inspiration.

Sofia, your support and love have been much more important than I could say and than I deserved. Despite the ocean, the different time-zones, and all the difficulties, writing this thesis *never could be sweeter than with you*^{††}.

The support of my parents, and my entire family, as been absolute. This thesis is the result of an on going learning process that you started, and this is much less obvious than it should be.

Finally, the University of Pisa and the Department of Physics. This is a tricky acknowledgment, since I really had the best chances I could wish here, especially the INFN/LIGO program that brought me to Caltech in the first place, and the funds for thesis abroad for a second round trip. Looking back, I am happy I decided to stay here. However, I feel lucky and this should not be. I think that for institutions like these, it is not quite enough to give the best chances available to a few lucky students, and let all the many others struggle in very unnecessary, uncomfortable situations. In this years I have seen too many things getting worse (especially the absence of exams). Too many times I felt that the assumption that students are just lazy, and that some kind of “social darwinism” must be applied to select them. This should change, otherwise physicist coming out from here will only learn the bad side of the coin of social interactions in an academic environment.

[†]G. Galilei, “Il Saggiatore”

[‡]L. Dalla, “Attenti al lupo”, Cambio

^{*}D. Adams, “The Hitchhiker’s Guide to the Galaxy”

^{††}Edward Sharpe and the Magnetic Zeros, “Home”, Up from Below