Alma Mater Studiorum · Università di Bologna

School of Science
Physics and Astronomy department
Master Degree in Astrophysics and Cosmology

Analysis of stellar evolution models for galaxy formation simulations

Supervisor:
Prof. Federico Marinacci

Candidate:
Lorenzo Vannini

Co-Supervisor:
Dr. Giuliano Iorio

Academic year 2023/2024

Contents

1 Introduction **6**

1.1 Galaxy formation and evolution 6

1.2 Historical recap of numerical methods 12

1.3 Numerical methods in Astrophysics 14

1.4 Discretization 17

1.4.1 Eulerian methods 18

1.4.2 Lagrangian methods 19

1.4.3 Arbitrary Lagrangian-Eulerian codes 19

1.5 N-body solvers 20

1.5.1 PM-Method 21

1.5.2 Tree-method 22

1.5.3 Tree-PM Method 23

1.6 AREPO 23

1.6.1 Computational mesh 25

1.6.2 Hydrodynamic solver 27

1.7 Cooling and heating 29

1.8 Star Formation and Stellar Feedback 31

1.8.1 Star Formation 32

1.8.2 Stellar Feedback 33

2 Methods **37**

2.1 Introduction 37

2.2 Fire-3 38

2.2.1 Supernovae 39

2.2.2 OB and AGB winds 43

2.2.3 Yields 46

2.3 SMUGGLE 48

2.3.1 Model Theory 49

2.3.2 IMF 52

2.3.3 Data interpolation . . . 53
2.3.4 Lifetime function . . . 53
2.3.5 SN rates . . . 54
2.3.6 SNII mass ejection . . . 56
2.3.7 Low mass stars . . . 57
2.3.8 Total mass ejection . . . 58
2.3.9 Elemental Yields . . . 60
2.3.10 Total Metal Yield . . . 63

3 Model Construction 67

3.1 Introduction . . . 67
3.2 The SEVN code . . . 68
3.3 Parameter dependencies . . . 70
3.3.1 Spin parameter dependencies . . . 71
3.3.2 Type II Supernova Formalism . . . 71
3.4 End-of-life mass loss . . . 73
3.4.1 Lifetime function and SNII rate . . . 74
3.4.2 Efficiency function of the new model . . . 78
3.4.3 Mass ejection rate . . . 82
3.4.4 Total mass ejection fraction . . . 85
3.5 Stellar Winds . . . 88
3.5.1 Interpolating Mass Ejection Rate by Stellar Winds . . . 90

4 Comparison and discussion 93

4.1 Introduction . . . 93
4.2 Mass ejection rate . . . 94
4.3 Total ejected mass fraction . . . 97
4.4 Elemental yields . . . 101
4.5 Metal yield . . . 103

5 Summary and conclusions 105

Bibliography 109

Appendixes 113

A .hdf5 files . . . 114
B Interpolation of the lifetime function . . . 115
C New DeathMatrix implementation in SEVN . . . 116

A Love Supreme

Abstract

Our theoretical understanding of galaxy formation relies heavily on numerical simulations to model the evolution of these systems over cosmic time. Most state-of-the-art simulations incorporate stellar evolution to allow galaxies to evolve self-consistently. However, since resolving single stars in galaxy-scale simulations is computationally unfeasible, stellar evolution must be represented through effective prescriptions at scales below the resolution of the simulation, an approach known as "sub-grid physics". In this thesis, we investigate how stellar evolution is implemented in two widely used galaxy formation models: SMUGGLE (Marinacci et al., 2019) and FIRE-3 (Hopkins et al., 2022). We compare and characterize the key features of their respective stellar evolution prescriptions and explore the potential implications that these differences have on the evolution of the interstellar medium (ISM) in terms of mass, chemical composition and star formation. We then use the stellar population code SEVN (Iorio et al., 2023) to develop a new stellar evolution module for the SMUGGLE model. To achieve this, we create a software pipeline that replicates SMUGGLE implementation of stellar evolution and, using input obtained from SEVN, generates updated stellar evolution models. This pipeline can be applied to other simulation frameworks that share the same stellar evolution infrastructure as SMUGGLE, such as the Illustris and IllustrisTNG models. Finally, we compare the original SMUGGLE stellar evolution model with the updated version derived from SEVN, finding that the newer model predicts a lower mass return to the ISM, but with a higher level of metal enrichment. We highlight that pipeline developed in this thesis has the potential to facilitate the generation of future stellar evolution models and to assist in analyzing the results of galaxy simulations, especially when a precise determination of the stellar mass return and total metal production are needed.

1

Introduction

1.1 Galaxy formation and evolution

"Galaxies are gravitationally bound systems made of stars, interstellar matter(gas and dust), stellar remnants (white dwarfs, neutron stars and black holes), and a large amount of dark matter"(A. Cimatti, F. Fraternali, C. Nipoti, 2019) . Their discovery dates back to the 16th century when mankind started building and using telescopes to study the night sky. Still, until the 20th century, a great debate raged in the scientific community between those who thought that the so-called "Fuzzy-nebulae" were objects similar to other diffuse sources, like planetary nebulae or HII regions, and those who believed that these nebulae were much more distant objects, with the Milky Way being one of those. The debate ended in the early 20th century when Edwin Hubble used variable stars, called Cepheids, to measure the distance to the Andromeda Galaxy and found that this distance was several orders of magnitude higher than other diffuse objects of the Milky Way (Hubble, 1925). From this point on, the modern concept of galaxy emerged. Our current understanding of the formation and evolution of galaxies is an interdisciplinary endeavor that spans from sub-atomic physics to cosmology, passing through chemistry, theory of stellar evolution, and magneto-hydrodynamics, among others. The incredible variety of the phenomena at play in shaping galaxies' properties and the large range of scales that they span result in one of the most complicated and fascinating topics in all modern science. The very first intuition that there might be a universal law that

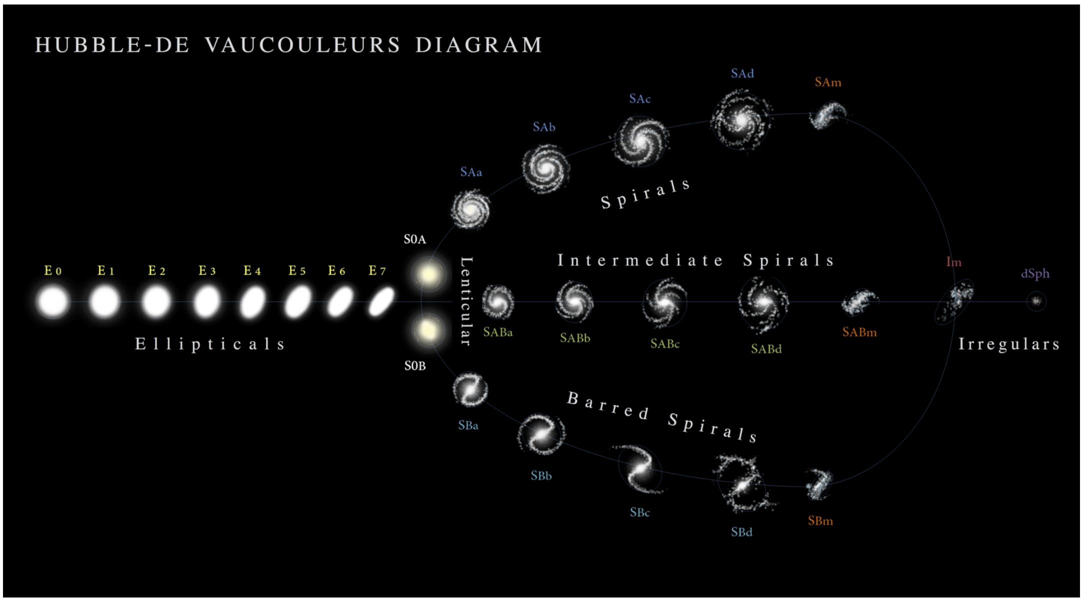


Figure 1.1: Visual representation of the Hubble-de Vaucouleurs classification. Notice not only how galaxies are distinguished between elliptical and spirals, but they are also classified according to their apparent ellipticity or depending on the winding of spiral arms, the prominence of the bulge, and the presence of a bar. (Antonio Ciccolella / M. De Leo).

governs how these systems evolve comes from the observations of their morphology and the fact that galaxies obey precise scaling relations. The study of galaxy morphology dates back to the early observations of the "fuzzy nebulae" mentioned earlier. The current classification is called the "Hubble- de Vaucouleurs" classification (De Vaucouleurs, 1959), Fig. 1.1. This classification divides galaxies into two main types: ellipticals and spirals. Over the years astronomers realized that this dichotomy corresponds most of the time to the absence or presence of cold gas clouds collapsing and forming stars. It has been proposed that the difference in morphology, as well as the difference within the same morphological classes of galaxies, emerges from different formation histories (A. Cimatti, F. Fraternali, C. Nipoti 2019 and reference therein). This hypothesis has been corroborated by many researchers over the years. For instance, the abundance of a specific type of galaxy in different cosmological environments is a strong indication that different morphologies are a result of distinct types of evolutionary scenarios between galaxy classes; we tend to find big elliptical galaxies in the central part of large galaxy clusters while spiral galaxies and irregular tend to be more present in the outskirts of these big DM potential wells (Balogh et al., 2004). Another key argument that suggests a different evolutionary path between different morphological types comes from the so-called "downsizing scenario" (Thomas et al., 2010). This term indicates the tendency of big, elliptical galaxies that do not show relevant SF to have formed earlier and faster with respect to spiral galaxies, which most of the time are still forming stars. A satisfying model for galaxy formation should explain then both morphological abundances and the downsizing scenario. Not only morphological characteristics should drive our quest for a comprehensive model for galaxy formation and evolution. In fact, researchers were able to find tight scaling relations between many different properties of the galaxies. These relations connect observed (or derived from observations) properties to other physical quantities of galaxies, indicating a regularity in the way these systems form and evolve (A. Cimatti, F. Fraternali, C. Nipoti, 2019). The most important are:

- Tully-Fisher relation (Tully, R. B. & Fisher, J. R., 1977), relating the speed of rotation with the luminosity of spiral galaxies,
- the fundamental plane for elliptical galaxies, relating the surface brightness, the effective radius and the velocity dispersion (Jørgensen et al., 2006);
- the star formation main sequence $SFR \propto M^{\beta}$,relating the SFR of star forming galaxy to their mass (Popesso et al., 2022);
- Black Hole Mass-Bulge Mass Relation, relating the mass of the central black hole with the mass of the galaxy's bulge and its velocity dispersion, indicating that the central black hole plays a fundamental role in the evolution of galaxies (Marconi & Hunt, 2003).

It is important to highlight that a "correct" model describing how galaxies form, evolve, and interact must reproduce these observed relations and the galaxies' morphological mix. This requires a detailed description of the dynamical properties of such objects. Our current understanding of galaxies is based on the notion that these systems form from the collapse, virialization, and subsequent hierarchical growth of the perturbation from the primordial matter density distribution in the Universe. If we want to build a theory that describes the evolution and formation of galaxies, we then need to assume a cosmology. Our current cosmological model for the Universe, which is able to satisfactorily explain most of the observational evidence, is called ΛCDM, a geometrically flat universe made of five different components: Radiation, Neutrinos, Baryonic matter (matter that interacts with the electromagnetic filed), Dark matter (that does not interact with the electromagnetic field), Dark energy (a component of unknown origin that counters gravitational attraction on a cosmological level). Cosmologists use the Ω_i parameter to quantify how much of a specific component is present in the Universe.

They define $\Omega_i = \frac{\rho_i}{\rho_c}$ where ρ_i is the energy density of a component i and $\rho_c = \frac{3H^2}{8\pi G}$ is the critical density that the Universe needs to have in order to have a flat geometry. Current measurements of density parameters (DESI Collaboration et al., 2024; Planck Collaboration et al., 2020) give $\Omega_m \sim 0.315$ (matter, including dark matter and baryonic matter), $\Omega_\Lambda \sim 0.685$ (Dark Energy) with $\Omega_b \sim 0.049$ (Baryonic Matter) and $\Omega_r \sim 0.0001$ (Radiation). The cosmological context in the formation of galaxies is important for a variety of reasons. The most evident is related to how much dark matter (DM) and baryonic matter is present in the Universe. The former composes 85% of the total budget of the matter while the latter accounts for the remaining 15%. This disparity plays a fundamental role in the formation and evolution of galaxies, in fact, the gas (baryonic matter) is dragged where DM clusters, forming virialized structures called DM halos, evolving in the potential well provided by these halos. A different ratio between DM and baryonic matter would result in different appearance and formation processes of galaxies in the Universe. When considering the hierarchical growth hypothesis –a direct consequence of the ΛCDM model – where smaller virialized dark matter halos merge to form more massive ones, we obtain a scenario in which baryonic systems formed within these small halos do not evolve in isolation. Instead, they interact and eventually merge, with these processes playing a significant role in the evolution of the galaxies hosted by the halos.

Gravitational interactions are not the only phenomenon at play in shaping the evolution of galaxies. Baryons differ from DM in their capacity to interact with the electromagnetic field and this introduces a plethora of processes that influence the evolution of the baryonic component of the Universe. For instance, baryons can emit photons that carry away energy, allowing for further collapse of this component after it becomes self-gravitating, and can also exchange energy and momentum through collisions.

Moreover, gas clouds collapse and form stars. Stars are the main light emitters in the optical UV and NIR part of the electomagnetic spectrum. The constant output of energy stars give modify the appearence of the ISM through radiation, mass loss during the lifetime (stellar winds), and endlife events (Supernovae and planetary nebulae.) All these sources of energy from stars couple with the ISM in various ways. For instance, radiation can inject momentum and push the interstellar gas through radiation pressure and can ionize the gas. Winds can also inject momentum and shock the ISM. Supernovae explosions are able to expel the gas in small system and drive the mixing of the ISM.

Modeling the evolution of galaxies then requires calculating correctly the gravitational potential they live and evolve in, understanding and modeling how gas interacts and becomes stars, and describing how stars and the AGN modify the environment. All of this must be carried out using an accurate model based on (differential) equations that quantify and describe each mechanism that we think could be relevant in the context of galaxy formation and evolution. In order to quantify how the simultaneous effects of all these aspects in galaxy formation interface with each other during the evolution of the Universe, we need to create a numerical model capable of solving this complex mathematical problem. Astrophysicists then started to produce and test models that encoded physical processes in numerical simulations. This type of theoretical research aims at improving all the aspects of modeling based on computational techniques, from developing and coding more efficient ways to solve differential equations to implementing more physically-faithful formulations that describe a specific phenomenon (for example how to implement stellar evolution inside a simulation).

This thesis focuses on comparing different implementations of stellar evolution in simulations of galaxy formation with the ultimate goal of refining how stellar evolution processes are included in the SMUGGLE model, ”an explicit and comprehensive stellar feedback model for the moving-mesh code AREPO” (Marinacci et al., 2019). We decided to investigate this topic since stellar evolution in galaxies is one of the crucial aspects in shaping their evolution. Galaxies are the birthplace of most of the stars in the Universe and the environment in which stars live, evolve, and die. The continuous interplay between the interstellar medium (ISM) and the stars that live there makes accurate modeling of stellar evolution a fundamental priority if we aim to obtain realistically-looking objects as a result of a cosmological hydrodynamic simulation.

The ISM is profoundly influenced by the stars that inhabit it in various ways. First of all massive stars (namely O/B type stars) emit photons capable of ionizing the hydrogen atoms ($E_\nu \geq 13.6$ eV). Regions where the rate at which the gas is ionized at the same rate at which it recombines are defined as being in photoionization equilibrium. This continuous injection of energy from the absorption of photons heats the gas around these stars to ($T \approx 7000$–$13000\,\mathrm{K}$). Recombination, instead, results in the prominent emission lines of metals and hydrogen so that this region is kept in equilibrium between the energy injected by stellar photons and emission from recombination. The presence of HII regions corresponds to the presence of massive stars with a lifetime of the order $\sim 10^6$ yr. We can then use the presence of the HII region as a tracer of active star formation (SF) in galaxies, approximating the SF timescale as one of the life of massive stars. Having a correct model for massive stars is then fundamental to having an accurate description of ISM physics.

Another process by which stars modify the ISM is by injecting momentum and energy through stellar winds. These winds influence the properties of the ISM by increasing the overall temperature, changing its density, and creating shocks and cavities. As Jeans mass (i.e. the critical mass for gravitational collapse) is inversely proportional to the square root of the gas density and directly proportional to the square of the temperature, we expect winds to influence the star formation rate of the simulation in both ways, both increasing it by compression and diminishing it by increasing overall temperature (Lancaster et al., 2021). The nature of the complex relation between stellar winds and star formation must then be investigated in detail, improving models and understanding how they impact the outcome of the simulations.

Another key mechanism of energy and momentum injection in the ISM is SNe explosions. SNe are thought to be responsible for a variety of effects in the ISM since they deposit on average an energy of $E_{SNII} \sim 10^{51}$ erg at the moment of the explosion. Because of this large injection of energy, SN explosions are believed to be responsible for both triggering SF, by compressing the ISM and depleting of gas smaller gravitational bound systems that have binding energy lower than E_{SNII}. SNII explosions are also the cause of the onset of the so-called galactic fountain cycle, which is a main driver for the mixing of the expelled material and the CGM in galaxies and a key phenomenon that shapes their evolution (Fraternali, 2017).

In addition to the dynamical effects that they have on the ISM, stars are the driver of the chemical enrichment of the interstellar medium. Nuclear reactions produce heavy elements up to iron (the heaviest element that can be produced by exothermic fusion channels), SNII explosions can produce heavier elements in the explosion mechanism through neutron capture processes. Therefore, stellar evolution pollutes the ISM changing its metallicity when mass is released by events occurring at the end of the life of a star or through stellar winds ejections. The chemical enrichment of the ISM has dramatic effects. Astrophysical gases are able to lose energy via the emission of photons, in order to quantify this energy loss as a function of the gas temperature and its composition astrophysicists defined the cooling function $\Lambda = \Lambda(Z, T)$. This function encodes the rate of energy loss for unit time and for unit mass for gas at temperature T with metallicity Z. The importance of metals in the cooling function is given by their ability to enhance the cooling rate of the system by emitting photons through forbidden lines. A cloud of gas that cools tends to increase its density and lower its temperature, ultimately leading to the formation of giant molecular clouds (GMC). GMC are the main sites where star formation occurs (Chevance et al., 2022). This connection shows that cooling is related to star formation and that different levels of metal pollution will result in different cooling functions that will change the SF history of a given galaxy. Chemical enrichment is also responsible for the formation of dust grains, these grains change the appearance of the ISM in galaxies, they provide shielding for GMC from ionizing radiation and they reprocesses light emitted from stars changing the appearance of ISM by blocking visible and UV light, and emit in the near-infrared band.

Finally, the mass ejected from stars as a consequence of stellar evolution processes can be reused to make new stars. Even if not all the gas is eventually recycled in the younger stellar generations, we expect that the more the mass is injected back into the ISM, the more stars the galaxy will form owing to the availability of a larger quantity of gas. We can then assume that, in simulations of galaxy formation, the star formation rate depends on the stellar model implemented.

As already discussed, the aim of this thesis is to improve the modeling of stellar evolution in the SMUGGLE model, which is part of the moving-mesh code AREPO (Weinberger et al., 2020). In the framework of the model, stars cannot be described as a single entity due to memory and computational limitations, but rather as star particles representing a single simple stellar population (SSP), an ensemble of stars distributed following a given initial mass function (IMF) all born in the same star formation event. To gain a deep understanding of how different choices in the detailed implementation of the model for stellar evolution will impact the simulations, we will describe in the next subsections of this introductory chapter both the code AREPO and how all the key aspects related to stellar physics and feedback are currently implemented in SMUGGLE. In Chapter 2 we will then describe in detail how stellar evolution is implemented in SMUGGLE and another major framework, namely the FIRE (Hopkins et al., 2022) model, that will be used for comparison. In Chapter 3 we will show how we used the code SEVN (Iorio et al., 2023) to build an updated stellar evolution model that can be used for all the simulations that share a similar code infrastructure with SMUGGLE. In Chapter 4 we will then discuss the results obtained with SEVN and compare them with the models presented in Chapter 2. Lastly, in Chapter 5 we will summarize the results of this work and we will briefly introduce future potential improvements for stellar evolution models in hydrodynamical simulations of galaxy formation and evolution.

1.2 Historical recap of numerical methods

The use of numerical methods is widespread and spans all disciplines that use some kind of mathematical modelization. These methods trace back to ancient civilizations, such as Ancient Greek and even Babylonian civilizations. From this epoch, we find archaeological proofs of numerical algorithms used to approximate the square root of numbers and examples of linear interpolations between data points to approximate what we now call functions (Høyrup, 2002). It is then clear that since the very dawn of mathematics, many scholars adopted numerical approximation to gain insight into mathematics itself and natural sciences. In the seventeenth century, Physics emerged as a quantitative science and the potential use of numerical methods sparked. In fact, at the very foundations of Physics stands the assumption that we can describe a generic system of bodies with a mathematical model. Usually, a model is formalized in a set of arbitrarily complicated differential equations whose solution describes the result of an experiment.

We can define a differential equation as an equation in which the unknown quantities are functions and their derivatives. For example, Newton's second law is a differential equation, whose solution is the function that describes the position as a function of time. In Physics, we assess the validity of a model by solving its constitutive differential equations and by comparing such solutions with the experimental data. This requires methods that can find the solutions of differential equations in the first place.

From the very moment calculus was invented/discovered a considerable amount of work has been carried out to find analytic solutions and new ways to approach them, for example with the definition of Sobolev spaces and the introduction of the weak formulation (Adams, Robert A. & Fournier, John, 1975). While the quantity of problems that theoretical physicists and mathematicians were able to solve analytically is remarkable, finding an analytical solution is a very complicated and time-consuming task with no certainty of success. Especially for complicated, non-linear partial differential equations finding a general analytical solution might be unfeasible.

After the concept of derivative was introduced by Newton and Leibnitz, it was immediately clear the possibility of getting an approximate value of this quantity by simply taking the definition:

$$\frac{df(x)}{dx} \sim \frac{f(x+\Delta x)-f(x)}{\Delta x} \tag{1.1}$$

So, if it is possible to approximate derivatives, it is possible to approximate the derivatives in differential equations. This methodology can approximate differential dx as discrete difference Δx and get a numerical approximation of all problems that present these mathematical entities, like integrals, derivatives, and, most of all, differential equations. Until the first half of the 20th century, this approach required to perform all the calculations involved by hand. Given the considerable amount of time needed to carry out these calculations in such a way, most of the work concentrated on getting analytical solutions over studying better numerical methods. This situation changed completely after the invention of Von Neumann machines (Von Neumann, 1945), in which the architecture of the modern computer was theorized. The computer allowed for very fast computation of a number of operations unimaginable before which output could be kept in memory to perform other calculations. From this point, mathematicians started to search for more accurate and less computationally expansive methods to solve differential equations using the computer. Astrophysics was the perfect ground to test these new ideas most of the models that presented themselves in astrophysics consisted of complicated systems of partial differential equations. With the computer, astrophysicists were able to find approximate solutions to equations that were until that moment, practically unsolvable. With the advent of numerical methods new and more refined models emerged boosting the research in the field.

1.3 Numerical methods in Astrophysics

As mentioned in the previous section, the use of numerical methods is widespread in astrophysics. Even the simplest problems, like for example the three-body problem, in which three point-like masses interact only through gravitational forces, cannot be solved analytically for a general set of initial conditions and the use of numerical simulations is needed to investigate all the parameter space. Therefore, it is not surprising that computational astrophysics has become a key discipline in modern astrophysics research. This is particularly true in cosmology and the study of galaxy and structure formation in the Universe. One of the main research branches in cosmology is the study of how the initial perturbation in the mass density field (i.e. the density contrast $\delta = \delta\rho/\rho$) leads to the formation of the structure we observe in the present day Universe. Investigating the evolution of the density contrast requires solving a continuous problem in which each infinitesimal element of space has a gravitational force acting on it and makes up for the total potential of the gravitational field. This introduces the very first assumption that we assume when we perform a numerical analysis of a system governed by differential equations: we must discretize the problem to make it solvable by a computer. It is possible to solve analytically the evolution of the perturbations in linear regime (P. Coles, F. Lucchin, 2002), still, when perturbations start to become highly nonlinear ($\delta >> 1$) the problem must be tackled using simulations to to get solutions and data to test our models on observations.

Most cosmological simulations are performed using DM only, i.e. matter that interacts only through gravity. These tools allow us to find a numerical solution for the evolution of the total gravitational potential generated by an initial density field solving the famous Laplace equation:

$$\nabla^2\Phi(\vec{x}) = 4\pi G\rho(\vec{x}) \tag{1.2}$$

Where Φ is the gravitational potential, G is the gravitational constant, and $\rho(\vec{x})$ is the mass density field. After we solve for the potential, we can evolve the system using the equation of dynamics, and assuming a stationary potential, then we recompute the potential and iterate this procedure.

In our Universe also matter that interacts with the electromagnetic field is present. In the current cosmological model (ΛCDM) the contribution to the matter-energy budget of the Universe of the different components is $\Omega_\Lambda \approx 0.685$ for the density parameter of dark energy, $\Omega_c \approx 0.264$ for cold dark matter, and $\Omega_b \approx 0.049$. Therefore, the matter contribution is dominated by dark matter, and this is why historically many cosmological simulations are carried out without the baryonic components. Years of research in cosmology showed that the formation of structures in the Universe is driven by DM (A. Cimatti, F. Fraternali, C. Nipoti, 2019), with baryons consequentially falling where DM clumps. Dark energy becomes relevant only at late times $z \approx 1$, this component is necessary for explaining the observed acceleration of the universe (Riess et al., 1998) and its inferred flat geometry from the cosmic microwave background analysis (Spergel et al., 2003).

As mentioned above, DM drives the collapse and therefore the evolution of structures, still we can only observe baryonic matter. The framework of this thesis is the subgrid physics in simulations of Milky-Way type galaxies' evolution and the modelization of baryonic processes that happen inside of them. Baryons interact with the electromagnetic field and can emit photons. This crucial difference between dark matter and baryonic matter allows the latter to cool and collapse further after reaching virial equilibrium, as a virialized baryonic structure can lose energy through photon emission. Another key difference is that baryonic matter is collisional, meaning that particles can interact and exchange energy and momentum through electromagnetic collisions. These interactions give rise to viscosity, turbulence, feedback mechanisms, and pressure gradients. Together, these features create a much more complex and varied structure for baryonic matter, that we can indeed observe directly with telescopes. To accurately describe these effects, it is necessary to incorporate hydrodynamical equations, which can describe accurately the dynamics of gas, taking in account all the effects listed before. We can write the hydrodynamical equations, also known as Euler equations, neglecting magnetic fields as:

$$\begin{cases} \dfrac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{v}) = 0 \\ \dfrac{\partial (\rho \vec{u})}{\partial t} + \nabla \cdot (\rho \vec{v} \vec{v}^T + P) = 0 \\ \dfrac{\partial E}{\partial t} + \nabla \cdot (E + P) \vec{v} = 0 \end{cases} \tag{1.3}$$

where ρ is the density, $\vec{v}$ is the velocity, P is the pressure, E is the total energy density of the fluid. The energy density is defined as:

$$E = \frac{\rho v^2}{2} + \frac{P}{\gamma - 1} \tag{1.4}$$

where we can clearly notice that the first part is associated to the motion of the fluid cell and the second is related to the motion of microscopic particle inside the fluid cell itself (i.e., thermal motion). To solve system (1.3) we need another equation that relates the pressure with energy, this is the so-called equation of state and can assume different forms depending on the assumption we made on the system. In the case of perfect gas (the most common assumption adopted in cosmological simulations) we can write $P = (\gamma - 1)\rho e$ where γ is the adiabatic index and e is the specific internal energy (energy per unit mass).

We can write Euler equations (1.3) adopting a vector notation for simplicity:

$$\mathbf{U} = \begin{pmatrix} \rho \\ \rho \vec{v} \\ \rho e \end{pmatrix} = \begin{pmatrix} \rho \\ \rho \vec{v} \\ \rho u + \frac{1}{2} \rho v^2 \end{pmatrix} \tag{1.5}$$

where ρ is the mass density, $\vec{v}$ is the velocity field, and $e = u + 1/2\rho v^2$ is the total energy per unit mass; u gives the thermal energy per unit mass, which for an ideal gas is fully determined by the temperature. All these quantities are a function of the position and time, so we have $\mathbf{U} = \mathbf{U}(\vec{x}, t)$. Each component of the vector $\mathbf{U}$ corresponds to a conserved quantity if integrated over a (control) volume, we can then write analogously the fluxes for each quantity as:

$$\mathbf{F}(\mathbf{U}) = \begin{pmatrix} \rho \vec{v} \\ \rho \vec{v} \vec{v}^T + P \\ (\rho e + P) \vec{v} \end{pmatrix} \tag{1.6}$$

In this way we can write all the Euler equations as a single (vector) conservation equation:

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbf{F} = 0 \tag{1.7}$$

In galaxy simulations, and more in general in all large-scale simulations that model not only gravity, including the hydrodynamical equations and solve them numerically is not sufficient to obtain realistic results. Our current computational capabilities do not allow us to resolve the smallest scales that are involved in shaping a galaxy evolution, while simultaneously describing the full cosmological context in which such evolution takes place. For example, we are not able to simulate every single star inside the galaxy as a single entity since their number in a galaxy similar to the Milky Way averages $\sim 10^{11}$. Simulating a problem with such several entities would be unfeasible due to both time and memory limitations. These computational limitations require an approximate treatment in which we model the collective effects of an ensemble of unresolved physical entities inside a single cell of a simulation. This is known as sub-grid physics.

The work in this thesis focuses on describing, comparing, and refining the stellar evolution model implemented in galaxy formation simulations. As we said, considering all the stars in a galaxy as single entities would be computationally unobtainable, so all stars burn in the same star formation event are grouped together and evolved as single entities. The ensemble of all the stars burning in the same stellar formation event is called SSP. In our work we analyze and compare how SSP evolution presents in many different works and how we were able to build a new model for SSP evolution (we will refer to SSP evolution also as stellar evolution, since in the context of these types of simulation they are equivalent).

In order to understand how different models could impact the simulation in which they are implemented we first need to introduce the context in which they insert. What follows then is a description of all the steps needed to initialize a computer a cosmological simulation. We first describe which are the main approaches to discretize the problem and how to solve gravity. Then we introduce AREPO, the code in which we will implement our new stellar evolution model. We will describe how it solves hydrodynamical equations. Lastly, we introduce SMUGGLE and we will show how unresolved physical mechanisms are implemented in galaxy formation simulations.

1.4 Discretization

To write a code that solves numerically a set of differential equations we need to discretize the domain of the problem. Functions that are solutions of differential equations representing a concrete system are, in general, defined in such a way: $f : A \subset \mathbb{R}^n \to B \subset \mathbb{R}^m$, this means that solutions are defined between uncountable sets. This peculiarity forces us to approximate a continuous problem into a finite set of discrete values to make it solvable by a machine with finite memory. In cosmological simulations, we deal with functions that go from a set of four coordinates (indicating the positions in space and time) to a scalar value (the mass density for example) or a vector (the velocity). When we deal with equations of hydrodynamics, we can write such differential equations in two ways. We can choose to study the evolution of a fluid quantity at a point in space considering the partial time derivative and the variation due to the motion of the fluid (Eulerian formulation), or we can follow a fluid element and describe how its properties change as it moves (Lagrangian formulation). These two approaches correspond to two different formulations of the equations and are tightly connected to the way we solve the numerical problem. Following these two types of approaches, we can discretize the problem in three ways. We can divide the domain of the simulation in a grid, assign to each element of this mesh the quantities we are interested in, and then calculate the evolution of such quantities in each grid element. Alternatively, we can discretize the fluid by tracking fluid elements as they move with the flow, following their evolution over time. This approach models the interaction between these elements based on the governing physical laws, such as the conservation of mass, momentum, and energy. Finally, we can use a combination of the two previous methodologies. These approaches are known as Eulerian methods, Lagrangian methods, and Arbitrary Lagrangian-Eulerian methods, respectively. In the next subsections, we will introduce and briefly describe their salient traits.

1.4.1 Eulerian methods

As anticipated, in Eulerian codes the discretization is made by partitioning the simulation domain in a mesh. The mesh appears as a collection of grid elements, and in Eulerian methods, we study the evolution of the quantity of interest (such as density, momentum, etc.) inside each grid element. The choice of the numerical scheme to solve the evolution depends on the required order of accuracy and the type of physical problem that we are to solve (for example, upwind methods work better for describing transport, while FTCS for diffusion problems; Brighenti 2021). In Eulerian methods, different approaches to approximate the quantities inside a grid cell can be adopted. For instance, in the finite volume approach the quantity of interest is treated as an average value over the entire grid cell (or control volume), whereas in finite difference methods, the quantity of interest is typically considered as a value at a specific point in the grid cell or at the grid nodes (usually at the cell center).

When treating general problems, shocks (discontinuous solutions whose derivatives

do not exist everywhere) can occur. This imposes stringent requirements on the numerical scheme adopted to solve the problem. To mitigate numerical issues associated with shocks we need a conservative scheme to ensure stability in the numerical solution (Eleuterio F. Toro, 2009). One of the most widely used methods are the Godunov methods. These methods consist of calculating the fluxes of physical quantities at the boundary of the cells by solving a Riemann problem (i.e. a set of hyperbolic partial differential equations with initial conditions that have a single discontinuity in the domain) for each mesh element.
Still, Eulerian methods present problems in mixing, and complex algorithms to locally increase the grid resolution in the region of interest (Adaptive Mesh Refinement, AMR for short, Berger & Colella 1989) are needed if an accurate set of results is desired. Eulerian codes are also not invariant under Galilean transformations (Springel, 2010b), this might be a problem in cases where high relative speeds ($v \gg c_s$) between entities in the simulations are present, giving rise to artifacts that can compromise the validity of the results.

1.4.2 Lagrangian methods

The Lagrangian method approximates a problem by breaking it down into a finite number of particles, each representing a portion of the system's mass. These particles are then tracked individually, with their motion governed by the relevant equations of the system. Each entity is a particle that carries physical quantities like energy, momentum, and mass and can interact with its surroundings. There are many different approaches that use this scheme, the most used in astrophysics is the so-called Smoothed Particles Hydrodynamics (SPH) technique (Springel, 2010a). In this method particle properties are 'smoothed' over a certain volume using a kernel function, allowing the method to approximate continuous fluid quantities. The main advantage is that the resolution increases with the density of particles, making more denser regions better resolved by default. This comes at the price of poor shock resolution and low order accuracy for contact discontinuities given the fact that density and pressure are smoothed around a particle.

1.4.3 Arbitrary Lagrangian-Eulerian codes

A third approach to discretization of the problems is to merge both the previously discussed methods to obtain a hybrid approach. In this class of numerical schems for hydrodynamics, Arbitrary Lagrangian-Eulerian (ALE; Donea et al. 2004) methods represent a flexible way to combine the advantages of the Eulerian and Lagrangian approaches. The idea of this type of approach is to let the grid move and adapt to different environments by evolving the cell-generating point and then reconstruct the grid at each timestep. This allows a dynamical change in the grid resolution. Implementing a method based on this approach is no easy task, many control algorithms to preserve the regularity of the continouosly updating mesh have to be implemented. Years of research perfected this type of approach, whose use became more widespread for galaxy simulations in the recent past, given its advantages for full adaptivity of the mesh resolution and accuracy in describing fluid discontinuities. The work of this thesis concentrate in implementing a new stellar evolution model in a Milky-Way type galaxy simulation. This simulation is performed using the code AREPO, a cosmological magnetohydrodynamical moving-mesh simulation code. AREPO is an ALE code that uses a moving Voronoi mash to solve hydrodynamical equation with a finite volume approach. In section 1.6 we will describe in detail the features of the code.

1.5 N-body solvers

In astrophysics, the larger the spatial scale examined the more gravity becomes the predominant force. In this sense, the gravitational field (and in particular the one produced by the dark matter component) is the driver of galaxy evolution. Therefore, if we aim to faithfully reproduce in a simulation the evolution of a (Milky Way-type) galaxy, we need to solve for the gravitational potential at each time step to model gravitational dynamics and accurately evolve the system in time.

We are interested in describing how the AREPO code solves gravity since it is the framework in which this thesis work is carried out. AREPO is an ALE code, we follow the dynamics of a set of N particles that generate the gravitational potential (and the hydrodynamic mesh as well). This section will show how to integrate particle evolution through gravitational interaction. The set of equations we need to solve is:

$$\begin{cases} \ddot{\vec{x}}_i = -\nabla_i \Phi(\vec{x}_i) \\ \Phi(\vec{x}) = -G \sum_{j=1}^{N} \frac{m_j}{[(\vec{x}-\vec{x}_j)^2+\epsilon^2]^{1/2}} \end{cases} \tag{1.8}$$

Where $\vec{x}_i$ is the position of the ith particle, $\vec{x}$ is a generic position, m_j is the mass of the jth particle, and ϵ is the softening parameter included to ensure the collisionless behavior of the particles in the simulation. The initial distribution of particles depends on the type of simulation we want to carry out, if we are to simulate a Milky Way-type galaxy in an isolated environment we use an initial condition that resembles a galaxy at equilibrium.

The system of equations is divided into two parts, a dynamical part (the first equation) and the potential (the second equation). The first equation is Newton's second law with the only force acting on the particle being gravity. A widely known theorem called first-order reduction states that all ordinary differential equations (ODE) of order n can be written as a system of n first-order differential equations. In this sense, we can rewrite the first equation in (1.8) as:

$$\begin{cases} \dot{v}_i = -\nabla\Phi(x_i) \\ \dot{x}_i = v_i \end{cases} \tag{1.9}$$

Where $\dot{v}_i$ is the time derivative of the ith particle's velocity vector, Φ is the gravitational potential and $\dot{x}_i$ is the time derivative of the ith particle's position. AREPO solves equations (1.9) using a second order leapfrog scheme (Weinberger et al., 2020). To solve the numerical system (1.9) we need to compute the potential first. The straightforward method would be the so-called direct summation: at the position of a particle, we calculate the potential acting on it summing directly all the contribution of all the other particles in the simulation. The problem of this method is that the computation needed to calculate the potential is proportional to N^2 where N is the number of particles. This poor scalability makes the computation unfeasible when we enhance the number of particles to increase the simulation resolution. It is possible to improve scalability using algorithms that approximate the contribution of groups of cells. These are the Particle-mesh and the Tree algorithms, that will be described in the next subsections.

1.5.1 PM-Method

(From slides of HPC course, Baldi) The particle-mash (PM) method (R.W Hockney, J.W Eastwood, 1988) allows us to compute the gravitational potential in a faster way compared to direct summation by defining a discrete grid inside the simulation domain and assigning to each grid element a mass density. More specifically we exploit the the properties in Fourier space of the Poisson equation (1.2). Assuming the initial density distribution, we can write down equation (1.2)'s Green function. In general, for a linear differential equation, we can write the solution as the convolution of the source term of the field (in our case the density field) and the Green function:

$$\Phi(\vec{x}) = \int g(\vec{x} - \vec{x}')\rho(\vec{x}')\,d\vec{x}', \tag{1.10}$$

where $\Phi(\vec{x})$ is the gravitational potential, g is the green function of the equation and ρ is the mass density distribution. One of the key results of Fourier analysis is that the Fourier transform of a convolution is simply the multiplication of the Fourier transform of the functions being convolved, equation (1.10) then becomes $\hat{\Phi}(\vec{k}) = \hat{g}(\vec{k}) \cdot \hat{\rho}(\vec{k})$, where the hats indicate the Fourier transform. To get the potential we need to compute the Fourier transform of the density field, then multiply it with the Fourier transform of the Green function to obtain the Fourier transform of the potential, and finally compute the inverse Fourier transform of the $\hat{\Phi}(\vec{k})$. Since we are working with discrete particles, we need to evaluate a continuous density field in the first place. Doing this requires initializing a discrete grid and assigning to each grid cell a value of such a continuous density field.

We first define for each particle a shape function called $S(\vec{x})$, which is the volume occupied by a particle in which we have mass density distribution. After we define a shape function for each particle, we search for all the grid cells the intersection between the cell itself, and the shape function of all the particles. In this way, we compute the density field. After we get the density field we can proceed with the algorithm and evaluate the gravitational potential. The great advantage of this method is the fact that the number of computations needed scales linearly with the number of the grid cells, this allows to implement finer grid without an overwhelming computational cost. Still, at scales comparable to the grid length, a considerable amount of information on the particle distribution at small scales is lost due to the grid approximation. This might be a problem if we aim to study highly non-linear regimes in which the density field span over many orders of magnitude. The PM algorithm is well suited for the computation of large almost uniform density field but might fail in situations where large ranges in density values or substantial particle clustering are present.

1.5.2 Tree-method

A different method to solve the N-body problem is the so-called Tree algorithm (Barnes & Hut, 1986). In this algorithm, the simulation domain is recursively divided into subdomains (tree nodes) forming the different levels of a hierarchical tree structure. Firstly, the algorithm divides the domain into eight identical cubes, for each particle a flag indicates to which cube it belongs. This is the level 1 assignment, then the procedure is iterated for each cube of level 1. So each cube is divided into 8 smaller cubes and for each particle, a flag indicates to which cube is assigned, this is a level 2 assignment. This procedure is iterated M-times until every cube in level M has just one particle. After the algorithm finishes every particle has an "address" indicated by the nodes it occupies on the corresponding level. In this way, the algorithm can choose, depending on the distance to the target position for which the acceleration is needed, to approximate the gravitational potential of an entire level and all the particles in it with a multipole expansion. The accuracy of the approximation can be adjusted by defining what "distant" means. This is done by setting a threshold for the angle under which a given tree node (i.e. a group of particles) is "seen" from a target particle on which the potential is to be computed. Such a threshold is called an opening angle. This procedure is carried out for all the particles.

The main advantage of this method is the fact that no intrinsic restriction to the dynamic range is present; in fact, spatial resolution automatically increases in regions where particles cluster. The main disadvantage of the tree method is that for highly homogeneous matter distributions the almost vanishing force on each particle is the result of the cancellation of many larger contributions. This makes it numerically expensive to obtain high accuracy in the force calculation in this particular configuration. The number of operations needed to compute the gravitational acceleration scales as O(N log[N]), with N indicating the particle number, which is far better than direct summation, but worse than the PM method.

1.5.3 Tree-PM Method

In order to overcome the disadvantages of the Tree method, and to exploit the computational efficiency of the PM method, many N-body codes (including AREPO) implement a combination of the Tree and PM algorithms, also known as the Tree-PM method (Bode & Ostriker, 2003). In this algorithm the Fourier transform of the potential $\phi_k = \frac{4\pi G}{k^2}\rho_k$ is split in a long range potential $\phi_k^{\text{long}} = \phi_k \exp(-k^2 r_s^2)$, computed using the PM method (r_s is the force splitting scale), and a short range part $\sigma_k^{\text{short}} = \phi_k\big[1 - \exp\big(-\frac{k^2}{r_s^2}\big)\big]$, computed using the tree method after we perform a inverse Fourier transform. In this way we exploit the accuracy of the tree algorithm for small distances, but we can still perform more accurate and fast calculation of long range forces using the PM method. All of this comes at the cost of some inaccuracy in the potential value when the distance is equal to r_s, which is chosen to be larger than the PM cell size so that the error on the gravitational force can be minimized.

1.6 AREPO

AREPO is multipurpose gravitational and finite-volume magnetohydrodynamical (MHD) moving mesh code for cosmic structure formation and more general astrophysical problems (Weinberger et al., 2020). AREPO is one of the most up-to-date gravitational solvers that can solve magneto-hydrodynamical equations. Several important simulation projects use AREPO as the backbone of their work. This code has been used by many researchers through the years, from seminal works like the Illustris simulation (Vogelsberger et al., 2014) to the next generation of such project, IllustrisTNG (Pillepich et al., 2018) and also the Auriga simulations (Grand et al., 2017). As anticipated, AREPO is an ALE code, each run initializes a set of discrete particles that sample the initial conditions. For a galaxy simulation, these would be all the components of the galaxy, namely dark matter, stars, and gas. From these points, and only for the particles representing the gas, a grid is generated, using the Voronoi tessellation (see 1.6.1 for details). The physical properties of these gas particles, such as mass, momentum, and energy, are spread over the volume of the generated cell, in this sense AREPO uses a finite volume approach to solve hydrodynamics. Each particle in the simulation is assigned with a specific type, and each type is treated differently by the code. For instance, in a galaxy formation simulation performed with SMUGGLE we have:

- **type 0**: The particles of this type are considered to represent gas. Only for this type hydrodynamics is computed (see subsection 1.6.2) at the interface between other particles of the same type.
- **type 1**: These are dark matter particles that provide most of the gravitational potential well.
- **type 2 and 3** these are particles counted as ”old” stars from the bulge and the disk, respectively. They are treated as non-collisional components not undergoing stellar evolution.
- **type 4**: these are stellar particles that are initialized after a star formation event. These stars go through evolution and affect the ISM via feedback processes (see section 1.8).

Depending on the type of particle, different calculations are performed. For types 1, 2, and 3 AREPO solves the equations of collisionless particle dynamics in an expanding flat spacetime described by the Friedmann–Lemaître–Robertson–Walker metric. The calculations are performed in comoving coordinates. In this way, we can select in the run options the comoving edge of the simulation box that will remain constant through the time evolution. The cosmological evolution of the simulation is calculated assuming a ΛCDM flat cosmology, in line with the current best-fitting cosmological model. The code solves the evolution of non-collisional components, stars, and dark matter, integrating the non-collisional Boltzmann equation, also known as the Valsov equation:

$$\frac{df}{dt} = \frac{\partial f}{\partial t} + \frac{\partial f}{\partial \vec{x}}\frac{\vec{v}}{a} + \frac{\partial f}{\partial \vec{v}}\left(\frac{\nabla\Phi}{a^2} + \frac{\dot{a}}{a}\vec{v}\right) = 0 \tag{1.11}$$

Where a is the scale factor, f is the phase-space distribution function, $\vec{v}$ is the velocity of the particle, and Φ is the gravitational potential. The latter is calculated using a Tree-PM algorithm similar to the one described in subsection 1.5.3.

Astrophysical gas is collisional, in order to solve for this component we need to write the hydrodynamic equations, AREPO allow also the implementation of magnetic fields. For type 0 particles AREPO solves the equations of (magneto)hydrodynamics using a second-order accurate finite-volume discretization with ρ, $\vec{v}$, and $\vec{B}$ stored as properties of the cell at its center averaged over the volume. Gradients are estimated with the corresponding values for neighboring cells. The mesh is generated using a Voronoi tessellation (see subsection 1.6.1). Using the gradients, the primitive variables are extrapolated to all mesh interfaces, for which fluxes are calculated using a second order Godunov method at each interface (see subsection 1.6.2).

1.6.1 Computational mesh

To calculate the fluxes of physical quantities between points we need to define a mesh. The mash interfaces are determined by a Voronoi mesh. Every point that discretizes the problem generates a Voronoi cell, we then call these points "mesh-generating point". For each point, its correspondent cell is defined as the region of space that is closer to the mesh-generating point. In this way, mesh interfaces are orthogonal to the segment that connects two neighboring mesh-generating points, and intersects the segment in its middle point. To compute the mesh, the code first finds all the segments connecting neighbor points, it then uses a Delaunay triangulation to build the mesh. Indeed, it is possible to demonstrate that the Delaunay triangulation is the topological dual of the Voronoi tessellation so that after the triangulation is completed, finding the mesh interfaces requires only some geometrical calculations (Weinberger et al., 2020). This type of triangulation forms a set of tetrahedra in which the vertex are the cell-generating points (Fig. 1.2), inside each generated tetrahedron no other points are found. The algorithm to compute the triangulation following these steps. At first, it defines a tetrahedron that encompasses all the mesh-generating points, then at each step adds one mesh-generating point with connections from the point to the corners of the tetrahedron the point falls in (if the point is the first inserted connect the vertices of the enclosing tetrahedra, otherwise connect it to the vertices of the smaller tetrahedra generated by points already inserted).

For each newly generated tetrahedron, it computes the sphere that encompasses that tetrahedron, if a mesh generating point falls in this sphere, remove the tetrahedron. Then iterate this procedure for all the points left. Once the Delaunay triangulation is completed, all mesh-generating points are part of the Delaunay mesh, and the Voronoi mesh is obtained as its topological dual.

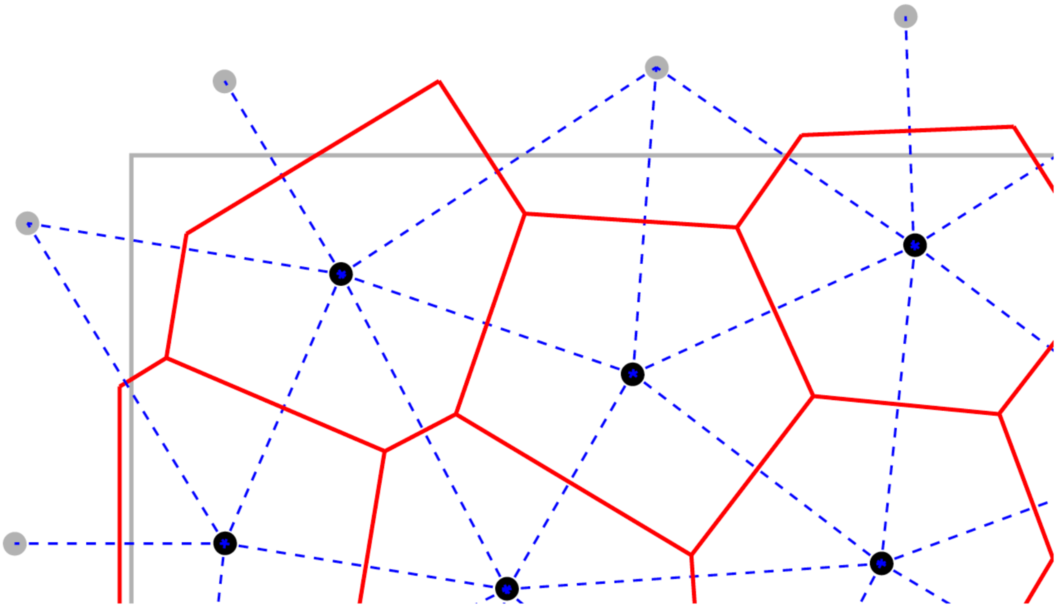


Figure 1.2: Visual representation of Voronoi tessellation (red lines) in 2D. The Delaunay triangulation is represented with the blue dotted lines. Notice how no cell generating point falls in the Delaunay triangles. In 3D the triangles become tetrahedra (from Springel 2010b).

AREPO is an ALE code in the sense that cells move in a fluid-like manner. Each cell moves with the same bulk velocity of the generating point, that depends on all the forces acting on it (gravity, viscous, Lorentz, etc.). The generating point motion can lead to very irregular cell shapes, leading to increased numerical artifacts (Springel, 2010b). In AREPO a routine to prevent this from happening is present. First, the code identifies the irregular shape cells, this is done by calculating the max angle by which any of the cell's faces are seen from its generating point. The angle is calculated as $\alpha_{\rm face} = \left(\frac{A_{\rm face}}{\pi}\right)^{1/2}/h_{\rm face}$ where A_{face} is the area of a given interface and h_{face} is the distance from the mesh-generating point to the interface. The code defines $\alpha_{\rm max} = \max(\alpha_{\rm face})$. If α_{max} is higher than a given threshold (0.75 β, with β being a free parameter) the mesh-generating point is shifted by a fraction $f_{shaping}$ of the characteristic speed of the cell toward the cell's center of mass. The corrective velocity is parameterized as:

$$\vec{v}_{\text{corr}} = \begin{cases} 0 & \text{for } \alpha_{\text{max}} \leq 0.75\beta \\ f_{\text{shaping}} \frac{\alpha_{\text{max}} - 0.75\beta}{0.25\beta} v_{\text{char}} \hat{n} & \text{for } 0.75\beta < \alpha_{\text{max}} \leq \beta \\ f_{\text{shaping}} v_{\text{char}} \hat{n} & \text{for } \alpha_{\text{max}} > \beta \end{cases} \tag{1.12}$$

where v_{char} is the modulus of the bulk velocity of the mesh generating point and $\hat{n}$ is the versor of the characteristic speed. In this way, the bulk velocity of the particle is corrected to avoid irregular cell shapes.

1.6.2 Hydrodynamic solver

In this subsection, we describe the procedure used by AREPO to solve hydrodynamical equations. AREPO uses a second order finite volume approach. In this sense, the primitive quantities we introduce in eq. (1.5) are defined as averages inside each grid cell, this allows to recover integral quantities as:

$$Q_i = \begin{pmatrix} m_i \\ \vec{p}_i \\ E_i \end{pmatrix} = \int_{V_i} \mathbf{U} \, dV_i. \tag{1.13}$$

Where $\mathbf{U}$ is defined in eq. (1.5). If we integrate eq. (1.7) and use Gauss theorem to transform divergence integral in a surface integral we can write the variation of the integral quantities as:

$$\frac{d\mathbf{Q}_i}{dt} = -\int_{\partial V_i} \left[\mathbf{F}(\mathbf{U}) - \mathbf{U}\vec{w}^T \right] d\mathbf{n}, \tag{1.14}$$

where $\vec{w}$ is the velocity of the interface between cells. The cells of the Voronoi tessellation have a flat interface between nearby elements. A_{ij} is the oriented area of the interface between cells i and j (pointing from i to j). The average flux of a conserved quantity across this area is:

$$F_{ij} = \frac{1}{A_{ij}} \int_{A_{ij}} \left[\mathbf{F}(\mathbf{U}) - \mathbf{U}\vec{w}^T \right] dA_{ij}. \tag{1.15}$$

In this way we can write equation (1.14) over the discretized cell faces as:

$$\frac{d\mathbf{Q}_i}{dt} = -\sum_j A_{ij} \mathbf{F}_{ij}. \tag{1.16}$$

With this equation in mind we can write also the equation for a discrete time step as:

$$\mathbf{Q}_i^{(n+1)} = \mathbf{Q}_i^{(n)} - \Delta t \sum_j A_{ij} \mathbf{F}_{ij}^{(n+1/2)}. \tag{1.17}$$

Calculating the fluxes F_{ij} has been one of the main focus of numerical analysis in the past years. In order to make the solution stable a second order method is needed, AREPO uses the Godunov second order method MUSCL Hancock scheme (more detail on the procedure can be found in van Leer Bram 1984). To solve the discretize equation (1.17) the code follows these steps:

(i) Calculate the Voronoi grid based on the position of the cell generating points.

(ii) For each cell i, calculate the primitive quantities $\vec{W}_i = (\rho_i, \vec{v}_i, P_i)$.

(iii) Calculate the gradients of primitive quantities of each cell between all the neighbors: $\nabla\vec{W}_i$.

(iv) Give each cell a velocity $\vec{w}_i$.

(v) Determine the timestep using the Courant criterion.

(vi) Compute for each cell the fluxes $\mathbf{F}_{ij}$.

(vii) Update the primitive quantities using equation (1.17).

(viii) Move the mesh generating points using their previously calculated speed w_i.

We already discussed how the Voronoi mesh is initialized in subsection 1.6.1, and we mentioned the MUSCL scheme used to calculate the flux $\mathbf{F_{ij}}$. In (iii) the code extrapolates the gradients of primitive quantities (ρ, P, and $\vec{B}$, stored at the center of any given cell) between each neighbor grid element. To do this the code adopts the least-square gradient estimate presented in Pakmor et al. (2016). Suppose W_i is a primitive quantity in cell i and $\tilde{W}_j$ is the same primitive quantity in cell j. We can estimate the gradient ∇W_i by imposing:

$$\tilde{W}_j = W_i + d_{ij}\nabla W_i \tag{1.18}$$

where d_{ij} is the distance between cell points. Since every cell has more than a neighbor cell, the estimation of the gradient will be different depending on the particular neighbor used. However, we can determine a more accurate gradient estimate by minimizing the residuals:

$$S_{tot} = \sum_j g_j(W_j - W_i - d_{ij}\nabla W_i)^2 \tag{1.19}$$

where $g_j = A_{ij}/|d_{ij}|$ are the weights with A_{ij} area of the interface between cell i and j. Then the code in (iv) calculates the speed of each cell $\vec{w}_i$. The easier way to accomplish this is to set $\vec{w}_i = \vec{v}_i$ so that the speed of the cell is the speed of the fluid in the cell.

This might not capture the fact that interfaces between cells can move at speeds that are different from their centers, making the code non-purely lagrangian since there might be mass exchange between cells. Still, this mass exchange is in general small compared to the mass content of the involved cells and a quasi-lagrangian behavior is retained. The main problem is that after some iteration the grid may present (high) topological distortions that must be corrected (see subsection 1.6.1).
To calculate the timestep AREPO uses a CFL criterion and the timestep is evaluated as:

$$\Delta t_{CFL} = C_{CFL} \frac{R_i}{c_i + |\mathbf{v}_i'|}, \tag{1.20}$$

where R_i is the effective radius defined as $R_i = \left(\frac{3V_i}{4\pi}\right)^{1/3}$, assuming the cell is spherical with volume V_i. C_{CFL} is the Courant–Friedrichs–Levy coefficient (usually $C_{CFL} \approx$ 0.4–0.8), c_i is the sound speed, and $|\vec{v}_i'| = |\vec{v}_i - \vec{w}_i|$ is the gas speed relative to the cell speed. After we described how AREPO solves gravity and hydrodynamics we turn our attention to how the galaxy formation sub-grid physics (i.e. all the processes that we cannot capture with hydrodynamic equations with given resolution) is implemented in the SMUGGLE model in the next sections.

1.7 Cooling and heating

Cooling and heating mechanisms are fundamental aspects of the physics of galaxy formation and evolution. Cooling makes the internal energy of a gas cloud diminish via the emission of photons. A cloud that loses energy can then enhance its density and reach a very low temperature ($T \sim 10\ K$). Is common knowledge that colder gas is much more prone to collapse and form stars (Chevance et al., 2022). This means that encoding a correct parametrization of cooling is key if we pursue a realistic representation of galaxy proprietaries. In SMUGGLE cooling is implemented following the approach presented in Vogelsberger et al. (2013) with some extra new mechanisms added. In the code, we find a detailed modeling of all two body mechanisms of emission for a gas that is a mixture of hydrogen and helium. The processes included are: collisional excitation, collisional ionization, recombination, dielectric recombination, and free-free emission (see Katz et al., 1996), Compton cooling off CMB photons (Ikeuchi & Ostriker, 1986), and photoionization from a spatial-uniform UV background (Faucher-Giguère et al., 2009) are also taken into account. Cooling from metal emission lines is implemented as well. These processes are mapped in lookup tables computed with the software CLOUDY (Ferland et al., 1998) and assuming ionization equilibrium for a dust-less and optically thin medium.

The final parametrization for the cooling function is then:

$$\Lambda(T,\rho,z,Z) = \Lambda_p(T,\rho,z) + \frac{Z}{Z_\odot}\Lambda_m(T,\rho,z,Z_\odot) + \Lambda_c(T,\rho,z). \tag{1.21}$$

In eq. (1.21) we have three contributions, Λ_p is the primordial contribution by two body interactions, Λ_m is the cooling from metal lines and Λ_c is the cooling from CMB inverse Compton. Notice how the cooling contribution from metals is calculated at the solar metallicity and is simply rescaled to the actual metallicity of the cooling gas cell. In principle, is possible to implement the contribution of every element, since the single abundances are traced in AREPO. However, it was decided not to implement cooling in this way since the uncertainties on the single element production are large. Finally, the UVB background works as a heating source with the contribution of both background quasars and star-forming galaxies.

In SMUGGLE the possibility of gas cooling via low-temperature metal lines, fine-structure, and molecular cooling processes was added. These processes allow the gas to cool to temperature $T < 10^4\ K$, forming molecular clouds that are the birthplace of stars. The implementation of molecular cooling is based on tables computed with CLOUDY and it is parameterized as follows:

$$\begin{aligned}\Lambda_{\rm mol} = 2.896{\times}10^{-26}\left[\left(\frac{T}{128.215\,\rm K}\right)^{-4.9202} + \left(\frac{T}{1349.86\,\rm K}\right)^{-1.7288} + \left(\frac{T}{6450.06\,\rm K}\right)^{-0.3075}\right]^{-1} \times \\ \left(0.001 + \frac{0.1 n_H}{1+n_H} + \frac{0.09 n_H}{1+0.01 n_H} + \frac{(Z/Z_0)^2}{1+n_H}\right) \times \\ \left(\frac{1+Z/Z_0}{1+0.00143 n_H}\right) \times \exp\left(-\left[\frac{T}{158000\,\rm K}\right]^2 + \right) \ {\rm erg\ s^{-1}\,cm^{-3}},\end{aligned} \tag{1.22}$$

where T is the gas temperature, n_H is the hydrogen number density, and Z and $Z_\odot$ are the gas and the solar metallicities. Self-shielding of molecular gas from ionizing radiation is implemented by multiplying eq. (1.22) by a factor f_{ssh}:

$$f_{ssh} = (1-f)\left[1+\left(\frac{n_H}{n_0}\right)^\beta\right]^{\alpha_1} + f\left[1+\left(\frac{n_H}{n_0}\right)\right]^{\alpha_2}, \tag{1.23}$$

where n_H is the hydrogen number density and (n_0, α_1, α_2, β, f) encode the redshift dependence (see Rahmati et al. 2013, Table A1). There are two important heating mechanisms that are relevant for low temperature gas and are included in SMUGGLE. The first is the heating due to the cosmic ray background. These are implemented based Guo & Oh (2008) as:

$$\Lambda_{CR} = -10^{-16}(0.98 + 1.65)\tilde{n}_e e_{CR} n_H^{-1}\,{\rm erg\ s^{-1}cm^3}. \tag{1.24}$$

In eq. (1.24) $\tilde{n}_e$ is the electron number density in unit of hydrogen number density, and e_{CR} is the cosmic rays energy density that is parameterized as:

$$e_{cr} = \begin{cases} 9 \times 10^{-12}\,\mathrm{erg\,cm^{-3}} & \text{for } n_H \geq 0.01\,\mathrm{cm^{-3}} \\ 9 \times 10^{-12}\,\frac{n_H}{0.01}\,\mathrm{cm^{-3}\,erg\,cm^{-3}} & \text{for } n_H < 0.01\,\mathrm{cm^{-3}}. \end{cases} \tag{1.25}$$

In this way, the fact that cosmic ray heating becomes less and less important for low gas densities is explicitly taken into account. The second heating mechanism that has been implemented in SMUGGLE is the photoelectric emission of electrons by dust grains, called photoelectric heating. The mechanism for this source of heating is thought to be the interaction of the interstellar radiation field and the polycyclic aromatic hydrocarbons (PAHs) molecules found in interstellar dust grains. The parameterization used to model this effect comes from Wolfire et al. (2003) and reads:

$$\Lambda_{phot} = -1.3{\times}10^{-24}\tilde{e}_{\nu}^{pe} n_H^{-1} \left(\frac{Z}{Z_\odot}\right) \times \left(\frac{0.049}{1 + (x_{pe}/1925)^{0.73}} + \frac{0.037\,(T/10^4\mathrm{K})^{0.7}}{1 + (x_{pe}/5000)} \right)\ \mathrm{erg\ s^{-1}\,cm^3}, \tag{1.26}$$

$$x_{pe} = \frac{\tilde{e}_{\nu}^{pe} T^{0.5}}{\Phi_{PAH}\,\tilde{n}_e\,n_H}, \tag{1.27}$$

where $\tilde{e}_{\nu}^{pe}$ n is the photon energy density, normalized to the Milky Way value $\tilde{e}_{\nu}^{pe} = e_{\nu}^{pe}/3.9 * 10^{-14}$ erg cm^{-3} and Φ_{PAH} is a factor that incorporates the uncertainties in the interaction rates between atoms and dust grains in the molecular regime.

1.8 Star Formation and Stellar Feedback

Stars in galaxies play a major role in driving galaxy evolution. They interact with the interstellar medium (ISM), influencing its structure and dynamics, and their presence shapes the appearance of galaxies across almost all wavelengths observable. Therefore, simulating a galaxy requires modeling all the physics associated with the formation, evolution, and death of stars, describing how they interact and modify the ISM in each phase of their evolution. We describe how the ISM can be influenced by stars in sec 1.8. This tight interplay between the ISM and the stellar component of the galaxy imposes modeling stellar evolution as accurately as possible if we want to obtain a realistic representation of the ISM properties and its evolution. As we discussed earlier, AREPO uses an unstructured Voronoi mesh to model gas dynamics. The typical size of a cell in a galaxy formation simulation (typically on the order of $\sim 10\,\mathrm{pc}$) does not allow for the resolution of individual stars and their evolution. This implies that both star formation, evolution, and interactions with the ISM must be implemented as sub-grid algorithms.

We will explain in detail how these aspects related to the stellar component stars are implemented in the SMUGGLE since our work focuses on studying and updating the model for stellar evolution in this framework.

1.8.1 Star Formation

To discuss how a cold gas region is converted into a star-particle we will describe the criterion used in the SMUGGLE model to trigger star formation. SMUGGLE adopts a probabilistic approach (Marinacci et al., 2019). When a star formation event is triggered (see below) a cell of the grid containing gas is converted into a so-called stellar particle. Every star particle represents a simple stellar population (SSP) that follows a Chabrier (2001) initial mass function. Star particles are created by sampling a probability distribution derived from the star formation rate (SFR) computed for each cell, which is given by:

$$\dot{M}_* = \begin{cases} 0 & \text{if } \rho < \rho_{th} \\ \epsilon \frac{M_{gas}}{t_{dyn}} & \text{if } \rho \geq \rho_{th}. \end{cases} \tag{1.28}$$

In eq. (1.28) ϵ is an efficiency factor set to 0.01 based on observations (Krumholz & Tan, 2007), $t_{dyn} = \sqrt{3\pi/32G\rho_{gas}}$ is the gravitational dynamical time and ρ_{th} is a threshold set at $100\,\mathrm{cm}^{-3}$, which is the average density of giant molecular cloud[1]. Another parameter for the conditional collapse is present, this is the so-called viral-parameter α. This parameter restricts star formation only to gravitationally bound regions, preventing gas cells not prone to gravitational instabilities from collapsing in an unrealistic and non-observable way. For the ith cell the alpha parameter is calculated following Hopkins et al. (2018) as:

$$\alpha_i = \frac{\|\nabla \otimes \vec{v}_i\|^2 + (c_{s,i}/\Delta x_i)^2}{8\pi G\rho_i}. \tag{1.29}$$

In eq. (1.29) v_i is the speed of the ith gas cell, $c_{s,i}$ is its speed of sound, Δx_i is the radius of the cell (i.e., the radius of the sphere having the same volume of the cell), ρ_i is its gas density and G is the gravitational constant. We define

$$\|\nabla \otimes \vec{v}_i\|^2 = \sum_{i,j} \left(\frac{\partial v_i}{\partial x_j}\right)^2. \tag{1.30}$$

In the equation (1.29) we find at the numerator terms that are proportional to both kinetic and thermal energy of the gas that prevent the collapse. At the denominator we

[1] In Marinacci et al. (2019) it is discussed how a higher (or lower) value of this threshold does not impact the star formation history nor the Kennicutt-Schmidt relation. However, the precise value of the threshold might change the clustering of SN explosions

have a term proportional to the gravitational attraction, forcing the collapse. A cloud is allowed to collapse and form a star particle only in cells that have $\alpha < 1$. If a particle satisfies the conditions for the collapse described in eq. (1.28) and eq. (1.29) it can become a star particle. To determine if this happens we define $p = 1 - \exp(-\frac{\dot{M}_* \Delta t}{M_i})$, then we extract a random p^* from a uniform distribution $[0, 1[$, if $p^* < p$ the cell is converted into a star particle and a SSP is initialized.

1.8.2 Stellar Feedback

Feedback from supernovae

Supernovae inject energy and momentum into the ISM (e.g., Agertz et al., 2011), they also pollute the gas with newly synthesized elements through nucleosynthesis within stars and during explosions. The momentum injection from SNe drives turbulence (Martizzi et al., 2016) and contributes to galactic-scale outflows (Li et al., 2017). Given the significance of these processes in shaping the ISM, SNe need to be modeled as accurately as possible. The first task is to compute the number of SNe events happening in a simulation timestep. Since stellar particles represent an entire stellar population we can have more than one event in a timestep for a stellar particle. This aspect is an integral part of the new stellar evolution model that we will develop in this thesis work. Therefore, we defer its discussion in section 2.3.1, where we explain and analyze in detail how the code takes into account end-life events from stars.

We assume that for each SNII explosion the energy released is $E_{SN} = f_{SN} E_{51}$ where $E_{51} = 10^{51}$ erg and f_{SN} is an efficiency factor. Assuming that the blast velocity is $v_{SN} = \sqrt{2E_{SN}M_{SN}}$ (derived from the kinetic energy equation), the momentum of the shock is $p_{SN} = M_{SN}v_{SN} = \sqrt{2E_{SN}M_{SN}}$. To accurately capture the effect of a SN blast we need to take into account the first adiabatic phase of the explosion (also known as the Sedov-Taylor phase). During this phase, the momentum imparted to the ISM gas is generated by an over-pressurized central gas bubble that expands and sweeps up the surrounding, unshocked ISM. An accurate implementation of SNII then takes into account this first phase of the explosion even if not explicitly resolved in the simulation. This phase lasts until the shock arrives at the cooling radius (the radius at which the cooling timescale is equal to the shock expansion timescale). In SMUGGLE this effect is accounted for by boosting the momentum ejected in the nearby cells by a factor that depends on the gas thermodynamical state (see Marinacci et al. 2019, for more details). In the model, a prescription for how the energy and momentum are spread in the nearby cells is also present. This is achieved by calculating the weight of each nearby cell.

SMUGGLE first identifies the neighboring cells, so the quantity $N_{ngb} = \frac{4\pi}{3}h^3 \sum_i W(|\vec{r_i} - \vec{r_s}|h)$ is defined, where h is a search radius (coupling radius), W is the standard cubic spline SPH kernel, and $\vec{r_i}$ and $\vec{r_s}$ are the position vectors of the ith gas neighbor and the star particle and N_{nbg}, the number of neighbor cells, is a parameter set a priori (usually $N_{nbg} = 64$). The equation for the neighbor cells can then be solved for h, which represents the radius at which the SN explosion affects the ISM. The calculated feedback radius is then used to compute the distribution of energy and momentum in the neighbor cells by assigning a weight to each cell. The weight of the ith cell is defined as:

$$w_i = \frac{\Delta\Omega_i}{4\pi} = -\frac{1}{2}\left(1 - \frac{1}{\sqrt{1 + \frac{A_i}{\pi|\vec{r_i} - \vec{r_s}|^2}}}\right) \tag{1.31}$$

where $|\vec{r_i} - \vec{r_s}|$ is the distance between the gas cell generating point and the stellar one, A_i is the area of the gas cell defined as $A_i = \pi\Delta x^2$ where Δx_i is the cell radius found in eq. (1.29). The weights are normalized and multiplied by the equations for the total energy and momentum released. The momentum injected is directed radially away with respect to the stellar particle in which the SN explosion occurs.

Radiative feedback

Radiation emitted by stars can interact and influence the ISM through various mechanisms. Young massive stars can ionize the ISM by emitting photoionizing radiation, and radiation can modify the dynamic of the ISM through radiation pressure. In SMUGGLE such effects are also considered. In the model, HII regions are implemented as follows. Each stellar particle has a photoionizing radiation flux

$$N_* = \frac{L_*}{\langle h\nu\rangle} = \frac{\gamma_* M_*}{\langle h\nu\rangle}, \tag{1.32}$$

where L_* is the luminosity of the star particle, γ is the light-to-mass ratio, taken as $\gamma_* = 10^3 \mathrm{L}_\odot/\mathrm{M}_\odot$, and $\langle h\nu\rangle = 17$ eV is the average energy of the photoionizing photons emitted by massive stars. Also for the radiative feedback, we use the coupling radius h introduced in the previous subsection. The quantity of ionizing photons can ionize a mass that is usually smaller than the one contained in the coupling radius. The code then uses a probabilistic approach for ionizing the cell. Each has an ionization probability $p = n_*/(\alpha_{rec} n_H^2 V_i)$, where α_{rec} is the recombination coefficient of hydrogen, n_H is its numerical density, V_i is the cell volume and $n_* = \tilde{w}_i N_*$ is the number of photoionizing photons weighed with the same values defined in eq. (1.31). A random number in $p' = [0, 1[$ is extracted, if $p' < p$ then the cell becomes photoionized for a time

$t_{\rm off}$ equal to the stellar particle timestep, a background temperature $T = 1.7 * 10^4 K$ is set and cooling is disabled. After a cell is photoionized the thermal pressure makes the region expand, injecting momentum in the ISM (not to be confused with the momentum injection by winds and SNe). Besides the effect of ionization, radiation can transfer momentum directly via radiation pressure. In the code this is implemented by considering a momentum injection for the neighbor cell equal to:

$$\Delta p = \frac{L_*}{c} \cdot (1 + \tau_{IR}) \Delta t, \tag{1.33}$$

where the Infrared optical depth $\tau_{\rm IR}$ of the gas as a parameter to take into account the multiple scatterings that photons go through when generating radiation pressure. This radiation momentum is coupled to nearby cells in the same way the SN feedback is, using the same weights.

Feedback from stellar winds

Stars contribute to feedback also when they inject mass, and consequently momentum and energy, in a non-explosive way. Winds from massive O/B type stars are one of the most important sources of feedback from continuous mass loss. Also, the mass return happening in the AGB phase by stars that do not explode as core-collapse supernovae is important, but release their outer layers gently in the ISM forming a planetary nebula. In general, all the stars release part of their mass in the ISM during their lifetime, still the only appreciable contribution to energy, momentum, and mass ejected comes from stars with $M > 20 M_\odot$ (see Fig. 3.16). In the SMUGGLE model, O/B winds are implemented as a function that parameterizes the mass ejected per unit time. We will discuss in detail this aspect of the model in section 2.3.1. While to describe O/B type winds functions that parametrize the injection rate, for the AGB phase we use the same approach used to calculate both the number and mass ejection of SNII, presented in Vogelsberger et al. (2013) (see again section 2.3.1 for more details).

Stellar wind feedback is provided by the injection of both momentum and energy in the ISM from continuous mass losses. The energy injected by winds using the prescription (see also Hopkins et al., 2018):

$$E_{\rm winds} = \Delta t L_{\rm kin} = M_{\rm loss} \times \psi \times 10^{12} \text{ erg g}^{-1} \tag{1.34}$$

where $M_{loss} = (M_{tot}(t + \Delta t) - M_{tot}(t))$ is the mass loss by the stellar particle through winds in the interval Δt and ψ is a function of time that describe the wind speed as a function of the lifetime of the SSP. The momentum injection is simply derived from the energy of the winds as $p_{\text{winds}} = \sqrt{2M_{loss}E_{\text{winds}}}$. The same formalism for the coupling radius and weighting used for SN feedback is also applied for winds. It is also worth noting that in the code O/B wind injection and the associated feedback are only performed if the returned mass over a given time step is larger than 10^{-4} times the mass M_* of the stellar particle at birth. If this is not the case, the mass loss is accumulated until the first time step at which the mass return threshold for this channel is reached.

2

Methods

2.1 Introduction

This chapter describes different approaches to implementing stellar evolution models in hydrodynamical simulations of Milky Way-type galaxies and galaxies in general. Stellar evolution provides energy, momentum, and mass and alters the chemical composition of the ISM via stellar winds and supernovae, thus affecting the physics of the interstellar gas. Therefore, having an accurate numerical model for stellar evolution directly impacts the outcome of such a simulation.

In this chapter, we will first reproduce the stellar evolution modules implemented in two different widely used galaxy formation physics models, namely Fire-3 (Hopkins et al., 2022) and SMUGGLE (Marinacci et al., 2019). Then, in the next chapter, we will construct a new stellar evolution model to be implemented in SMUGGLE using the code SEVN (Spera et al., 2015, 2019; Iorio et al., 2023). The final goal is to compare, gain insights, and interpret the evolution of a Simple Stellar Population implemented in different numerical codes using various techniques. A quantitative analysis of each model, which we will provide in this chapter, is needed to achieve this.

We first examine how stellar evolution is implemented in the Fire-3 model. This is done by reproducing the stellar evolutionary plots shown in Hopkins et al. (2022) using the analytical prescriptions presented in that work. We then investigate the current stellar model used in the SMUGGLE model, based on the approach described in Vogelsberger et al. (2013). Contrary to the Fire-3 model, in which Supernovae rates and ejection rates from winds are described by analytical functions, the approach used in SMUGGLE consists of encoding stellar evolution in a set of .hdf5 tables. The code then derives the information needed by interpolating the data sampled in these tables (see 2.3.1 for an in-depth discussion). Finally, we describe the procedure to build a new model using the stellar evolution code SEVN (Iorio et al., 2023). We build this model by simulating the evolution of single stars (i.e. no binary stars are considered) and creating a new set of .hdf5 tables that are directly readable by the SMUGGLE code. A detailed comparison and a complete discussion of the different models are made in the next sections and chapters.

2.2 Fire-3

In Fire-3 (Feedback In Realistic Environments, Hopkins et al. 2022) parametrization of SNe rates and continuous mass loss rate are provided. Therefore, it is relatively straightforward to reproduce in a standalone code stellar evolution model[1]. We will do so in this section, where we will also briefly discuss the salient features of the Fire-3 model.

In Fire-3, stellar feedback input to the ISM is computed directly by passing the age and metallicity to fitting functions that represent, respectively, the mass rate ejected by the OB/AGB stellar winds and type II and type Ia supernovae rates. To obtain these functions, the model uses software that simulates the evolution of an entire stellar population given the metallicity, stellar evolutionary tracks, and IMF. Stellar population evolution is calculated using the code STARBURST99 (Leitherer et al., 1999) with a Kroupa (2001) IMF with slopes $(0.3, 1.3, 2.3)$ in the mass intervals $(0.01 - 0.08, 0.08 - 0.5, 0.5 - 100)$ $\mathrm{M}_{\odot}$, respectively, with SNII threshold 8 $\mathrm{M}_{\odot}$, adopting updated Geneva tracks as stellar evolution tracks. The STARBURST99 code creates files containing the lifetimes of stars given their mass, the mass ejected in the last phase of the evolution, and the continuous mass loss throughout all evolution. Fitting the output data from STARBURST99 as a function of stellar population age provides the analytical parametrization of the rate functions used as input for stellar evolution and feedback in Fire-3. The rates are normalized to the solar mass and, during the simulation, the mass of the stellar particle is recalculated every timestep to obtain the updated rate value. In the next subsections, we will describe each feedback channel implemented in Fire-3.

[1] The code used is available at `https://github.com/lorevannini/Thesis-project/tree/main/FIRE-3`

2.2.1 Supernovae

SNII are the main momentum, energy, and α-element injectors. In the Fire-3 model their rate per unit stellar mass formed is parameterized as:

$$\frac{R_{CC}/M_*}{Gyr^{-1}\,\mathrm{M}_\odot^{-1}} = \begin{cases} 0 & (t < t_{S,1} \text{ or } t > t_{S,3}) \\ a_{s,1}(t/t_{s,1})^{\psi_{s,1}} & (t_{s,1} t < t_{s,2}) \\ a_{s,2}(t/t_{s,2})^{\psi_{s,2}} & (t_{s,2} \le t \le t_{s,3}) \end{cases} \tag{2.1}$$

where $\psi_{s,1} = \ln(a_{s,2}/a_{s,1})/\ln(t_{s,2}/t_{s,1})$ and $\psi_{s,2} = \ln(a_{s,3}/a_{s,2})/\ln(t_{s,3}/t_{s,2})$ with $(a_{s,1},\ a_{s,2},\ a_{s,3}) = (0.39,\ 0.51,\ 0.18)$ and $(t_{s,1},\ t_{s,2},\ t_{s,3}) = (3.7,\ 7.0,\ 44)$ Myr.

SNIa, that represent the main channel for Fe enrichment, are parameterized by the simpler expression:

$$\frac{R_{Ia}/M_*}{Gyr^{-1}\,\mathrm{M}_\odot^{-1}} = \begin{cases} 0 & (t < t_{Ia,1}) \\ a_{Ia,1}(t/t_{Ia,1})^{\psi_{Ia,1}} & (t_{Ia,1} \le t) \end{cases} \tag{2.2}$$

with $t_{Ia,1} = t_{s,3} = 44\ Myr$, $\psi_{Ia,1} = -1.1$ and $a_{Ia,1} = 0.0083$

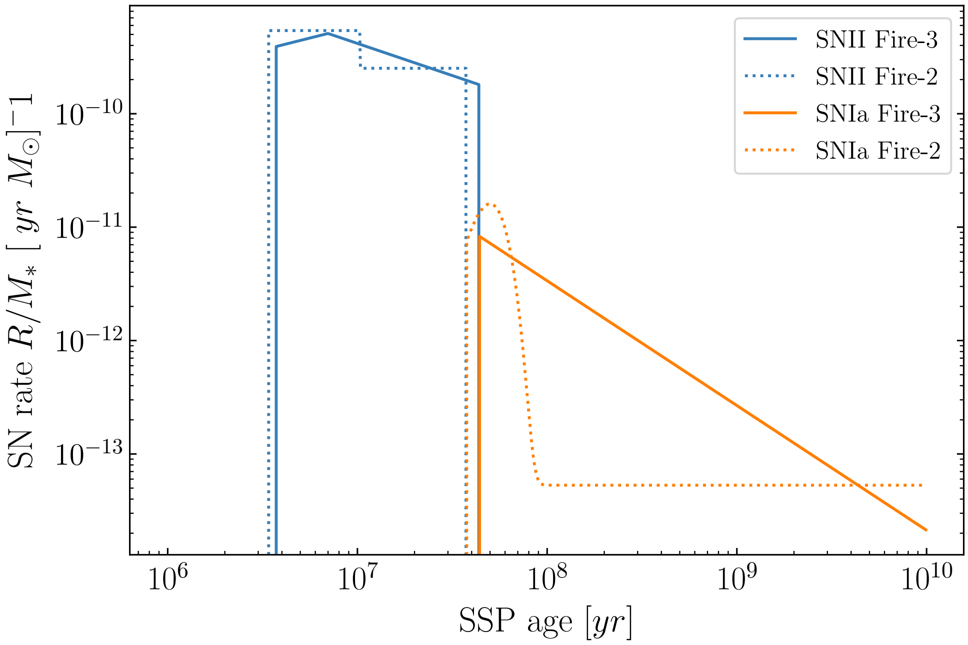


Figure 2.1: Parametrization of Type II Supernova rate from (2.1) and Type Ia Supernova rate from (2.2). We plot also the functions given in Fire-2, an earlier incarnation of the FIRE model, as a reference. The expression of Fire-2 fitting formulas can be found in Hopkins et al. (2018).

In Fig 2.1 we notice how SNII explodes with a rate ~ 2 orders of magnitude higher than the SNIa. The parametrization for type 2 Supernovae has been refined from the second version of the FIRE model (dotted lines in the figure) using two power laws to remove the discontinuity at $t \sim 10$ Myrs. While estimating the SNII explosion rate is in principle straightforward (see 2.3.1), estimating the SNIa rate is a much more challenging task. Indeed, the SNIa rate depends in principle on the progenitor population of SNIa, and on the stellar and binary evolution of that population (Maoz & Graur, 2017). Both of these factors are affected by great uncertainties and it is for this reason that SNIa rates are usually modeled using a so-called delay-time distribution (DTD). To give an estimate of the SNIa rates various methods have been developed, but the basic idea is that the (cosmological) volumetric rate of SNIa can be obtained by convolving the star formation history (SFH) of the Universe with a DTD function. In Fire-3, the DTD form was updated from the one prescribed in Mannucci et al. (2006) (used in Fire-2) to Maoz & Graur (2017), the latter being more consistent with the latest measurements of the cosmological SNIa rate. In this way, the DTD function becomes a pure power law starting when SNII stops exploding, while the old parameterization had a peak at $\sim 10^7$ yr and then stabilized to a constant value. Those trends are visible in Fig. 2.1.

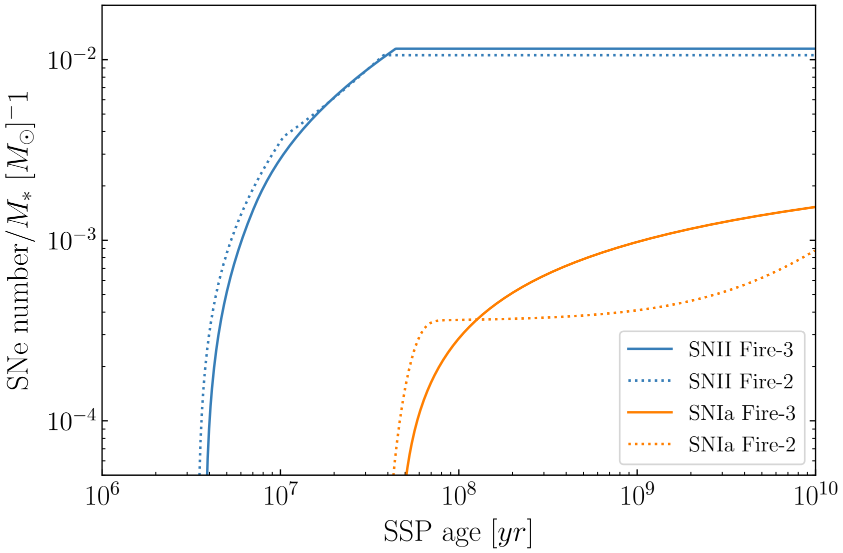


Figure 2.2: Total number of SN per unit stellar mass expected in Fire-3 (solid lines) and Fire-2 (dotted lines). The total number of events expected is fundamental to estimating the energy injected in the ISM. In the SNIa the different DTD used leads to a visible difference in the cumulative function. For SNII instead, the difference is much less evident; this is because the number of events expected depends only on the form of the IMF and the star's lifetime.

To obtain the cumulative SN number expected per unit stellar mass as a function of stellar population time, we initialized 1000 points for time with an equal log spacing, and then we integrated eq. (2.1) and eq. (2.2) using the trapezoidal method. Notice how in Fig. 2.2 changing the evolution model changes the total number of SNII exploding. This is an important quantity that depends only on the lifetime of the stars and on the IMF (eq. 2.7). We expect slightly different results for different IMF and different lifetimes of stars used. SNe are single events, this means that the rate does not represent the actual number of SN happening but rather a probability of an SNII to happen (see section 2.3.1 for a more in depth discussion).

After determining the rates and the total events we are interested in quantifying the total mass returned by the SSP in this model. This is a key aspect that might impact the outcome of a galaxy formation simulation since the gas returned is not only reusable to make new stars but also has a different chemical composition that changes the evolution of the entire galaxy. We can calculate the mass ejection rate in Fire-3 using the function that describes the mass ejected by a single SN event as a function of time: $M_{ej} = 10\,\mathrm{M}_{\odot}\big(t/6.5\mathrm{Myr}\big)^{-\psi_{CCM}}$ with $\psi_{CCM} = 2.22$ for $t \leq 6.5$ Myr and $\psi_{CCM} = 0.267$ for $t > 6.5\ Myr$ (Fig. 2.3).

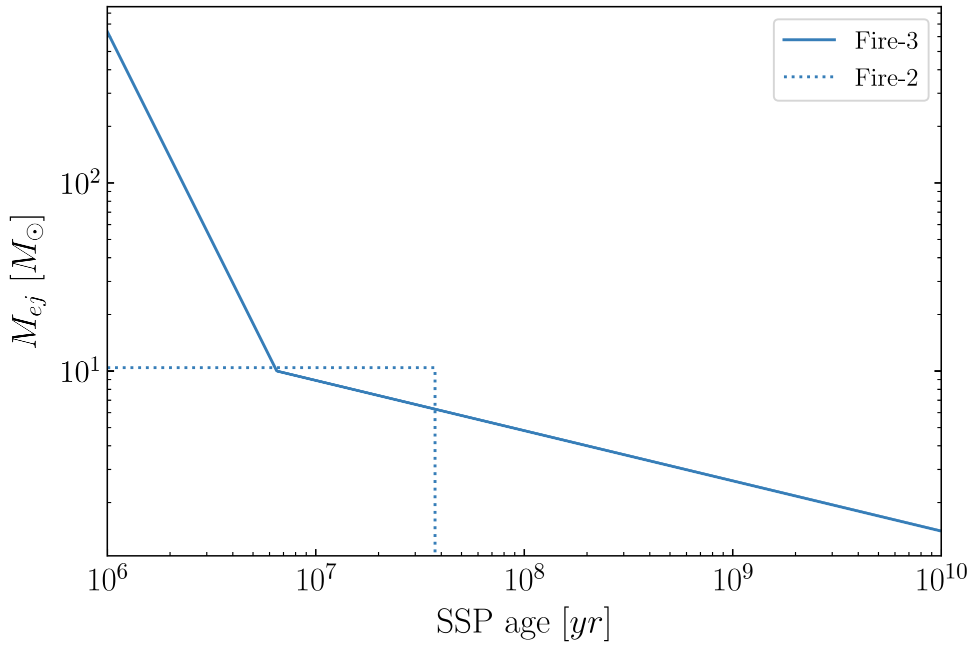


Figure 2.3: Mass ejected for single event (M_{ej}) over time for Fire-3 (solid line) and Fire-2 (dotted line). Fire-2 takes the mass ejected as constant

.

The function reported in Fig. 2.3 tells us the mass ejected by the star that ends its life at time t. We notice how in Fire-3 they improve the model by assuming a double power law to modelize the mass ejected by a star dying at time t, while in Fire-2 it was simply the IMF averaged mass ejected by all the SNII. If we multiply the SNII rate in Fig. 2.1 with the mass loss for a single event in Fig. 2.3 we obtain the mass injection rate per solar mass.

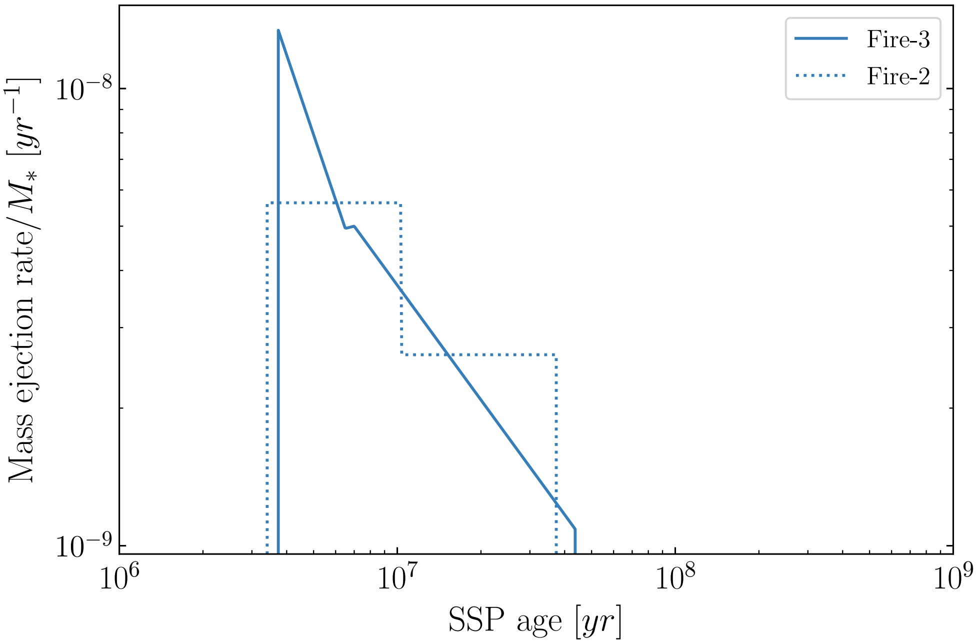


Figure 2.4: Mass ejection rate normalized to the stellar particle mass. While in the Fire-2 (dotted lines) the mass ejected for supernova is constant at $M = 10.4\,\mathrm{M}_\odot$ in Fire-3 (solid lines) this has been modulated depending on the initial mass of the star that explodes.

It is assumed that SNIa explode roughly when Chandrasekhar mass is reached, since no remnant is left we assume the mass injection as constant with a value of 1.4 $\mathrm{M}_\odot$ for a single event. This approximation is made since we have uncertainties on the progenitors channel that leads to SNIa explosion in the first place. To interpret best the results of galaxy formation simulations in which these stellar models are implemented we need to calculate the total cumulative mass released by SNe as a function of time. This quantity tells us how much mass becomes reusable after a given time has passed from the star formation event. Different stellar evolution implementations will inevitably predict different mass fractions, allowing us to compare models. We perform a numerical integral using the trapezoid method of the mass ejection rate (Fig. 2.5) to obtain this result.

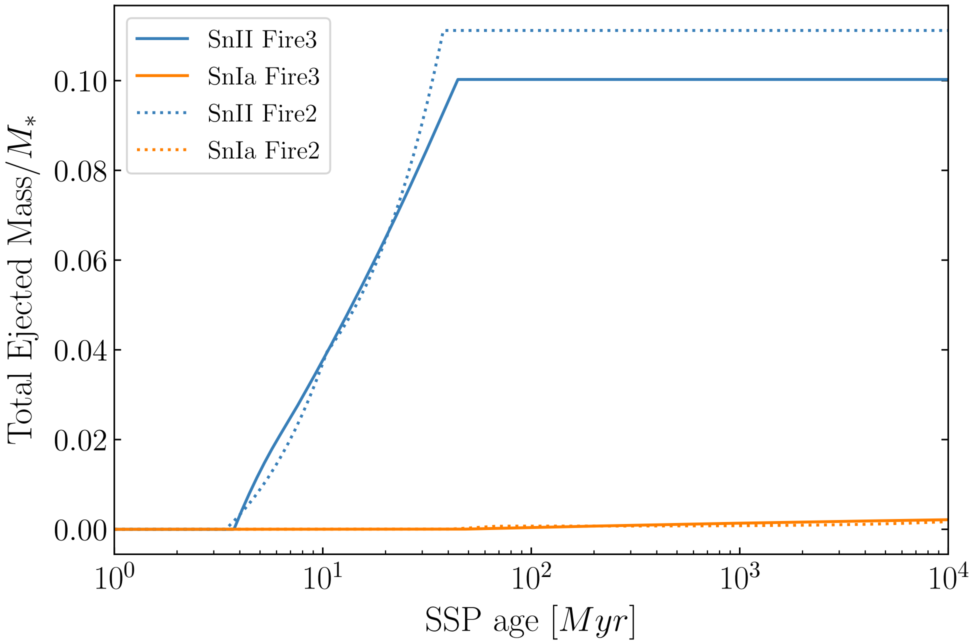


Figure 2.5: Total mass ejected by SNe normalized to the initial mass of the stellar population. Notice how from Fire-2 (dotted lines) to Fire 3 (solid lines) less mass is injected into the ISM, with a 0.15 per cent discrepancy respect tho the total mass of the SSP. Also, SNIa makes up for a very little contribution for the total mass injected.

The model then predicts about 10% of the total initial mass being re-injected in the ISM. Compared to the older version of the Fire model (Fire-2), we notice in Fire 3 a decrease in the total mass fraction after all the SNII exploded. This suggests that newer stellar evolution models predict a lower mass ejection compared to the older ones. We will show that this is indeed the case for other models and implementations.

2.2.2 OB and AGB winds

It is an observational fact that massive stars, especially O/B-type stars (Vink et al., 2001) expel a relevant fraction of their mass during their lifetime. Continuous mass losses have a non-negligible impact on the interstellar medium that needs to be taken into account. Moreover, also low-mass stars that end their life cycle as planetary nebulae are considered part of this continuous mass loss since the energy and momentum release in the ISM does not occur through an explosion. Therefore, the total parametrization for all the continuous mass losses in Fire-3 is given by:

$$\frac{\dot{M}_w/M_*}{\mathrm{Gyr}^{-1}} = \begin{cases} a_{w,1} & (t \leq t_{w,1}) \\ a_{w,1}(t/t_{w,1})^{\psi_{w,1}} & (t_{w,1} < t \leq t_{w,2}) \\ a_{w,2}(t/t_{w,2})^{\psi_{w,2}} & (t_{w,2} < t \leq t_{w,3}) \\ a_{w,3}(t/t_{w,3})^{\psi_{w,3}} & (t_{w,3} < t) \end{cases} + a_{A,1}(t_{A,1}/t)^{1.6}\{[e^{-(t_A/t)^6}+[a_{A,2}{}^{-}1+(t_A/t)^2]^{-1}\}, \tag{2.3}$$

with $\psi_{w,1} = \ln(a_{w,2}/a_{w,1})/\ln(t_{w,2}/t_{w,1})$, $\psi_{w,2} = \ln(a_{w,3}/a_{w,2})/\ln(t_{w,3}/t_{w,2})$, $\psi_{w,3} = -3.1$ $(t_{w,1}, t_{w,2}, t_{w,3}, t_A) = (1.7, 4.0, 20, 800)$ Myr,
$(a_{w,1}, a_{w,2}, a_{w,3}, a_{A,1}, a_{A,2}) = (3 * \tilde{z}^{0.87}, 20 * \tilde{z}^{0.45}, 0.6 * \tilde{z}, 0.11, 0.01)$ where $\tilde{z}$ is $z = 10^{[Fe/H]}$ that takes in account the dependencies of the winds on stellar metallicity. This equation encodes the continuous mass loss from stars. The first part until $\sim 10^9$ yr represents the winds from massive stars, while ages greater than $\sim 10^9$ yr take into account the mass loss from stars entering the AGB phase and then becoming planetary nebulae. We will call this phase of mass loss the AGB phase from now on.
In Fig. 2.6 we plot eq. (2.3). Notice how the mass ejection rate at 10^3 Myr is enhanced by two orders of magnitude due to stars entering the AGB phase.

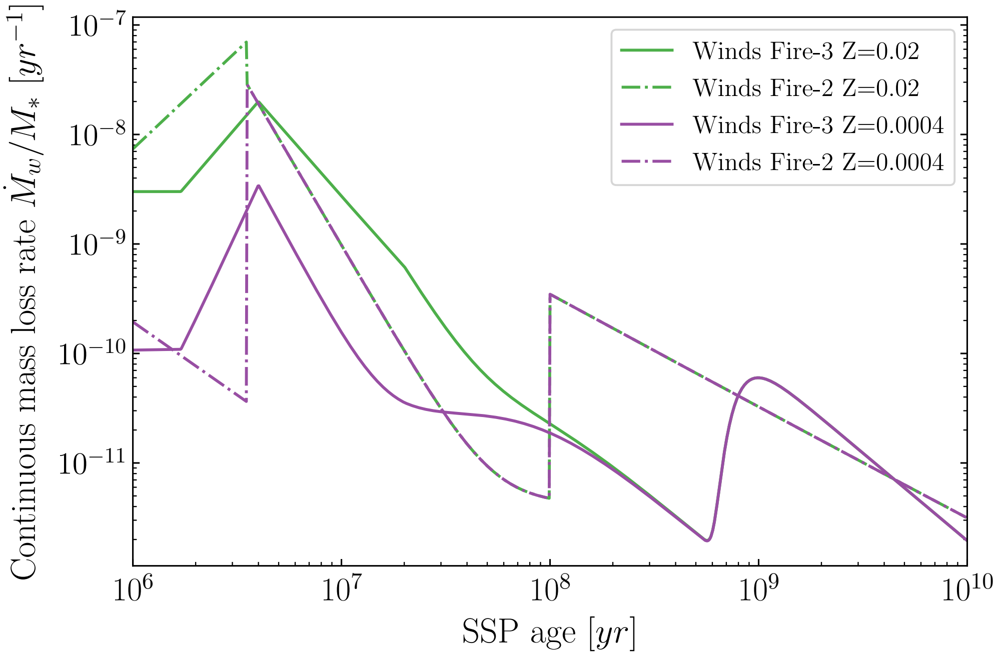


Figure 2.6: Continuous mass loss form stars in main sequence parametrized in equation (2.3) for Fire-3 (solid lines) at two different metallicities. Also in this case, we show the trends for Fire-2 (dotted lines) for comparison.

Changing the metallicity affects the first 3 Myr of the mass ejection rate. O/B stars with less metal content see a decrease in their mass loss rate. Notice how both Fire-2 and Fire-3 present a bump in the rate that takes into account the mass loss from stars that do not explode as Supernovae. Still in Fire-3 the bump is much less prominent and shifts form $\sim 10^8$ to $\sim 10^9$ yr.

As it was done for SNe, we numerically integrated these rates in time to obtain the cumulative mass ejection from continuous losses. The result is presented in Fig. 2.4. Even in this case, we note the sudden bump due to the AGB at 10^3 Myr.

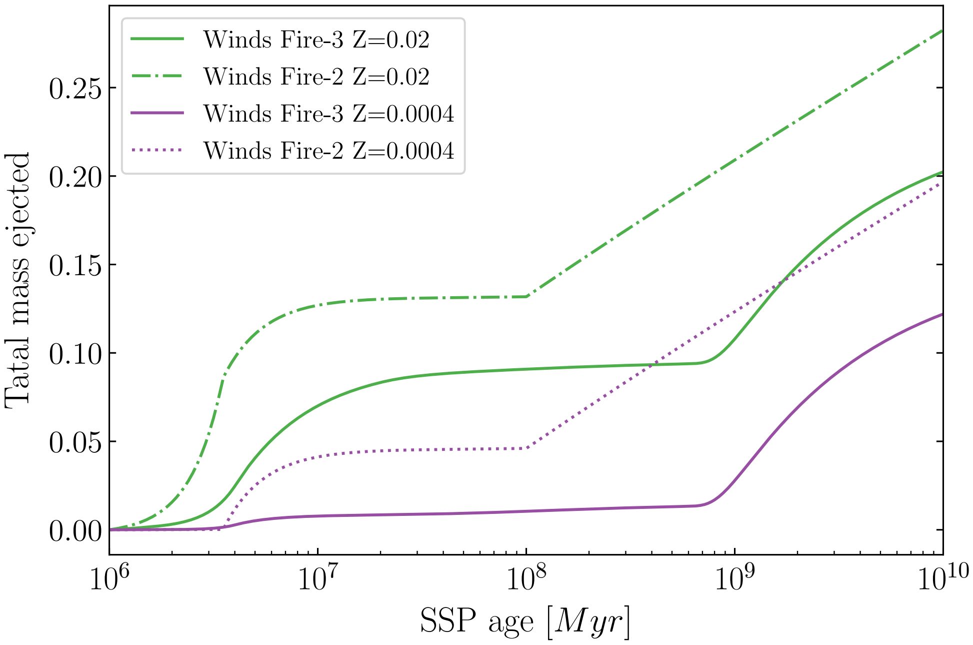


Figure 2.7: Total mass injection from continuous losses by stars. We plotted two metallicities values for both Fire-3 (solid lines) and Fire-2 (dotted lines) to show the dependence of the total ejected mass on the star chemical composition.

The bump in the rate due to the AGB losses creates a sudden increase in the total mass fraction of both models, the delay of the bump in Fire-3 makes the total mass ejected 7% less than Fire-2. In this case, the Fire-3 model predicts less mass released relative to Fire-2 at any metallicity. Furthermore, there is a strong dependency on metallicity in both models, with lower metallicity stars yielding $\sim 7\%$ less total mass fraction ejected.

2.2.3 Yields

Nuclear reactions inside stars create new chemical elements. The cooling function depends strongly on the presence of metals in the ISM. Thus, following and estimating the total quantity of metals released by stars plays a central role in accurately modeling the evolution of cooling in the simulation. Also, the chemical composition of stars alters their life cycle, changes the mass loss via winds, and modifies the cross-section of the matter that is influenced by radiation pressure. The amount of each specific element produced depends on the mass of the star.

In Fire-3 the production of elements is parameterized by introducing franctional yields. These take the form of functions of the metallicity of the SSP and time defined as:

$$y_{cc,j} = \frac{\text{ejection rate}_j}{\text{ejection rate}_{tot}} \tag{2.4}$$

Where j is an element in the list (He, C, N, O, Ne, Mg, Si, S, Ca, Fe) and tot stands for the total mass ejection rate. In order to get the rate of ejection of a specific element, we need to multiply its yield by the ejection rate given in eq. (2.1). Hopkins et al. (2022) provide the fit for the fractional yields of single elements for SNII as:

$$y_{cc,j} = \begin{cases} 0 & (t \leq t_{cc,j,1}) \\ a_{j,1}(t/t_{j,1})^{\psi_{cc,j,1}} & (t_{j,1}) < t \leq t_{j,2}) \\ \vdots & \vdots \\ a_{j,n}(t/t_{j,n})^{\psi_{cc,j,n}} & (t_{j,n} < t \leq t_{j,n+1}) \\ 0 & (t_{j,n+1} \leq t) \end{cases} \tag{2.5}$$

Where $\psi_{cc,j,i} = \ln(a_{cc,j,n+1}/a_{cc,j,n})/\ln(t_{cc,j,n+1}/t_{cc,j,n})$, j is the element, $a_{j,i}$ and t_j, i are found in sec. 3.11.2 of Hopkins et al. (2022).

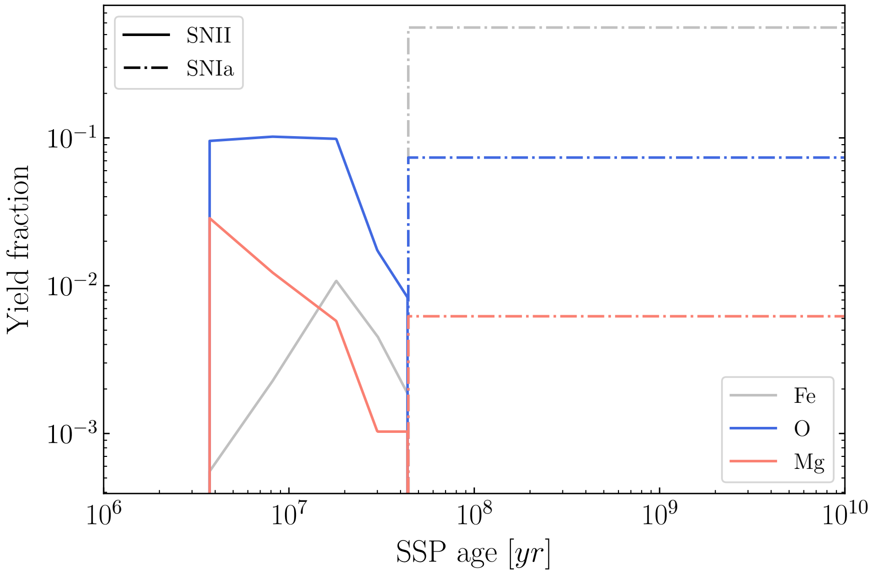


Figure 2.8: Fractional yields for SNe as a function of time obtained for Z=0.02. Continuous lines are the contribution of SNII and dotted-dashed lines are the contributions of SNIa.

Notice how half of the mass expelled by SNIa is in the form of Iron. While the total mass returned by SNIa is negligible with respect to the SNII, the metal enrichment coming from both is comparable. This makes SNIa critical for metal enrichment of the ISM.
The enrichment from winds is also considered.
To quantify the yield of an element from the continuous mass loss, the production of atomic species in Fire-3 is parameterized through the most important nuclear fusion channels. These are called nuclear channel yields. By combining all the nuclear channel yields the final fractional abundance production is obtained. The more important nuclear channel reactions given are:

$$y_{j=(HHe,CNO,HC)} = \begin{cases} a_{j,1}(t/t_{j,1})^{\psi_{j,0}} & (t \leq t_{j,1}) \\ a_{j,1}(t/t_{j,1})^{\psi_{j,1}} & (t_{j,1} < t \leq t_{j,2}) \\ \vdots & \vdots \\ a_{j,n}(t/t_{j,n})^{\psi_{j,n}} & (t_{j,n} < t \leq t_{j,n+1}) \end{cases} \tag{2.6}$$

Where $\psi_{c,i,j} = \ln(a_{j,n+1}/a_{j,n})/\ln(t_{j,n+1}/t_{j,n})$ and j is the channel. In this case both the time and $a_{j,n}$ are tabulated in subsection 3.11.3 of Hopkins et al. (2022). The fractional yield for each element is the combination of all nuclear channel yields. They are defined as $f_{He} = f_{He,0}(1 - y_{HeC}) + y_{HHe}f_{H,0}$, $f_N = f_{N,0} + y_{CN}f_{C,0} + y_{ON}f_{O,0}$, $f_C = f_{C,0}(1 - y_{CN}) + y_{HeC}f_{He,0} + y_{HC}f_{H,0}(1 - y_{HHe})$ and $f_O = f_{O,0}(1 - y_{ON})$, and the initial abundance (quantities denoted with the subscript zero) are given in Hopkins et al. (2022). We plot the fractional enrichment tabulated above in Fig. 2.9.

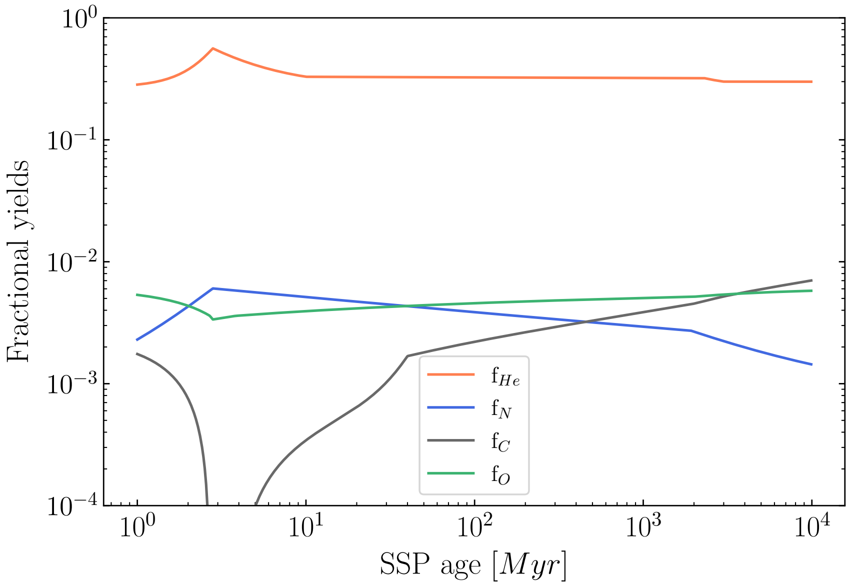


Figure 2.9: Fractional yields from continuous mass loss. All the stars are taken into account.

Notice how they parametrize only elements produced via nuclear fusion during the lifetime of the stars, they consider the abundance of other elements to be equal to the initial one.

2.3 SMUGGLE

In this section, we discuss the current implementation of stellar evolution in the SMUGGLE model (Marinacci et al., 2019). The model takes as input .hdf5 tables, one for the lifetime of the stars, one for the stars that explode as SNII, and another one for stars that end their life as planetary nebulae (from now on we will refer to these stars as AGB). The lifetime file contains the length of the main sequence for a set of stars with given masses and metallicities. Each one of the other two files contains for a subset of the stellar masses the mass ejected in the end-life phase (i.e. the post-main sequence phase) and the elemental yields (we describe the structure of these files in detail in Appendix A). The model interpolates (linearly in log space) data read from these files, obtaining continuous functions. Then using these interpolated functions, it evaluates the evolution of the stellar population as a function of time. In other words, the model basically self-consistently determines the rates involved for describing stellar evolutionary processes starting from the hdf5 tables. This has the major advantage that different stellar evolution models can be used by simply replacing the data tables without the need of implementing custom fitting functions.

It is worth mentioning two exceptions to this implementations. In SMUGGLE SNIa rates are not tabulated but instead parameterized with an analytic DTD function (see section 2.2.1 and Vogelsberger et al. 2013, for a detailed description). Similarly, winds from massive ($M > 8\mathrm{M}_\odot$) are implemented following a modified version of the Fire-2 prescription (Hopkins et al., 2018). Understanding, quantifying, and interpreting stellar evolution models in SMUGGLE requires a standalone code that emulates this procedure. What follows is a detailed description of this standalone code that was developed in this thesis work to reproduce the stellar evolution part of the SMUGGLE model[2]

2.3.1 Model Theory

It is a well corroborated fact that the lifetime of a star depends strongly on its initial mass and more weakly on its metallicity and rotation velocity (Groh et al., 2019; Amard & Matt, 2020). A stellar particle in a simulation performed with SMUGGLE represents a co-eval stellar population. Therefore, stars inside a stellar particle are assumed to have the same metallicity, so we treat the metal content as a fixed parameter. Fluctuations in stellar parameters (overshooting, rotation, stellar magnetic fields) and environmental effects (ISM pollution, encounters with other objects, magnetic fields, and cosmic rays) can make stars born with the same initial conditions evolve differently; however, the level of approximation at which we are working (we are treating a stellar population as a whole in the simulations) allows us to neglect these effects.

State-of-the-art stellar evolution codes can be used to calculate stellar lifetime given the star's initial mass and metallicity. Based on these results a lifetime function τ can be defined such that $\tau = \tau(M, Z)$ where M is the stellar mass and Z is the metallicity at birth of the stellar population. At fixed metallicity, τ is a monotonically decreasing function so it is invertible. Defining t_0 as the time when the SSP forms, the inverse function of τ is $\mathcal{M} = \mathcal{M}(t = t_0 + \tau, Z)$. $\mathcal{M}$ is the mass of the stars that end their life cycle at $t_0 + \tau$. Being more rigorous we say that the lifetime function represents the star with mass $\mathcal{M}$ that departs from the main sequence at a given time. In fact in SMUGGLE another key approximation is made, it is assumed that the post-main sequence evolution of a star happens instantaneously since it is on average 1/10 of the

[2] The code used is available at `https://github.com/lorevannini/Thesis-project/tree/main/SMUGGLE-Model`.

total life of the star. Given this approximation, we can calculate how many stars depart from the main sequence in the time interval $[t, t + \Delta t]$ by integrating the SSP over its initial mass function (IMF) between $[\mathcal{M}(t, Z), \mathcal{M}(t + \Delta t, Z)]$. For instance, we can calculate the number of SNII events as:

$$N_{\mathrm{SNII}}(t, t + \Delta t) = \int_{\min(M(t), M_{\mathrm{SNII,min}})}^{\max(M(t+\Delta t), M_{\mathrm{SNII,max}})} \phi(m) dm \tag{2.7}$$

where $M_{\mathrm{SNII,min}} = 8\,\mathrm{M}_{\odot}$ and $M_{\mathrm{SNII,max}} = 100\,\mathrm{M}_{\odot}$. We present an illustration of this approach in Fig. 2.10.

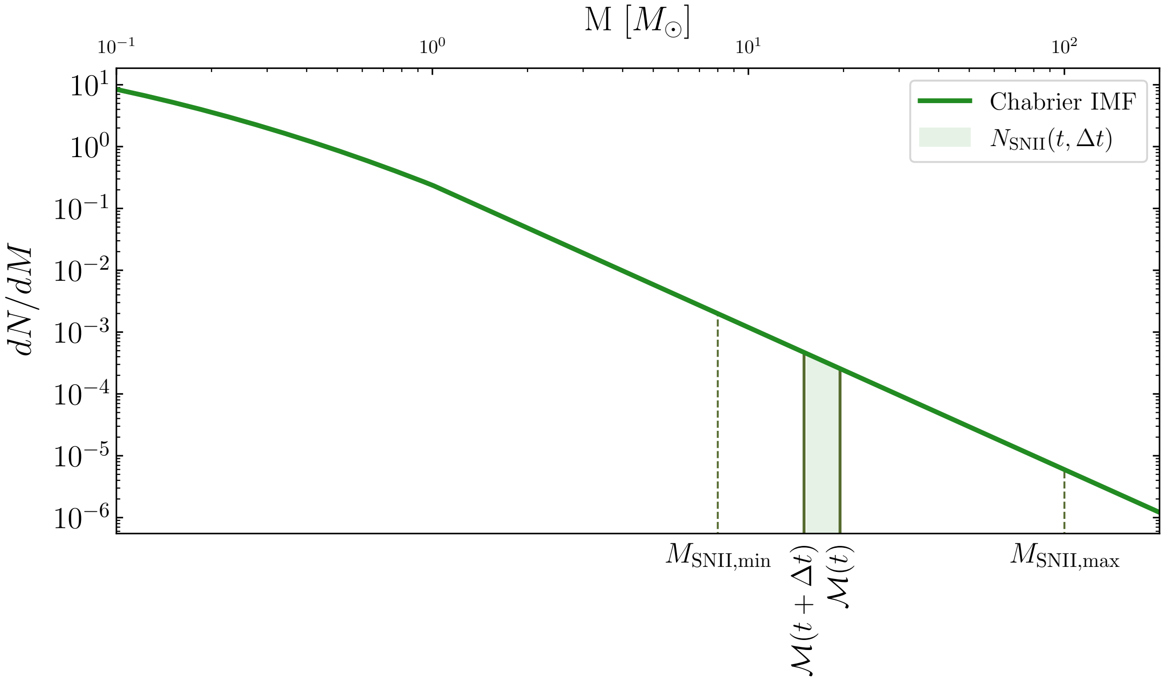


Figure 2.10: Chabrier IMF used in SMUGGLE. We present a visual representation of eq. 2.7. We took $t_0 = 0$ and omitted Z dependencies to simplify the notation.

SNII are discrete events. This means that the rate of SNII in eq. (2.7) is the expected number of explosions in a timestep Δt. To determine whether a SN event takes place, we sample form a Poisson distribution $p(n; \lambda) = \frac{\lambda^n e^{-\lambda}}{n!}$. In the last equation, p is the probability of having n supernova events given an expected value λ over the time step Δt. In our case $\lambda = N_{SNII} * M_*$, where M_* is the stellar population mass at birth and N_{SNII} derives from eq. (2.7).

We can use a similar approach to obtain the ejected mass as a function of time. We integrate the IMF multiplied with the star mass m and the so-called efficiency function $f_{rec}(m, Z)$. This function represents the percentage of the mass of a star that is recycled in the ISM and depends on the mass and metallicity of any given star in the SSP. We obtain $f_{rec}(m, Z)$ by dividing the end-life mass ejected and the correspondent stellar mass that we found in the tables, then we interpolate linearly in log-space between the points. The mass ejected by a SSP in the the time interval Δt can then be formalazed as.

$$\Delta M(t, \Delta t, Z) = \int_{M(t+\Delta t)}^{M(t)} m f_{rec}(m, Z)\phi(m)dm \tag{2.8}$$

Each file contains also the yields for the elements "H", "He", "C", "N", "O", "Ne","Mg", "Si", "Fe". In SMUGGLE yields are defined as the quantity of a specific element produced or consumed (depending on the sign) during stellar evolution. We define the yield for an element i as $y_i(M, Z) = M_{i,enrich}(M, Z) - MZ_i f_{rec}(M, Z)$, where $M_{i,enrich}(M, Z)$ is the total mass of an element i expelled by a star of mass M and metallicity Z and $MZ_i f_{rec}(M, Z)$ is the initial mass of the element inside a star of mass M and metallicity Z at the zero age main sequence. In this way, we have an expression that encapsulates the element mass produced or consumed during stellar evolution. We can then calculate the mass ejection rate for every traced element as follows:

$$\Delta M_i = \int_{M(t+\Delta t)}^{M(t)} (m f_{rec}(m, Z) Z_i + y_i(m, Z))\phi(m)dm. \tag{2.9}$$

While for high masses ($8\,\mathrm{M}_\odot \leq M$) it is assumed that ejection is due to SNII explosions, for the low mass end the release in the ISM of the gas is more gentle through continuous mass loss via AGB winds. In the next sections, we describe step by step the procedure used in SMUGGLE to obtain the SNII rate, the mass ejection rate, and the cumulative mass ejection in order to fully characterize the model.

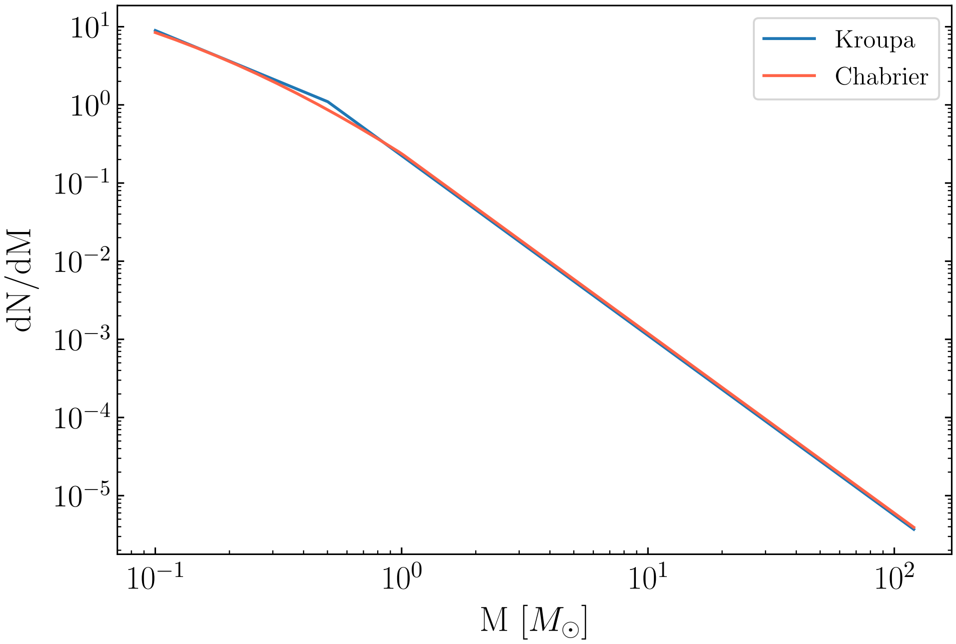


Figure 2.11: The Chabrier IMF used in SMUGGLE plotted against the Kroupa IMF used in FIRE 3. Notice how the Chabrier IMF has slightly large values at the high mass end relative to the Kroupa IMF. However, the two IMFs are largely comparable.

2.3.2 IMF

In this work we used both Chabrier IMF and Kroupa, used in Marinacci et al. (2019) and Hopkins et al. (2022), respectively. Notice how the two IMFs resemble each other. Therefore, we do not expect a great difference if we choose one respect to the other. Since the Chabrier at the high mass end is a bigger value than the Kroupa, and since we normalize the IMF from 0.1 $\mathrm{M}_\odot$ to 100 $\mathrm{M}_\odot$ we expect that Chabrier will return slightly more mass. For the Chabrier IMF, we use this parametrization:

$$\Phi_C(m) = \begin{cases} Am^{-1}\exp\left(-\frac{\log(m/m_c)^2}{2\sigma^2}\right), & \text{for } m < 1\,\mathrm{M}_\odot \\ Bm^{-2.3}, & \text{for } m \geq 1\,\mathrm{M}_\odot, \end{cases} \tag{2.10}$$

while for the Kroupa IMF we will use:

$$\Phi_K(m) = \begin{cases} Cm^{0.3}, & \text{for } m \leq 0.08\,\mathrm{M}_\odot \\ Dm^{1.3}, & \text{for } 0.08\,\mathrm{M}_\odot \leq m < 0.5\,\mathrm{M}_\odot \\ Em^{2.3}, & \text{for } 0.5\,\mathrm{M}_\odot \leq m, \end{cases} \tag{2.11}$$

where A,B,C,D and E are normalization factors depending on the mass range considered. From now on we will report all the results using Chabrier IMF normalized between 0.1 $\mathrm{M}_\odot$ and 100 $\mathrm{M}_\odot$ unless otherwise specified.

2.3.3 Data interpolation

Our code computes the stellar evolution of an SSP starting from data encoded in .hdf5 tables. The code works for generic metallicity Z, which is considered the same for all the stars in the SSP. The code derives the SSP evolution for an arbitrary metallicity Z following these steps. Firstly, it finds the interval $Z_{i+1} \leq Z < Z_i$, where Z_{i+1} and Z_i are metallicity present in the .hdf5 file. For a generic function $f_{spl}(M, Z)$ of both stellar mass M and metallicity Z the code calculates the linear interpolation for this metallicity as:

$$f_{spl}(M_i, Z) = \frac{f_{spl}(M_i, Z_{i+1}) - f_{spl}(M_i, Z_i)}{Z_{i+1} - Z_i} Z + f_{spl}(M_i, Z_i), \tag{2.12}$$

keeping the mass fixed at first. In this way for a generic Z we find $f_{spl}(M_i, Z)$ for every stellar mass present in the file. The code interpolate in log-space, this means that we calculate the log10 of all the quantities and then we calculate eq. (2.12) for the set of points we are interested in. This is because most of the quantity we are interested in interpolating resemble power laws. Power-laws appear as straight lines in log-space, making the linear interpolation more reliable. We want to find $f_{spl}(M, Z)$ for a generic stellar mass also. We can repeat the same procedure of eq. (2.12), but this time keeping Z fixed. We first found the interval $M_i \leq M < M_{i+1}$ and then we interpolate between masses:

$$f_{spl}(M, Z) = \frac{f_{spl}(M_{i+1}, Z) - f_{spl}(M_i, Z)}{M_{i+1} - M_i} M + f_{spl}(M_i, Z) \tag{2.13}$$

In our work $f_{spl}(M, Z)$ can be the lifetime function, the ejected mass in the end-life or the elemental yields. Every time we will need to interpolate one of this quantity we will refer to the procedure otulined in this subsection.

2.3.4 Lifetime function

In SMUGGLE, once a star particle forms, an SSP population is initialized at t_0 simulation time. The code finds an approximate expression for $\mathcal{M} = \mathcal{M}(\tau = t - t_0, Z)$ where τ is the SSP age. In Fig. 2.12 we see the data in the .hdf5 representing the lifetime of the stars plotted in log scale. Each point indicates the lifetime of a star given its mass, the points were read from the file. In the figure, we plotted two sets of data for 2 different metallicities. We assume that this set of points samples the function

$\mathcal{M} = \mathcal{M}(\tau = t - t_0, Z)$. Following the steps we introduced in 2.3.3, we first solve eq. (2.12) for the selected metallicity. Then we linearly interpolate between mass points in log-space evaluating eq. (2.13). In this way, we get the approximate lifetime function (in Fig. 2.12 the curve in green). This is a fundamental quantity used for any subsequent determination of quantities related to stellar evolution in the model such as, for example, the SNII rate in eq. (2.7).

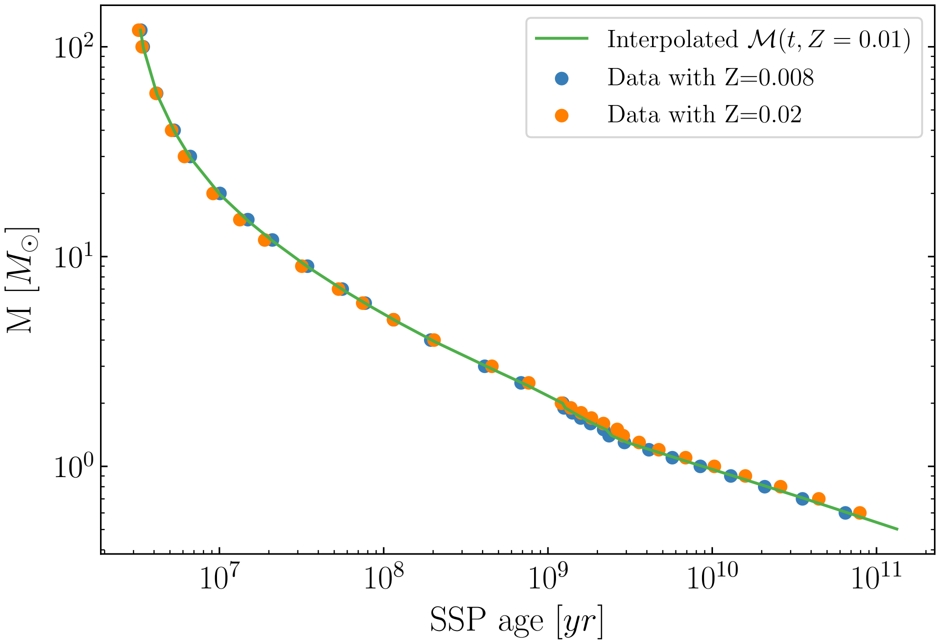
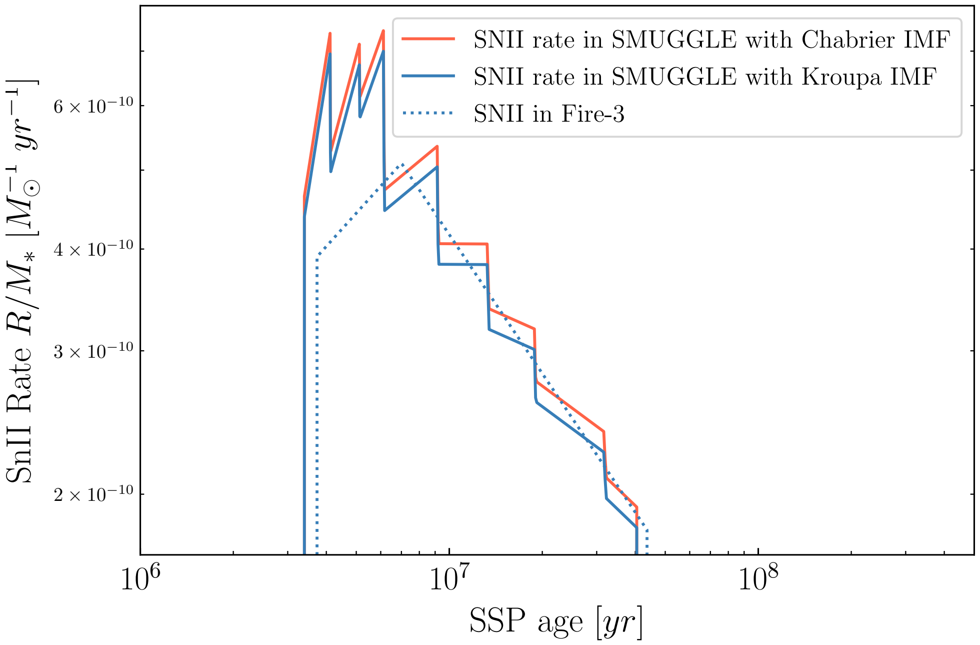


Figure 2.12: Interpolated lifetime function (green line). Notice how the green curve falls in the middle of the orange and blue dots, as expected from the interpolation in metallicity.

In Fig. (2.12) we notice how around $M \sim 2$ $\mathrm{M}_\odot$ we have higher density of sampled points in the file. This is because the lifetime function for those masses can only be accurately reproduced with a finer sampling. We also notice how for stars in the range $8\,\mathrm{M}_\odot \leq M < 100\,\mathrm{M}_\odot$ we only have 9 sampled points, making the sample in this interval more coarsely sampled with respect to the other part of the plot.

2.3.5 SN rates

In SMUGGLE, once a star particle forms, an SSP population is initialized at t_0 simulation time. The code finds an approximate expression for $\mathcal{M} = \mathcal{M}(\tau = t - t_0, Z)$ where τ is the SSP age. In Fig. 2.12 we see the data in the .hdf5 representing the Getting the rate of SNII requires calculating $N_{\mathrm{SNII}}(t, t + \Delta t)$ and dividing it with Δt. In our script, we initialize an array of 1000 masses $m[i]$ from $8\,\mathrm{M}_\odot$ to $100\,\mathrm{M}_\odot$ with equal logarithm spacing and for each of them we calculate the corresponding lifetime using the inverse of the interpolated function τ. Initializing times and computing the corresponding mass gives the same result. We then perform the numerical integral (2.7) using trapezoids to get the rate of the SNII. We set the trapezoid integration by calculating the sum of the IMF evaluated in masses with index i and $i+1$. To get the final rate we multiply by $\Delta M = m[i+1] - m[i]$ and divide this by the difference between the stellar masses' corresponding lifetimes $\Delta t = \tau(m[i+1], Z) - \tau(m[i], Z)$. With this procedure, we obtain the rate per unit stellar mass plotted in Fig. 2.13.



1

Figure 2.13: Type II supernova rates as a function of time extracted from SMUGGLE lookup tables. We also plotted the rate for a different IMF and the one from the Fire-3 model(Hopkins et al., 2022).

The discontinuities in the first derivative are an artifact of the interpolated lifetime function. This is confirmed by the fact that the number of discontinuities is equal to the number of mass points between $8\,\mathrm{M}_\odot$ and $100\,\mathrm{M}_\odot$ contained in the file. We expect this oscillation of the rate to not significantly influence the outcome of the simulations using SMUGGLE, since, as we discussed in section 2.3.1, the number of SNII is sampled from a Poisson distribution and therefore some degree of stochasticity is expected.

2.3.6 SNII mass ejection

In the hdf5 table, the amount of expelled material in supernova explosions is found for a set of masses and sampled metallicities. To be consistent with the SMUGGLE model, we need to derive from the data contained in the file the efficiency function, namely, the mass ejected normalized to the initial stellar mass. This efficiency function is interpolated by following the procedure in subsection 2.3.3. In this work, we used two different tables for high-mass stars ($M > 8\,\mathrm{M}_{\odot}$). The first table ignores the presence of continuous mass loss throughout the lifetime of the stars and returns all the mass when the star explodes. The second table uses the parameterization for OB stars winds provided in Fire-2 (Hopkins et al., 2018). In the latter case the mass ejection from SNII is re-normalized in a way that makes the cumulative mass loss equal to the case with no winds. Notice that implementing winds as an explicit function of the age of the stellar population makes the winds depend on the chosen IMF. If we were to change the IMF, we would need to change also the normalization of the ejected mass in the model with the winds. We report both efficiencies in Fig. 2.14.

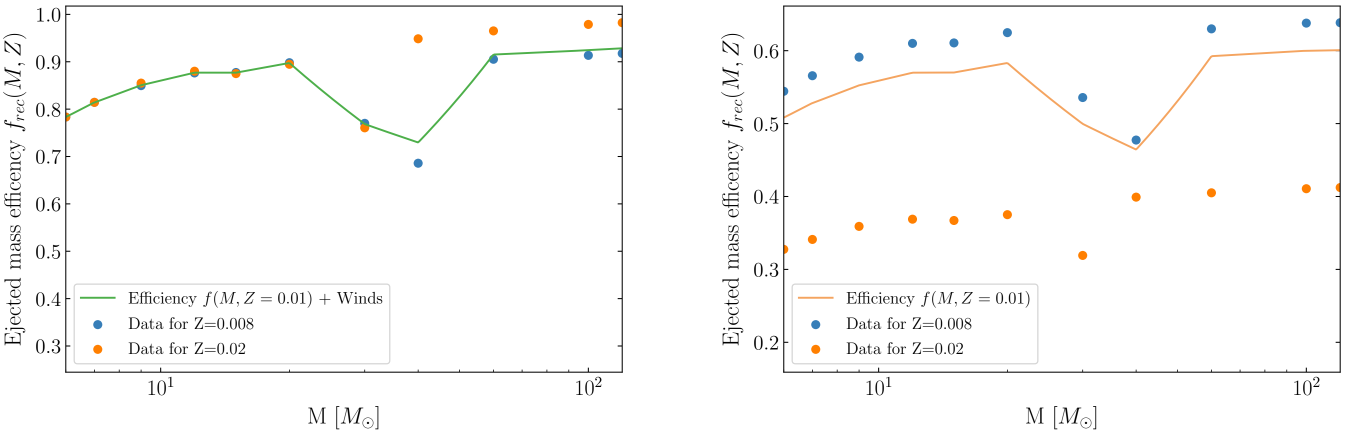


Figure 2.14: Ejection efficiency for SNII. The y-axis is not log-scaled. In the right panel, we show the model that considers the continuous mass loss from winds, on the right panel we do not have any continuous mass loss, and all the mass is returned by SNII explosions.

The efficiencies in the model that incorporates the winds have much less dependency on the metallicity of the SSP, while this dependency is much more enhanced in the model that implements winds. This is because winds depend on the metal content of the star and the mass loss from them becomes more prominent with increasing metallicity Vink et al. (2001). To calculate the specific mass ejection rate (i.e., the mass ejected by the entire SSP per unit mass and time) by SNII we solve numerically with trapezoids the integral (2.7) and divide by Δt. We integrate with trapezoids the IMF multiplied by the interpolated ejection efficiency and the mass of the star knowing the lifetime function. The result is shown in figure 2.15.

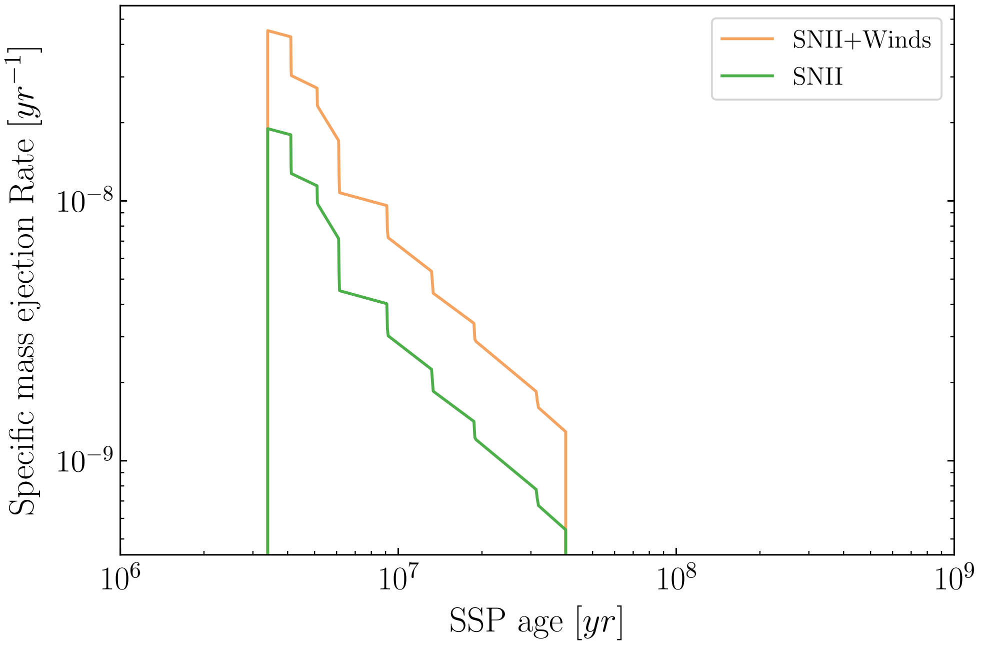


Figure 2.15: Specific mass ejection rate for both models calculated at Z=0.02. In gold we see the original model from Portinari et al. (1998), while in green we see the model currently used in SMUGGLE, in which the ejection from the SNII is renormalized considering the winds from Hopkins et al. (2018)

Notice how the two ejection rates have the same shape, but the model that considers all mass loss as occurring at the end-life phase consistently has a higher rate. This is expected based on how the model with winds has been defined.

2.3.7 Low mass stars

Stars with masses $M \leq 8\,\mathrm{M}_{\odot}$ release gas in the ISM without undergoing a core-collapse SN explosion, so they are attached to different feedback channels (equal to stellar winds). For this reason, we have a different table (with the same format as the one for SNII) from which we get the data. For simplicity, we will call the contribution from low-mass stars the AGB channel.
Calculating the mass rate from the AGB involves the same procedure as for the high-mass stars. Therefore, we just report the result in Fig. 2.16.

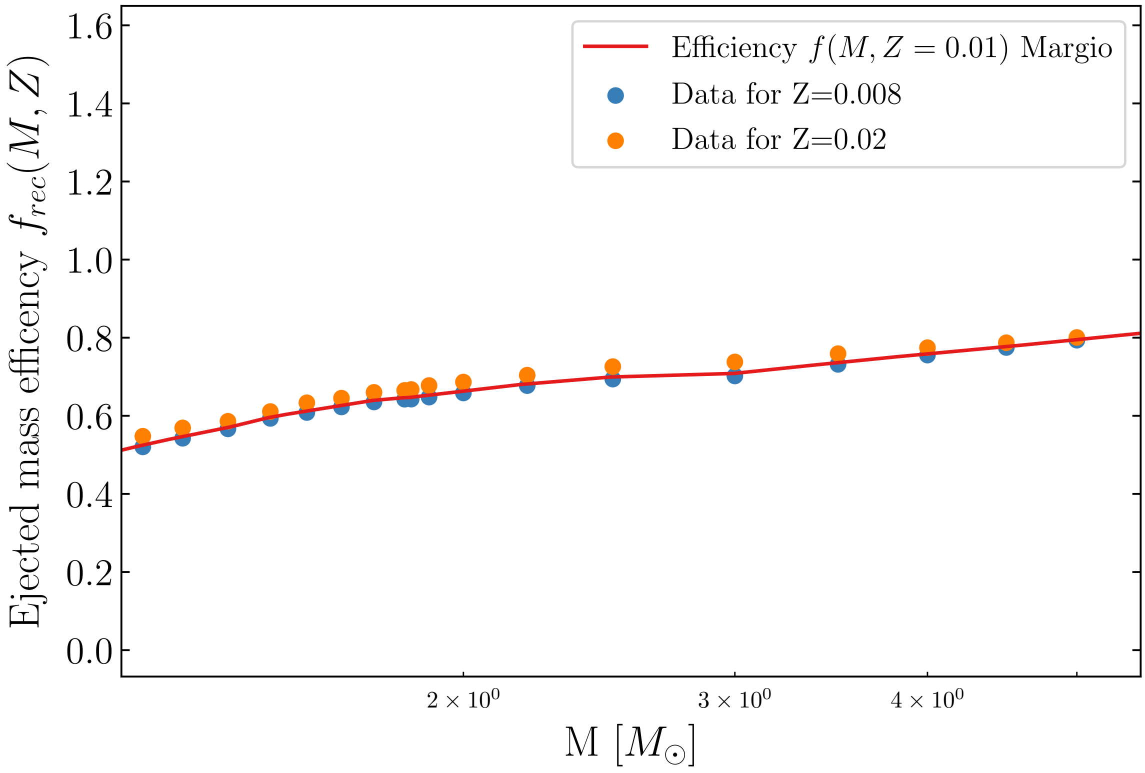


Figure 2.16: Ejection efficiency of stars with $M \leq 8\,\mathrm{M}_\odot$. Notice how the efficiency has a smoother trend with respect to the SNII case reported in Fig. (2.14)

The hdf5 file presents only masses up to 5 $\mathrm{M}_\odot$, so we kept the efficiency constant at $f(5\,\mathrm{M}_\odot, Z)$ in the mass interval $5\,\mathrm{M}_\odot \leq M < 8\,\mathrm{M}_\odot$. In the case of the AGB stars, all the mass is considered to be lost in the last phase of stellar evolution. The efficiency of the emission has a growing trend towards higher masses. This is because the mass of the remnant has no linear dependence on the initial stellar mass.

2.3.8 Total mass ejection

Adding all the contribution together we can plot the total stellar mass ejection rate from the SSP in Fig. 2.17.

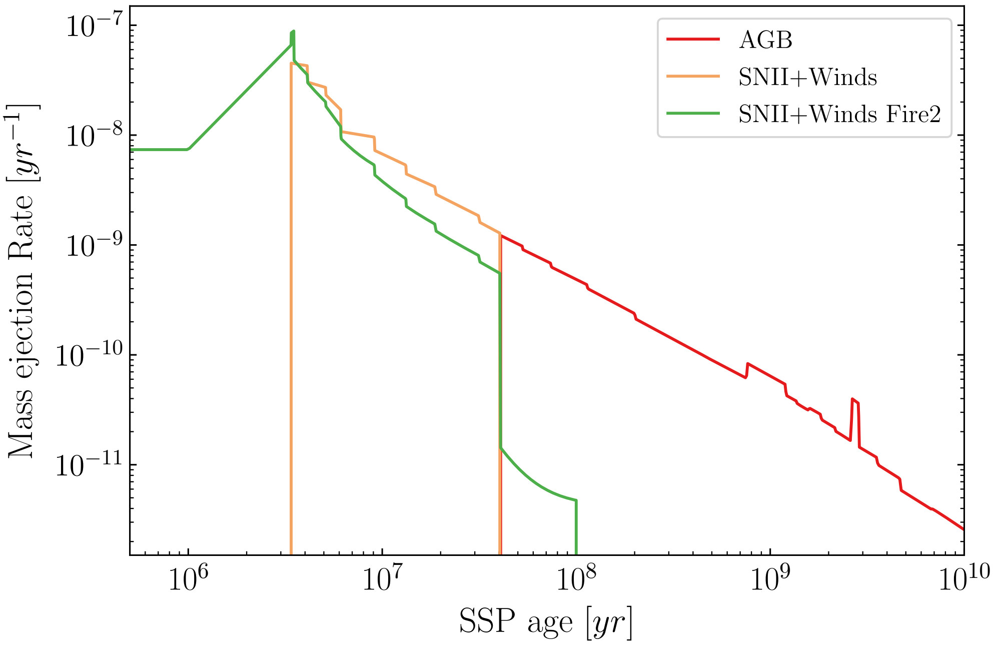


Figure 2.17: Specific mass ejection rate considering all stellar evolution channels for Z=0.02. We plot in green the model with the stellar winds taken from Hopkins et al. (2018) and in yellow the one that incorporates all the mass losses in the SN explosion. The red line represent the mass ejection rate due to AGB stars.

In Fig. 2.17 we observe how by taking into account O/B winds the SSP starts immediately to eject mass. This is a more realistic scenario compared to the case in which all the mass is released in SNII explosions, from now on we will refer to the model with winds implemented as the "current SMUGGLE model". We highlight how the ejection rate function resembles a power law. Adding the winds from Fire-2 in the model makes the slope of the ejection rate pretty different with a higher peak at ~ 2 Myr. We also have a prominent bump in the mass ejection rate at $t \sim 10^9$. This value is important since it corresponds to the place where we have more stellar masses in the table for sampling the lifetime function; we will show in the next section 3.4.1 that the bump is not an artifact of the interpolation, but a feature of the stellar evolution model adopted.

We can then calculate the total mass ejected per unit stellar mass integrating with trapezoids the specific mass ejection rate. We report both models, separating the different contributions in Fig. 2.18.

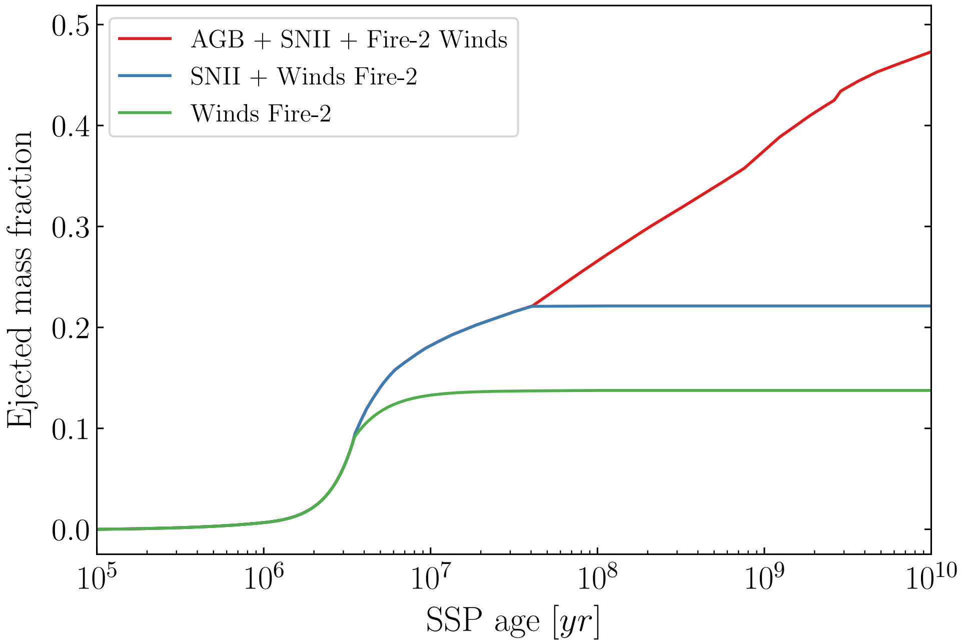


Figure 2.18: Cumulative ejected mass fraction from an SSP in the model currently implemented in SMUGGLE, with the winds taken from Hopkins et al. (2018).

We will discuss in detail the features of the current SMUGGLE model in Chapter 4 when we will compare it with all the other presented in this work.

2.3.9 Elemental Yields

As discussed in subsection 2.3.1, yields in the SMUGGLE model encode the variation of the abundance of a specific element in stellar evolution. In the hdf5 tables, we have for every mass at each metallicity the value of the yield of elements explicitly tracked by SMUGGLE and the total metal yield (the yield of all the elements *including* those not explicitly followed by the model that is not Hydrogen nor Helium). The procedure to interpolate the mass rate for a single element is the same as done for the mass rate (see subsection 2.3.3). We find the yield for the initial metallicity interpolating in log space for all the masses sampled in the table. We then interpolate the yield for a specific mass in log space. The result of this procedure are shown in Fig. 2.19 for stars with mass $8\,\mathrm{M}_\odot \leq M \leq 100\,\mathrm{M}_\odot$ and in Fig. 2.20 for $0.8\,\mathrm{M}_\odot \leq M \leq 8\,\mathrm{M}_\odot$.

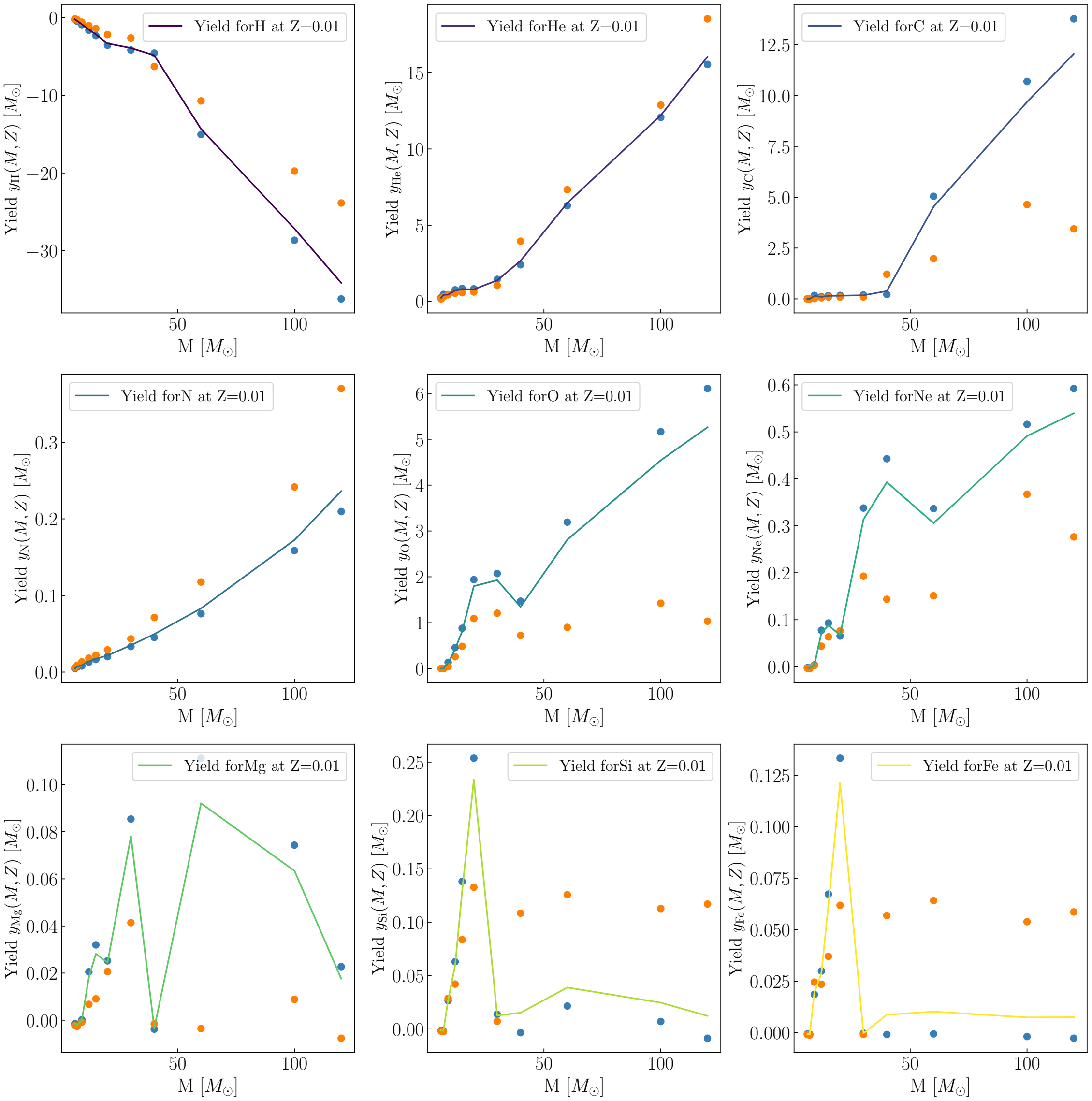


Figure 2.19: Yields for each chemical element that are explicitly tracked by SMUGGLE stellar evolution tables for massive stars that explode as SNII. All the yields are calculated for $Z = 0.01$.

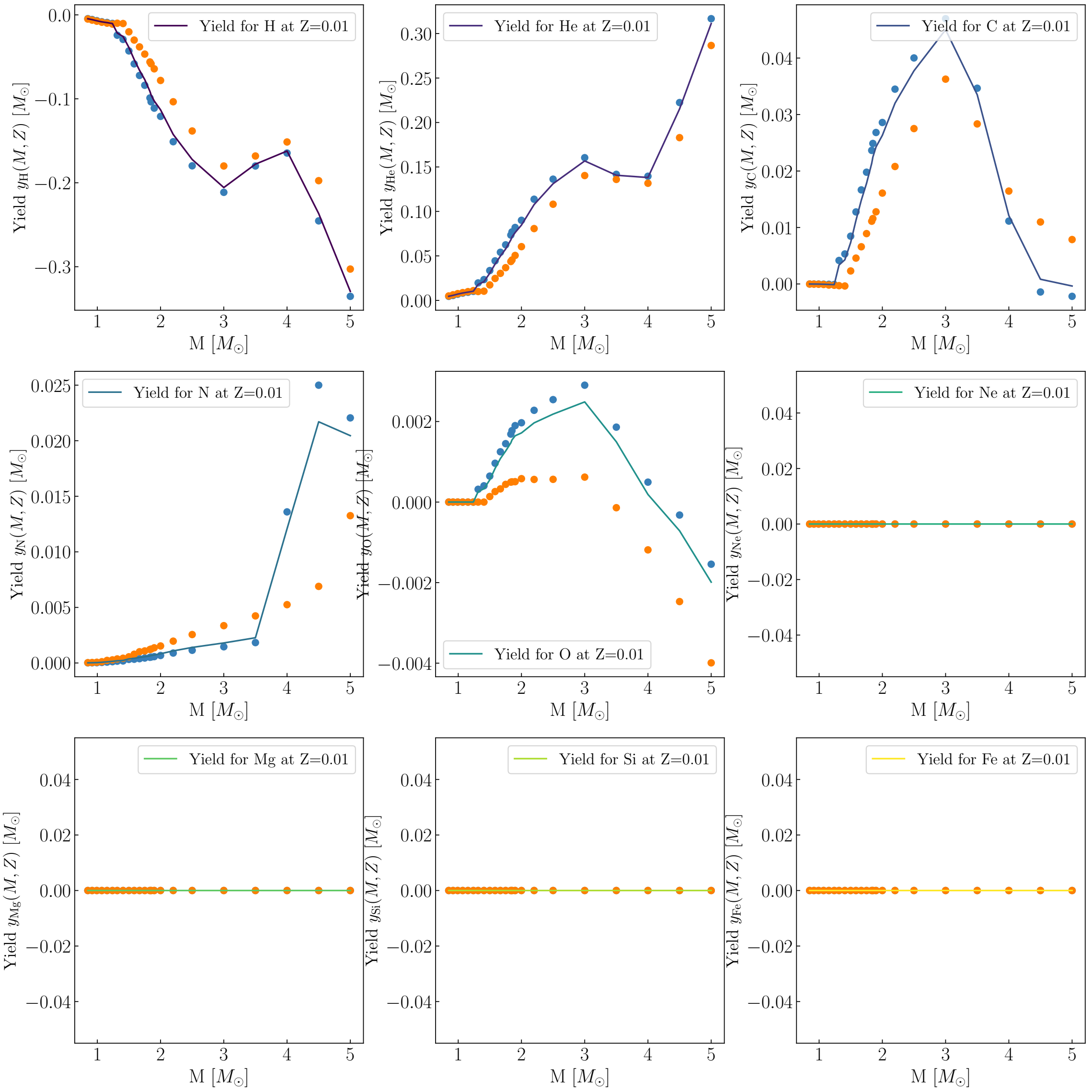


Figure 2.20: Yields for each chemical element that explicitly tracked by SMUGGLE stellar evolution tables for AGB stars. All the yields are calculated for Z=0.01. Heavier elements are not produced by low mass stars that do not explode as SNII, consistently the yield for those elements is zero.

To produce the plots, we solve eq. (2.9) numerically with the trapezoids and divide by Δt. In this way, we obtain the mass ejection rate for every species. Integrating the mass ejection rate for every single element in time results in the cumulative function for each element. We report the cumulative mass fraction for each element in Fig. 2.21

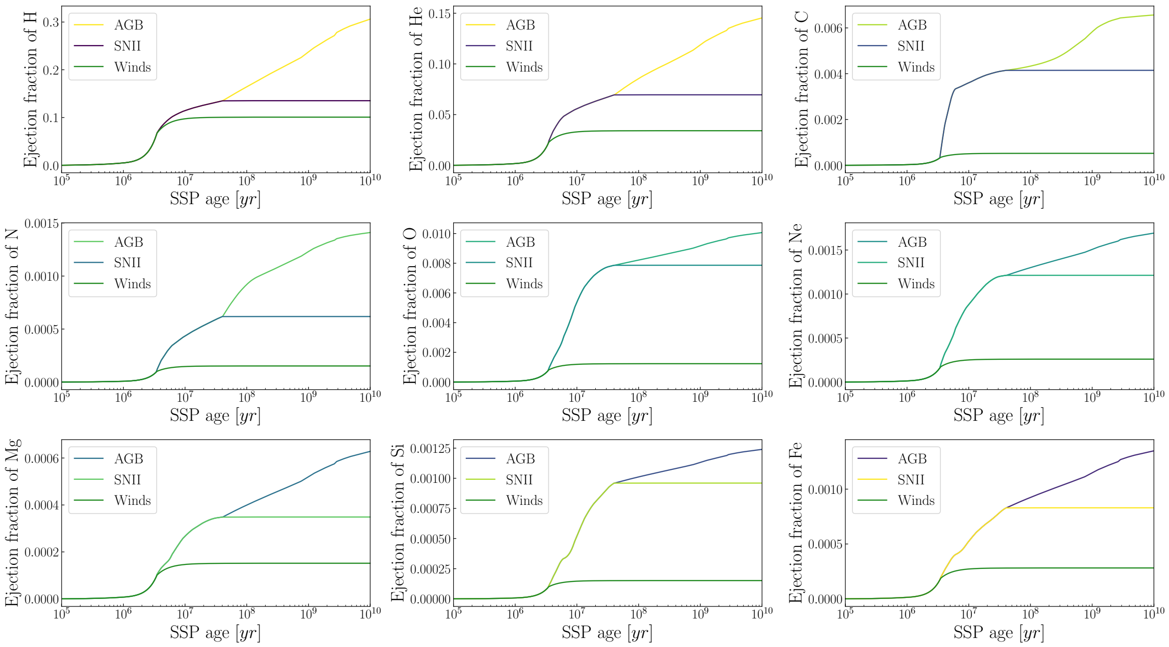


Figure 2.21: Cumulative mass function for all the elements followed by the SMUGGLE code.

As we notice, the highest mass fractions of elements ejected are those of hydrogen and helium. Winds become less and less relevant for heavier elements, this is because in SMUGGLE winds eject mass with the same initial chemical composition of the star (i.e. it is assumed that no production of new chemical species is involved). Notice also that for elements that have zero yields for the AGB, the cumulative function after SNII stops to explode is simply the total cumulative function re-normalized to the abundance of that element. This implies that the shape of the total cumulative AGB ejection is the same that we observe in Fig. 2.18.

2.3.10 Total Metal Yield

Metals are fundamental. The cooling of the gas is regulated by the function $\Lambda(T, Z)$. Λ is strongly effected by the metal enrichment of the gas (see sec. 1.7 for details), the more the metals the more the cooling. This directly effect star formation in galaxies. The hdf5 tables contain the yield for the total metal content, so that the Z of the gas can be estimated more accurately. By following the same procedure as for individual elements we can interpolate and get the expected metal yield as a function of stellar mass. An example of this procedure for a stellar metallicity $Z = 0.01$ is shown in Fig. 2.22.

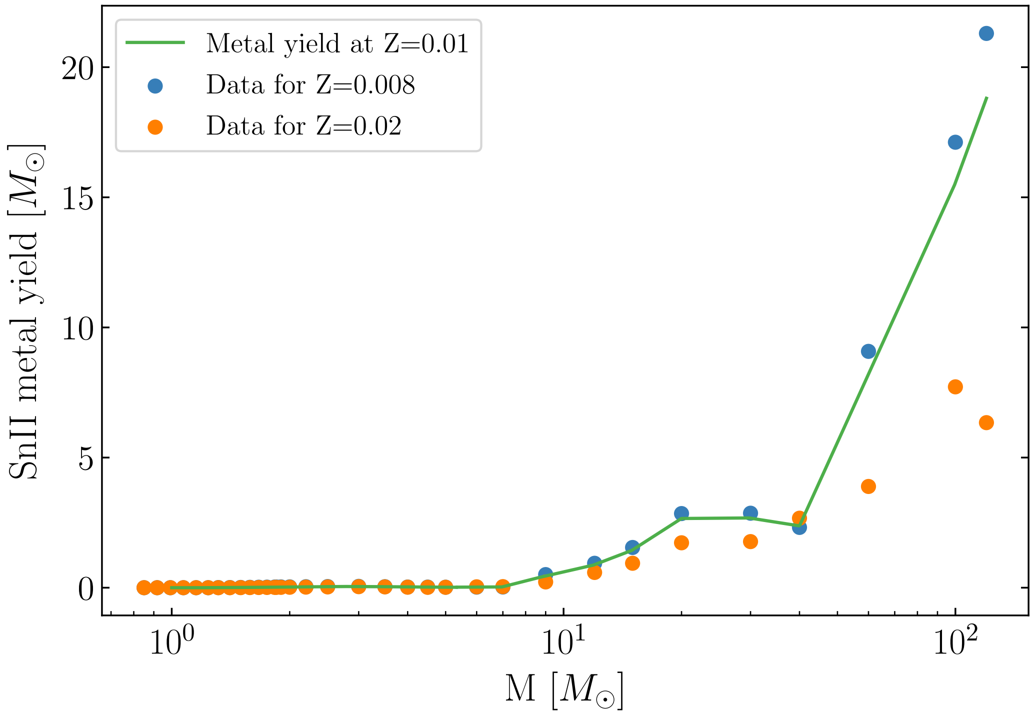


Figure 2.22: Yield for the total metal production of both AGB stars and massive stars exploding as SNII as a function of mass for $Z = 0.01$.

In Fig. 2.22 it is clear that most of the metal production is concentrated in stars with masses higher then $8\,\mathrm{M}_\odot$ (i.e. SNII explosions). Also a notable dependency on metallicity is present especially for $M > 40\,\mathrm{M}_\odot$. We then calculate and plot the metal ejection rate per unit stellar mass by numerically evaluating eq. (2.9). We report the result in Fig. 2.23.

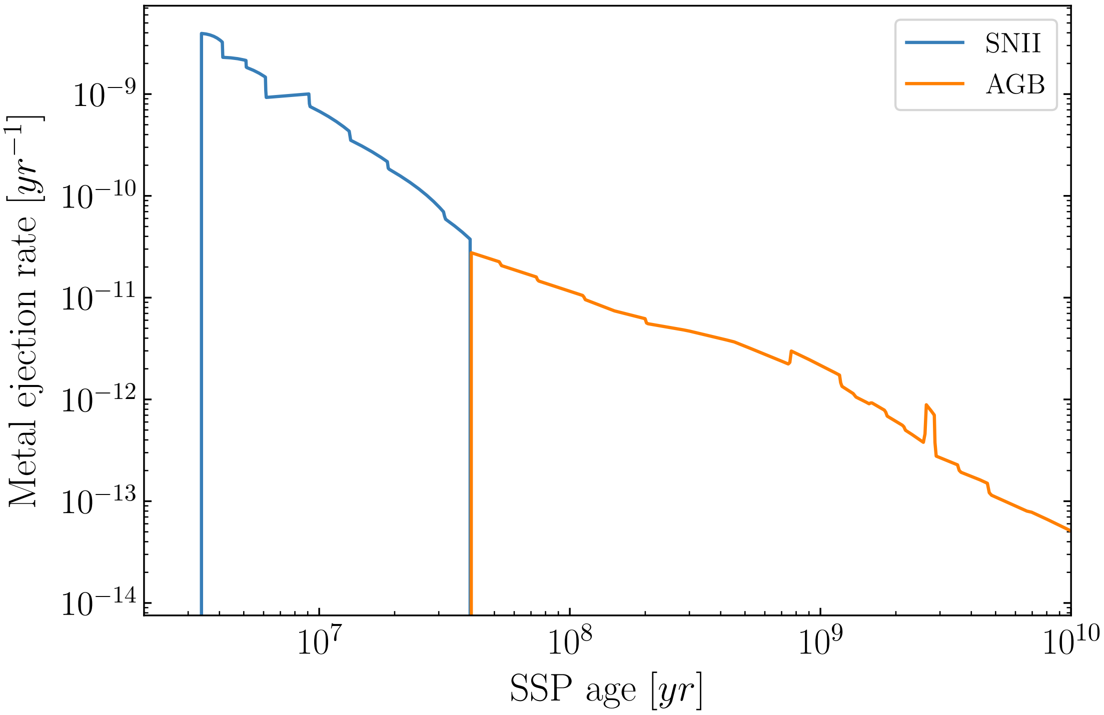


Figure 2.23: Specific metal ejection rate from total yield calculated for Z=0.02. For SNII we used the model not considerin stellar winds.

We notice how some of the features present in the mass ejection rate (Fig. 2.15 and Fig. 2.17) are also present in this plot. The rate is an order of magnitude lower than the one of the total mass. Is also worth noting that when passing from SNII to AGB the rate drops, this is due to the fact that metal production is lower in low-mass stars, as we can see in Fig. 2.22. We can better appreciate the contribution to the total metal ejection by plotting the cumulative functions of the metal ejection (Fig. 2.24).

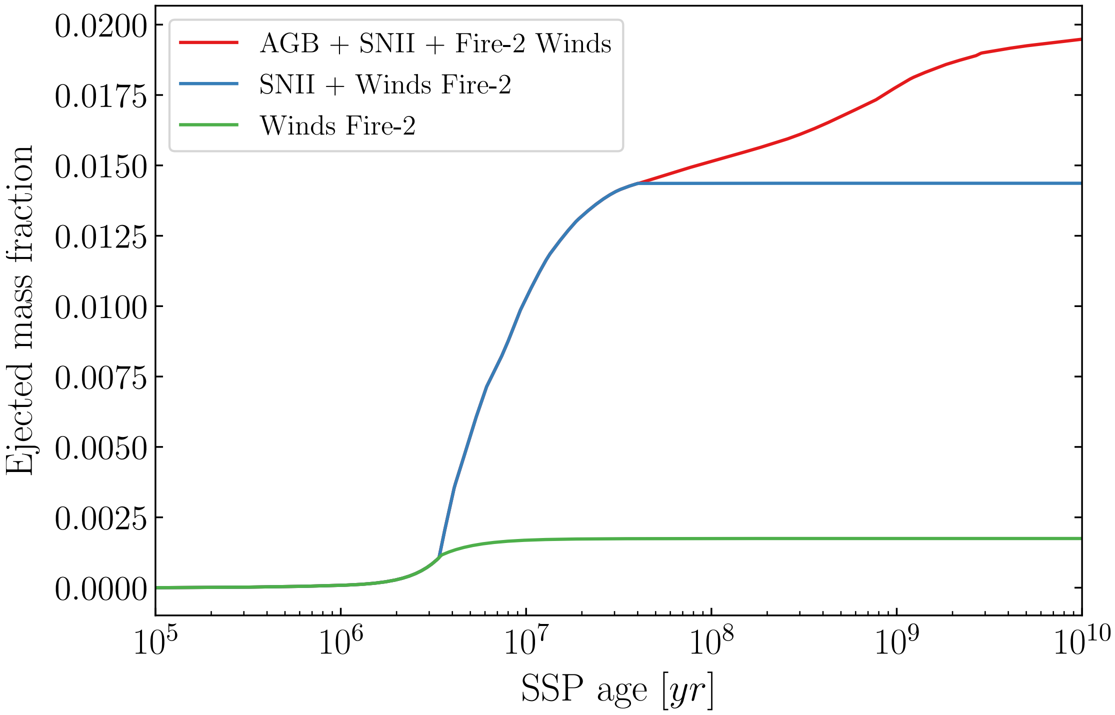


Figure 2.24: Cumulative metal ejection fraction from total yield calculated for Z=0.02. We plotted both the contributions from SNII and from AGB.

As we can see the total mass in metals at $t = 10^{10}$ yr is 2.3% of the initial mass of the SSP. The total mass ejected from SNII and AGB without including the winds is the 35% of total SSP mass. If we consider that winds expel the 7.5% of the total mass with $Z = 0.02$ (i.e. the initial metallicity) we have that the expelled gas has an average metallicity of $Z = 0.05$. Carrying the same analysis at lower metallicity, we found that the total mass of the metals released by the stars is 3% of the total SSP. This is a higher value relative to the high metallicity case. If we consider the winds, the gas expelled has on average $Z = 0.07$.

3

Model Construction

3.1 Introduction

One of the main goals of this thesis work is to build a new model for stellar evolution to be used in the ISM and stellar feedback model SMUGGLE. Since SMUGGLE reads the stellar evolutionary data from a set of hdf5 tables, we aim to produce a new set of such tables in a format directly readable by the model (see Appendix A for further details on the structure of these files). Making the new stellar evolution tables requires us to find a set of masses and metallicities that sample the lifetime, mass ejection, and elemental yields of a Simple Stellar Population (SSP). One way to approach the problem is to use software that simulates the evolution of an entire stellar population and translates this output into a set of hdf5 tables readable by SMUGGLE. Throughout the years many different SSP codes were developed, one of the most successful and widely used is STARBURST99 (Leitherer et al., 1999). The main advantage of STARBURST99 is the possibility to perform simulations of an entire SSP changing every time the initial conditions, such as the IMF and stellar spins and metallicities. New stellar evolution tracks can be added to the software using more and more refined stellar evolution codes. Tracks can be selected before launching the run. This flexibility allows for the continuous update of the output results, making STARBURST99 the standard choice for many researchers in the field, from pure simulation to more observational purposes (see, for example, Hopkins et al. 2022). In this work, however, we decided to opt for another approach. The natural way to produce a new stellar evolution model for SMUGGLE is to simulate the evolution of stars one by one. In recent years many different codes that simulate the stellar evolution of individual (or binary) stars were developed; software like MESA (Paxton et al., 2013, 2019), and Parsec (Bressan et al., 2012; Nguyen et al., 2022) are some of the many codes that simulate stellar evolution. The issue with these codes comes from the computational time needed to simulate very accurately the life cycle of a star. Especially at low masses (from 0.1 $M_\odot$ to 2 $M_\odot$), their performance can become time-consuming, since the lifetime of these stars is 3/4 orders of magnitude higher than the lifetime of the massive stars. The data we are interested in are the lifetime, and the mass loss, both from continuous (stellar winds and AGB stars) and explosive (SNII[1]) channels and the chemical element production. As discussed in the previous chapter, we approximate all the post-main-sequence evolution as a single timestep after the star leaves the main sequence. We can then approximate both SNII and (post)AGB evolution as instantaneous events that happen in a single timestep. Mass loss by winds is still a very active and debated topic in the stellar evolution community. Without going in into the very fine details of the properties of the winds, our objective is then to have winds that expel a quantity of mass that is consistent with the ejected mass in the end-life phase and the mass of the remnant. All these considerations on the type of analysis we want to carry out push us towards a fast computing code for the evolution of individual stars. We would like this code to run effectively many simulations that could explore a large array of different initial conditions. It is for these reasons that we decided to use the stellar evolution code SEVN to build our new stellar evolution model.

3.2 The SEVN code

SEVN (Iorio et al., 2023) is a publicly available code for stellar evolution. While detailed stellar evolution codes like MESA and PARSEC solve numerically the Euler equations for each timestep, stellar population synthesis models use analytical and semi-analytical recipes to solve both single and double stellar evolution. The distinctive feature of SEVN lies in the possibility of evolving both binary systems and using pre-computed evolution tracks for given initial conditions. The code interpolates among these tracks on the fly to get the stellar evolution for any set of initial parameters. SEVN installation comes with three different stellar evolution tracks, two from PARSEC (Nguyen et al., 2022; Iorio et al., 2023), and one from MIST tracks (Choi et al., 2016), computed with MESA. The

[1] The contribution of SNIa will still be treated with a DTD approach, see Section 2.2.1

PARSEC tables are computed with overshooting parameters $\lambda_{ov} = 0.4$ and $\lambda_{ov} = 0.5$, while MIST tracks adopt a different parametrization in which convective bubbles are destructed by diffusion effects, parametrizing the efficiency of the mixing instead (see section 3.6.2 of Choi et al. 2016 for more in depth details). The overshooting parameter is a factor that quantifies the distance a convection cell travels passing from an unstable convective region to a stable one. This is one of the most difficult parameters to model in stellar evolution, so we need to assume a fiducial set of tracks when we will implement the new model in SMUGGLE.

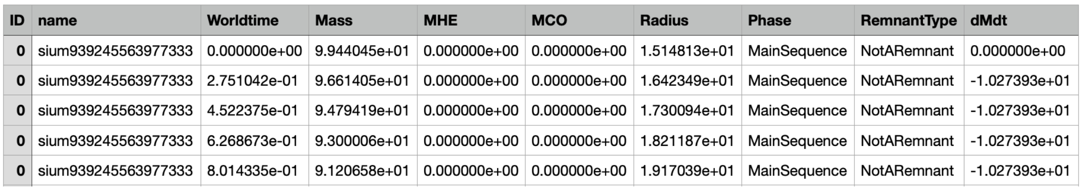
Is worth noting that SEVN allows for the inclusion of custom stellar evolution tracks; this will allow us to implement more up-to-date tracks once they will be used in the next generation of simulations.

The first aspect of SEVN we need to analyze is the effect of its key input parameters on the evolution of the simulated star. This analysis requires us to run the program many times, changing for each simulation the stellar mass, metallicity, spin, SN formalism (i.e. core-collapse explosion model), and stellar track available. This is feasible due to the speed and efficiency of the SEVN code. Time steps are adaptive, depending on the phase of the star's evolution. The operation performed for every timestep is an interpolation between pre-computed tracks. Therefore, the computation is very fast ($\sim 5\,s$ on the Apple M1 processor for simulating 1 star). To run SEVN we use a bash script in which we can modify the running parameters of the simulation. In the run script, we pass a text file containing the initial conditions of the simulation, that are reported in Table 3.1. For flexibility, we have written a bash script to launch the code looping around a given array of parameters; namely masses, metallicity, spin, and Sn formalism. We then store the output in a specific directory for each parameter combination.

M	Z	Ω_s	sn	t_{start}	t_{end}	dt_{out}	seed (optional)

Table 3.1: List of the initial conditions of a typical SEVN simulation. In the table, M is the initial mass, Z is the metallicity, Ω_s is the spin parameter (defined in subsection 3.3.1), sn is the formalism used for the eventual SNII explosion, t_{start} is the initial stellar time at which the simulation start (the zero-age main sequence, ZAMS, in our case), t_{end} is the final evolution phases we want our calculation to be performed (the formation of a remnant), dt_{out} is the time interval between the data we want to store in the output file, and seed is a random number used to set stochastic initial conditions.

The data of the simulated star are stored in a .csv file. Data that are printed in the same row of the output file all correspond to a specific simulation time. In the first column, we find the time starting from the zero-age main sequence (ZAMS) up to the end-life event; for each simulation time, we have the stellar mass, the mass of the helium nucleus, and the mass of the Carbon/Oxygen nucleus, the radius of the star, the actual phase of the stellar evolution, the type of remnant and the ratio between the mass difference and the timestep (i.e. the mass loss rate), are provided. In Fig. 3.1 we present an example of such a .csv output file.



ID	name	Worldtime	Mass	MHE	MCO	Radius	Phase	RemnantType	dMdt
0	sium939245563977333	0.000000e+00	9.944045e+01	0.000000e+00	0.000000e+00	1.514813e+01	MainSequence	NotARemnant	0.000000e+00
0	sium939245563977333	2.751042e-01	9.661405e+01	0.000000e+00	0.000000e+00	1.642349e+01	MainSequence	NotARemnant	-1.027393e+01
0	sium939245563977333	4.522375e-01	9.479419e+01	0.000000e+00	0.000000e+00	1.730094e+01	MainSequence	NotARemnant	-1.027393e+01
0	sium939245563977333	6.268673e-01	9.300006e+01	0.000000e+00	0.000000e+00	1.821187e+01	MainSequence	NotARemnant	-1.027393e+01
0	sium939245563977333	8.014335e-01	9.120658e+01	0.000000e+00	0.000000e+00	1.917039e+01	MainSequence	NotARemnant	-1.027393e+01

Figure 3.1: First rows of a SEVN .csv output file. Each row reports quantities corresponding to the same stellar time detailed in the first column. The Phase column can be used to distinguish the different evolutionary phases of the simulated star.

Using the phase column of the output file, we can separate the main sequence from the post main sequence evolution. This is particularly helpful because, as mentioned several times, we will adopt the approximation that all the post-main sequence mass loss occurs instantaneously in a single timestep.

3.3 Parameter dependencies

Our first task is to understand how changing the initial conditions affects the evolution of a single star simulated with SEVN. While for the mass and metallicity, we expect significant change, for the other two key run parameters, stellar spin and supernova formalism, this change is not immediate to quantify. To evaluate how changing parameters affect the simulation, we compare the total lifetime and the final remnant mass after changing spin and supernova formalism. We will assume the mass consumption due to thermonuclear fusion is negligible, this implies that the total mass injected in the ISM by a star in its lifetime is simply the difference between the star's initial mass minus the remnant. Comparing the mass of the remnant will allow us to estimate how much the different models for explosion and the spin will impact the total mass released in the ISM.

3.3.1 Spin parameter dependencies

We first concentrate on the spin parameter, this is defined as:

$$\Omega_s = \frac{\Omega}{\Omega_c} \tag{3.1}$$

where Ω is the angular velocity at the surface of the star and Ω_c is a critical angular velocity defined as the angular velocity at which the centrifugal force becomes as strong as the surface gravity of the star. We expect this parameter to not change the evolution of single stars since in all available tracks in SEVN are computed without the stellar rotation.

To confirm this expectation we ran various simulations for different stellar masses and different spins using an array of 20 spins spanning from $\Omega_s = 0$ to $\Omega_s = 1$. The results of these simulations show that no changes in stellar evolution are present. This was an expected result since all the stellar tracks in SEVN are computed without rotation. After verifying that rotation is not important we set this value at $\Omega_s = 0.1$.

3.3.2 Type II Supernova Formalism

The processes involved in core-collapse supernova explosions and the subsequent formation of remnants are still a very active subject of research. Throughout the years a large amount of theoretical and observational work has been performed to understand the mechanism of the explosion (Burrows & Vartanyan, 2021). While there is a general consensus on what generates the explosion nowadays, many authors have proposed various models that lead to different predictions for the mass and the type of remnant that a progenitor star leaves behind. Since ruling out models is a difficult task with the current observational evidence, many different formalisms are still very popular in the literature. It is for this reason that in SEVN four different SNII formalisms are available:

- Rapid formalism from Fryer et al. (2012).
- Delayed formalism from Fryer et al. (2012).
- Compact formalism from Mapelli et al. (2020).
- Deathmatrix formalism from Woosley et al. (2020).

The Rapid and Delayed formalisms have been widely used in recent years. The names Rapid and Delayed stand for the different timing for the reviving of the shock in the SN explosion. In both cases, the mass of the remnant depends on the mass of the C/O core.

In recent years, gravitational waves astronomy suggested that both Fryer formalims show some incosistency with the observational findings. Still the "Delayed" model seems to predicts better the mass distribution of the BH mass spectrum observed in the merger signals (Zevin et al., 2020). The Compact formalism is a very promising type of model, based on the compactness of the core defined as:

$$\xi_{2.5} = \frac{2.5}{R(2.5\ \mathrm{M}_\odot)/1000\,\mathrm{km}} \tag{3.2}$$

at the denominator, we find the radius at which 2.5 $\mathrm{M}_\odot$ are enclosed with 1000 km. We can use this parameter to determine if the star is going to expel the outer layers or go through a direct collapse into a black hole. Since in SEVN no precise stellar profile is given, they use the relation found in Mapelli et al. (2020) to correlate the mass of the C/O core with the compactness of a monotonic function. Since this relation is still debated, we decided to not consider this formalism, waiting for more consistent sets of results. The last formalism available is based on the results ofWoosley et al. (2020). In this work, they simulated pure He stars (i.e. stars made predominantly of Helium, without hydrogen).

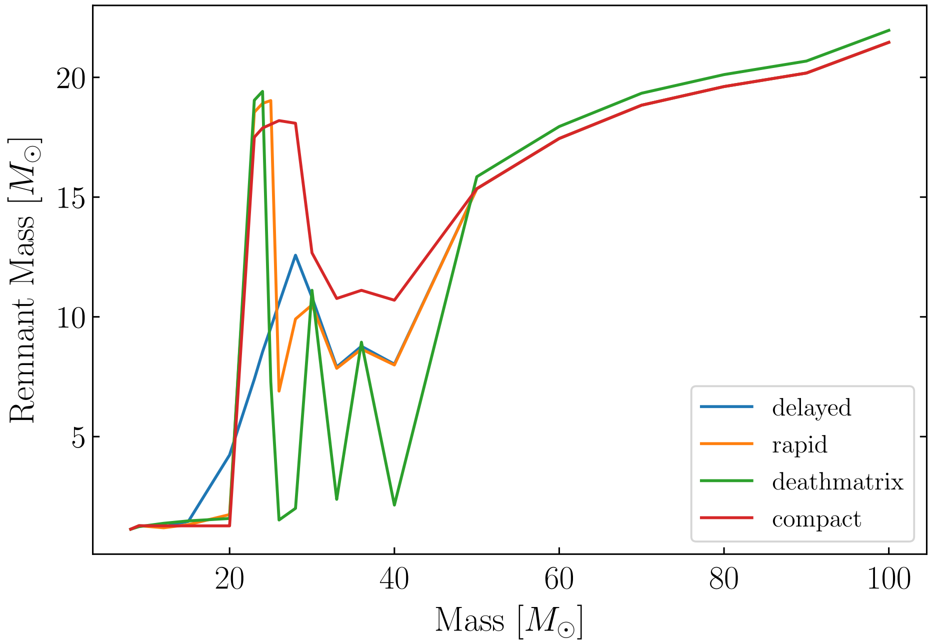


Figure 3.2: Remnant mass as a function of the initial stellar mass for the different SNII formalisms explored in this work. The simulations are performed with PARSEC tracks with $\lambda_{ov} = 0.5$, Z=0.02.

This implies that the mass of the compact object does not account for the mass of hydrogen left in the outer layer. We then modified the SEVN code adding a routine in which if the remnant mass is more than 99% of the stellar mass the event is considered as a direct collapse and all the mass of the hydrogen is added to the remnant, while, on the contrary, the hydrogen mass is expelled(see Appendix C for more detail on the manner).

After describing each selectable formalism, we started to study the mass of the remnant as a function of the initial stellar mass for different SN explosion models. As anticipated, this is critical to estimate the total mass injected back into the ISM.

The results of our tests (Fig. 3.2) show that different models of explosions predict different remnant masses, and so different mass injected into the ISM, as a function of the initial stellar mass. The difference between models of SNII explosion is also found for all the evolutionary tracks at all metallicities. The value of the remnant mass for Rapid, Delayed, and Compact formalisms assumes the same value for high masses ($M > 50\,\mathrm{M}_{\odot}$). The main differences between models are found in the mass interval $20\,\mathrm{M}_{\odot} < M < 50\,\mathrm{M}_{\odot}$. We can appreciate some common features like the threshold that separates the explosion from the direct collapse in a black hole, where the remnant mass becomes significantly larger. The main difference between the models emerges when, at fixed mass, the type of remnant they leave behind is not the same. For instance, if the star directly collapses into a black hole, the remnant mass would be much larger (and therefore the mass returned to the ISM consequently lower) than if the star ended its life as a neutron star. After verifying that spin does not influence the output, while the selected formalism does, we will discuss the construction of the new stellar evolution model in the remaining sections of the chapter.

3.4 End-of-life mass loss

In this section, we aim to construct the new hdf5 tables encoding the mass ejected at the end of the star's evolution using the output of the code SEVN. To achieve this, we first simulated ~ 100 stars at 10 metallicities for each mass, and then we chose a subset of simulated stars representing a better trade-off between an accurate representation of the SSP and the lowest possible number of points to save memory. In this way, we obtained the new sample of masses and metallicities that will be transcribed in the new .hdf5 tables. For each element of the new sample (i.e. a mass and a metallicity) we performed simulations for all the stellar tracks and SN formalisms available in SEVN. Notice that the lowest computable stellar mass for the PARSEC tracks is 2.2 $\mathrm{M}_{\odot}$. Therefore, to obtain a complete description of the entire SSP for all the tracks we compute the stars with mass $0.8\,\mathrm{M}_\odot \leq M \leq 2.2\,\mathrm{M}_\odot$ using the MIST tracks and then append the results to the ones obtained with the PARSEC model.

What follows is a description of the done steps to analyze and create the new stellar evolution model from the new stellar sample[2]. To achieve this, the choice of the stellar track and explosion model over which we perform the computation is critical. Hence, in the next subsections, we will also compare how different tracks and SN formalisms affect the resulting model.

3.4.1 Lifetime function and SNII rate

In this section, we describe how the new sample of simulated stellar masses better approximates the lifetime function and the end-of-life rate. For stars with $M \geq 8\,\mathrm{M}_\odot$ we sampled stellar masses as: [8, 9, 12, 15, 20, 23, 24, 25, 26, 28, 30, 33, 36, 40, 50, 60, 70, 80, 90, 100, 120] $\mathrm{M}_\odot$; in this regime we aim to describe more accurately the high mass range from 40 $\mathrm{M}_\odot$ to 100 $\mathrm{M}_\odot$. The old sample presented only 4 masses in this interval leading to an oscillating rate (Fig. 2.13). Also, we enlarge the sample from 20 $\mathrm{M}_\odot$ to 30 $\mathrm{M}_\odot$ since in this region we have the direct collapse of SNII (see subsection 3.4.2). If we do not carefully sample this area we risk not considering the direct collapse in the remnant mass (Fig. 3.2). For stars with $M < 8\,\mathrm{M}_\odot$ we chose: [0.8, 0.9, 1, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2, 2.1, 2.2, 2.3, 2.4, 2.5, 3, 3.5, 4, 4.5, 5, 6, 7] $\mathrm{M}_\odot$. The sample is pretty similar to the one in the old model. We just increased the number of masses around 2-3 $\mathrm{M}_\odot$ since in this region we have a peculiar behavior of the lifetime function (see Fig. 3.3). In the implemented code for the generation of the new .hdf5 tables, we treated separately the case of massive stars ending their life as SNII and low mass stars, which we refer to as AGB.

One of the main issues with the old model was the discontinuities found in the SNII rate and the injection mass rate (see Fig. 2.13 and Fig. 2.17). We take as a working hypothesis that such discontinuities were due to the coarse mass sampling of the old tables derived from Portinari et al. (1998). With SEVN we can refine the number of sampled stellar masses and verify this hypothesis. We construct a new lifetime function $\tau^S(M, Z)$ using the SEVN code, and we compare the old lifetime function and the new one in Fig. 3.3.

[2] All the code used in this section is available at `https://github.com/lorevannini/Thesis-project/tree/main/SEVN_model`.

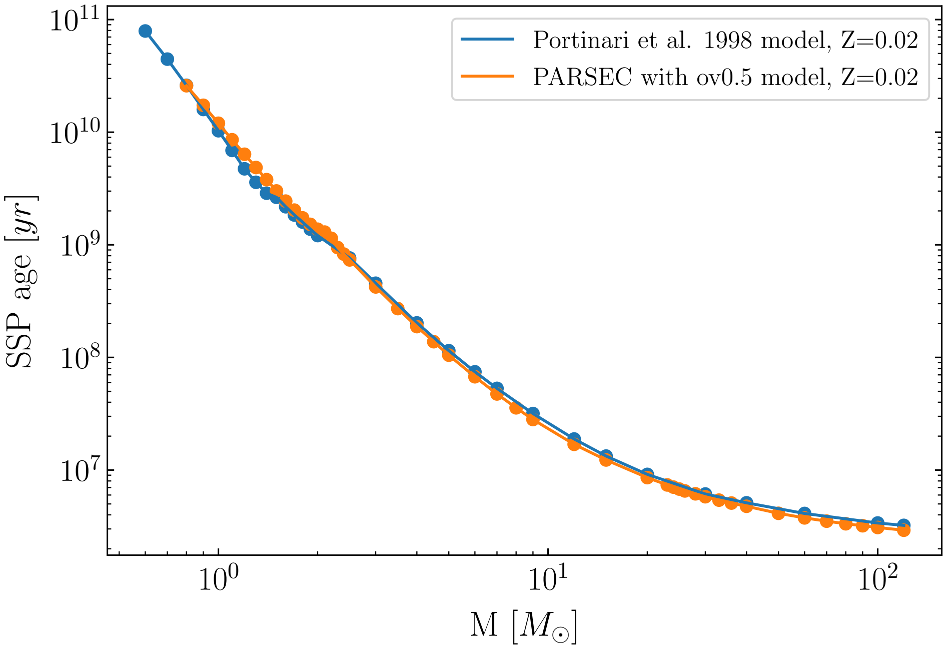


Figure 3.3: Lifetime function for the old (blue) and new (orange) SMUGGLE model for a metallicity $Z = 0.02$. Every track available in SEVN corresponds to a different lifetime function, in this plot, we show only the PARSEC track in order to emphasize the fact that the lifetime function of the new model is different from the old implementation used in SMUGGLE.

In Fig. 3.3 we show the lifetime function with the new sampling. Around 2 $\mathrm{M}_\odot$ the lifetime function has a complicated form. This feature is also present in the old lifetime function but is shifted to lower masses at higher times. We also enhanced the number of high-mass stars ($M > 10\,\mathrm{M}_\odot$), where the old model had a very thinned-out sampling (only 8 points from 10 $\mathrm{M}_\odot$ to 100 $\mathrm{M}_\odot$). Notice how the two functions are very similar to each other with the new model being a little bit lower in the high mass end. The procedure to obtain the SNII rate is analogous to the one described in section 2.3.1, in fact, the structure of the model is the same, we only change the data from which the lifetime of each star is interpolated. We then evaluate eq. (2.7) with the new lifetime obtained with SEVN. In such a way we get the rate of explosion of SNII. We report this result in Fig. 3.4.

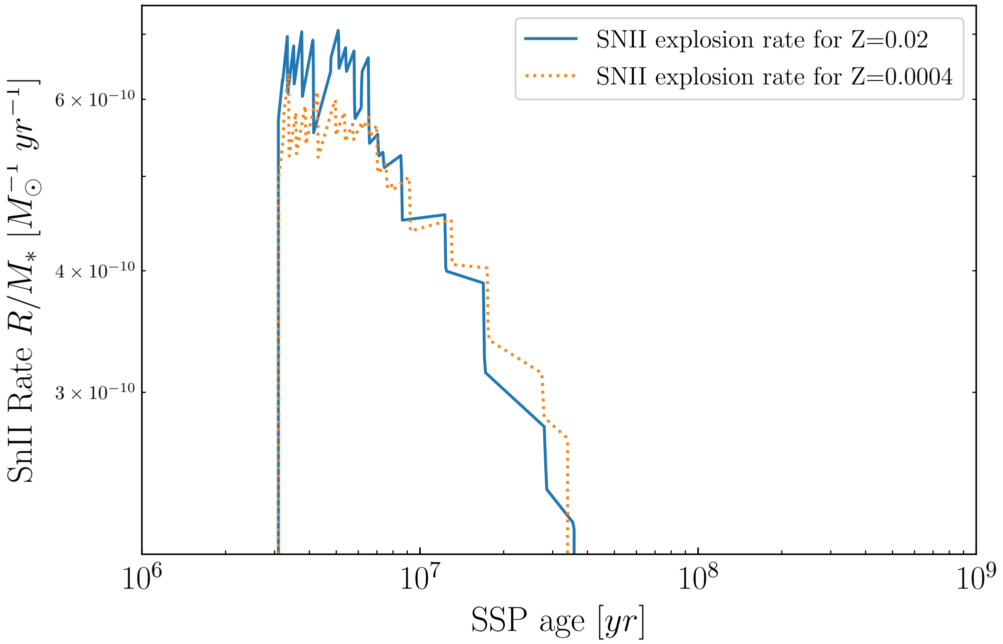


Figure 3.4: Specific SNII rate for the new model at the different metallicities as indicated in the legend. We used the PARSEC tracks with $\lambda_{ov} = 0.5$ for this plot. Notice how enhancing the number of masses leads to a less discontinuous rate compared with the old SMUGGLE model. The number of discontinuities is the same as the number of masses used in the interpolation, confirming the trend observed for the old model. Notice how at lower metallicity the rate decreases with respect to the one determined for $Z = 0.02$ for the massive stars with $M \geq 20\,\mathrm{M}_\odot$ and increases at the low-mass end.

In principle, we could enlarge the number of masses to have an even better depiction of the rate, but we need to keep in mind that these tables are to be used in (memory and computationally intensive) galaxy formation simulations. Thus, we are faced with a trade-off between an accurate description of stellar evolution and the keeping number of sampled masses as low as possible in order to save memory.

As previously mentioned, SEVN comes with various different stellar evolutionary tracks. Therefore, we investigate how choosing different tracks impacts the rate of SNII explosions. This is important since the mass ejection rate is closely related to the rate of explosion.

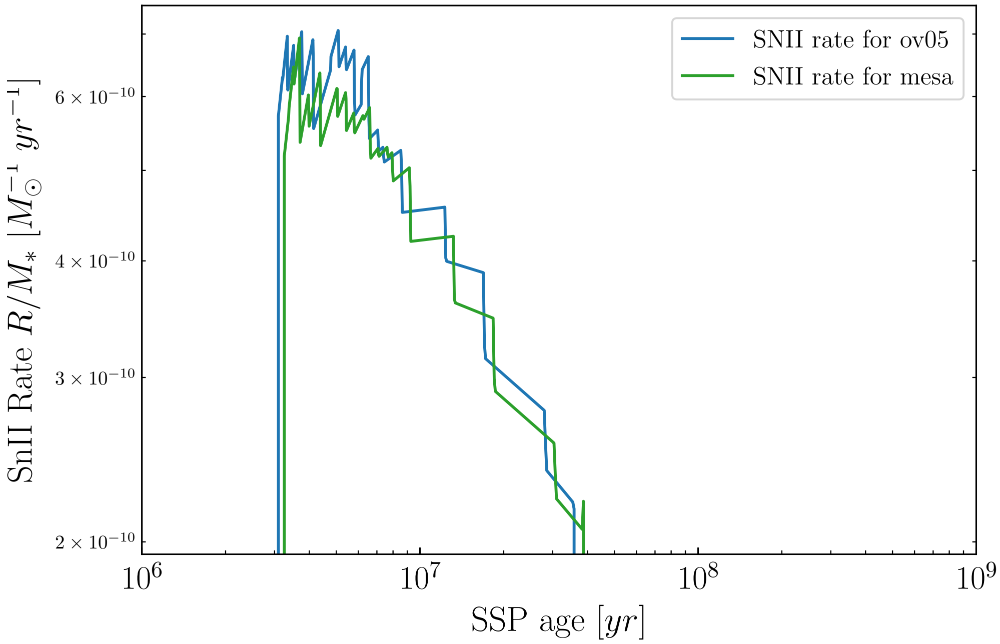
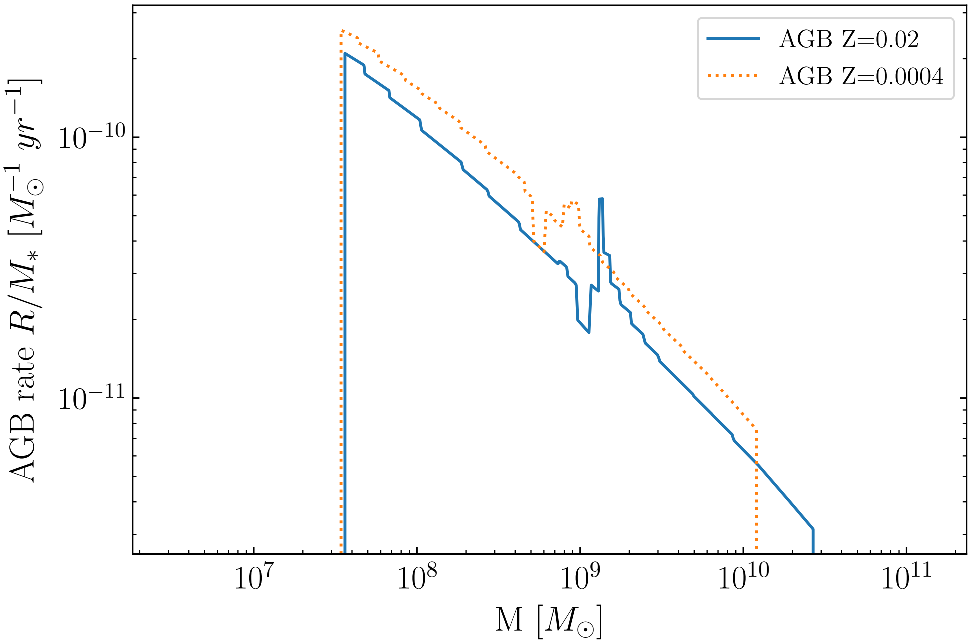


Figure 3.5: Specific SNII rate for the new model with MIST (green) an PARSEC (blue) tracks with $\lambda_{ov} = 0.5$. The two rates differ since the lifetime function from MESA has a different shape compared to the one in PARSEC.

In Fig. 3.5 is clear that changing the tracks changes the rate of SNII explosion, this is due to different lifetime functions from MIST and PARSEC. Notice that the MIST rate is lower and shifted to larger times relative to PARSEC. We expect the function describing the rate at which the mass is expelled by stars (mass ejection rate) to resemble the shape of the end-life rate. By repeating the same procedure described above, that is integrate the IMF using the inverse of the lifetime function as extremes for masses $0.8\,\mathrm{M}_\odot \leq M < 8\,\mathrm{M}_\odot$, we can also observe the rate of end-life events for low-mass stars. We plot this in Fig. 3.6.



[H]

Figure 3.6: AGB end-life-ate rate using the new sampling, we used the PARSEC tracks with $\lambda_{ov} = 0.5$ to obtain this plot. Notice how The lifetime of the stars reduces at low metallicity. This implies an increase in the event rate enlarges.

In Fig. 3.6, we notice how around 10^9 yr we have a spike in the rate. This effect is due to the lifetime function becoming flatter for $\sim 2\,\mathrm{M}_\odot$, around this mass the lifetime function has a lower steepness, and this reduced steepness makes the rate of events increase. To ensure that this spike has a physical origin and it is not an interpolation artifact, we used for this mass interval a very fine mass sampling. The result obtained with the finer sample resemble the one reported in this figure. After finding a new set of masses for the sampling and discussing the new end-life stellar rates, we proceed to present the efficiency function and the mass ejection rate of the new model in the next subsections.

3.4.2 Efficiency function of the new model

We want to quantify the mass ejected by a star using the SEVN code and then write the results in a .hdf5 table readable by AREPO. To do this we need to determine the mass ejected by each star in the sample. SEVN's output file contains a column with a string that specifies the evolutionary state of the star (see Fig. 3.1). We use this column to discriminate two phases: main-sequence and post main-sequence. All the mass loss during the main-sequence are considered as continuous stellar winds, while all the mass loss happening after the main-sequence is released in the end-of-life event. We can obtain the end-life mass loss by subtracting the mass of the remnant from the mass of the star at the last timestep before leaving the main sequence. As we can see in eq. (2.8) the mass ejection rate depends on the efficiency function $f_{rec}(M, Z)$. This function is defined as the ratio between the ejected mass in the post main-sequence phase (i.e. the difference between the mass entering the post main-sequence and the remnant mass) and the initial stellar mass (the mass at the ZAMS). We use the mass return efficiency function $f^S_{rec}(M, Z)$ (we used the index S to indicate the new function obtained with SEVN) to study in detail the new model. In this subsection, we first quantify how metal content affects the ejection of mass for all stars. Then we analyze how the efficiency is affected by the choice of the stellar track available and lastly, we investigate the differences that emerge from changing the SN explosion model. In Fig. 3.7 we see how the function $f^S_{rec}(M, Z)$ depends on the initial stellar mass and metallicity. To make this plot we used the PARSEC tracks with $\lambda_{ov} = 0.5$ and the delayed SN formalism.

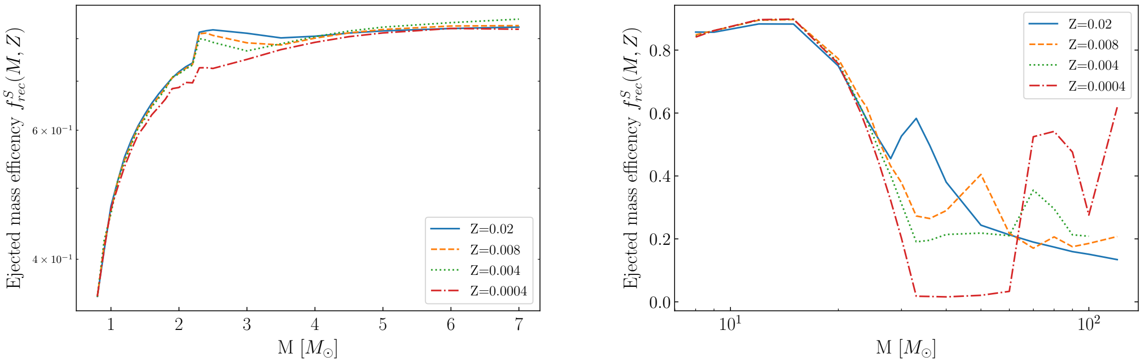


Figure 3.7: $f^S_{rec}(M, Z)$ as a function of masses plotted for the metallicities indicated in the legend. We separated low-mass stars that end their life as planetary nebula (left) and the ones that explode as SNII (right).

In Fig. 3.7 we see that while for $M < 20\,\mathrm{M}_\odot$ we have a mild dependency on metallicity, for higher mass stars metal content influences dramatically the shape of the function. We see a sudden drop at $M \sim 20\,\mathrm{M}_\odot$. This drop corresponds to the lower stellar mass at which the remnant is a black hole, making the end of life a direct gravitational collapse. The general trend with metallicity is clear. When the metal content of the star is decreasing efficiency decreases, so that less mass is returned on average at a fixed stellar mass. This stops to be true for massive stars with $M > 60\,\mathrm{M}_\odot$ where the trend seems to invert, this is due to mass losses from pair instabilities at high metallicity (see Figure 8 in Iorio et al. 2023). However, we still expect less mass injection at lower metallicities; in fact

stars with $M > 60\,\mathrm{M}_\odot$ contain a small percentage (about 3% for a Chabrier IMF) of the total mass of the SSP, whereas stars with $8\,\mathrm{M}_\odot \leq M \leq 60\,\mathrm{M}_\odot$ instead contain 20% of the total mass. For the other evolutionary tracks and SN formalisms, we get a similar trend of $f^S_{rec}(M, Z)$ with metallicity.
We then evaluate $f^S_{rec}(M, Z)$ with different evolutionary tracks to see how these changes affect the efficiency function.

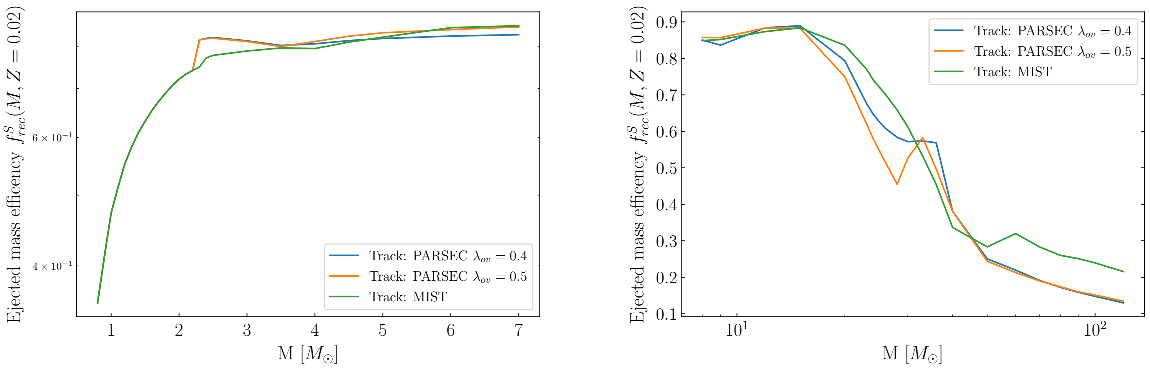


Figure 3.8: $f^S_{rec}(M, Z)$ as a function of mass plotted for the three evolutionary tracks available in the SEVN code at solar metallicity using the delayed SN formalism. Also in this case we separated the plot into two parts for low-mass stars (left) and SNII end-life events (right). The label "mesa" indicate the MIST tracks.

We present these results in Fig. 3.8. The two PARSEC tracks resemble each other, especially at the high mass end of the plot ($M > 60\,\mathrm{M}_\odot$) and for $M < 7\,\mathrm{M}_\odot$. We would like to point out that for $M < 2.2\,\mathrm{M}_\odot$ the efficiency becomes exactly the same since only MIST tracks can be used in this mass range. In general, we can observe common features between the models, such as the drop of f^S_{rec} at $M \sim 20\,\mathrm{M}_\odot$ and a decreasing trend in the function for $M > 20\,\mathrm{M}_\odot$. We then conclude that we will need to investigate more in detail how changing the tracks changes the total mass return fraction (i.e. how much mass the SSP returns to the ISM). As we discussed in section 3.3, we also expect the chosen SN formalism to affect the mass returned, if this is true, we should have significant differences in the efficiency of mass injection. To analyze this. In Fig. 3.9 we plot f^S_{rec} using PARSEC with $\lambda_{ov} = 0.5$ at $Z = 0.02$.

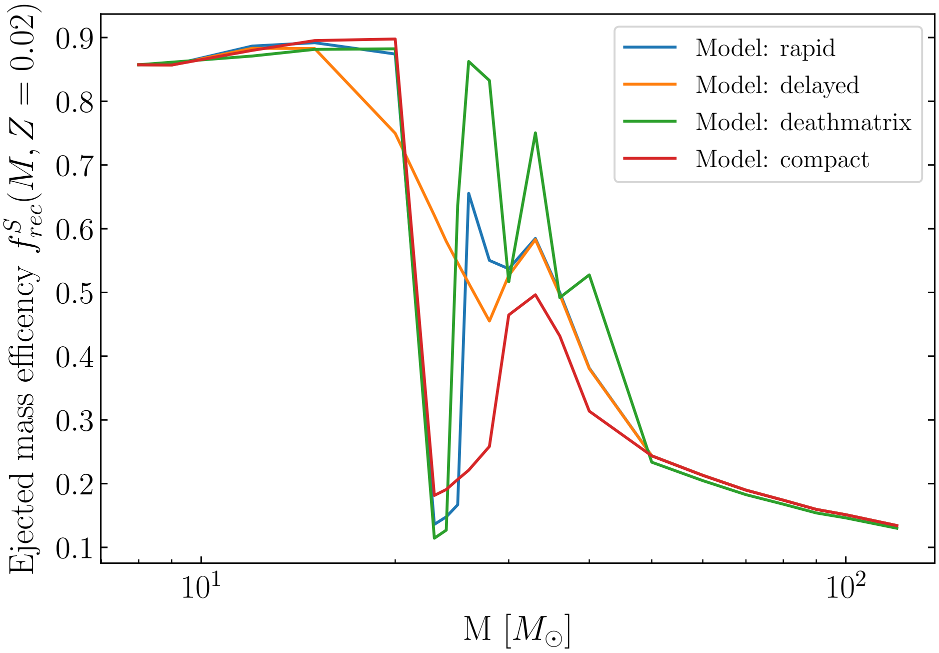


Figure 3.9: $f^S_{rec}(M, Z)$ as a function of masses plotted for the four SN formalisms available in the SEVN code at solar metallicity. Notice that for $M > 50\,\mathrm{M}_\odot$ and $M < 12\,\mathrm{M}_\odot$ the function are the same

All the models have a sudden drop at $\sim 20\,\mathrm{M}_\odot$, the "delayed" model shows a less prominent drop and a more smooth overall shape of the efficiency function. Given the difference between models in Fig. 3.8 we expect different mass ejection rates and total mass ejection fraction if we change the SN formalism used. Notice how the real difference between all of the four formalisms is in the range $20\,\mathrm{M}_\odot - 50\,\mathrm{M}_\odot$, at lower and higher masses the function is essentially unchanged. The differences in the mass range $20\,\mathrm{M}_\odot - 50\,\mathrm{M}_\odot$ are mostly due to the uncertainties on the BH remnant mass, which strongly depends on the explosion mechanism considered. We have investigated the new efficiency function and all its possible declinations depending on the tracks, SN explosion model, and metallicity. We confirmed an overall decreasing trend of ejection efficiency with metallicity. To understand and predict how the new model will affect the mass released in the simulation we will calculate the mass ejection rate and the total mass ejection fraction in the next subsections.

3.4.3 Mass ejection rate

In the SMUGGLE code, the mass released by a single stellar particle depends on the phase of the evolution of the stellar population encoded. For each timestep, a percentage of the initial mass is released from both stellar winds and end-life stellar events. Evaluating the mass ejection rate enables us to understand how fast the gas in an SSP is released back to the ISM and can be reused, for example, to make new stars. In this subsection, we adopt a structure simular to subsection 3.4.2. At first study how the mass ejection rate is affected by the metal content, we then study different evolutionary tracks and finally we concentrate on the SN explosion model. Once we obtained the efficiency function the procedure to calculate the mass ejection is the same as the one used for the older model. Namely, we solve numerically the integral (2.8) with the trapezoids method, using the new lifetime function $M^S(t, Z)$ and the new efficiency function $f^S_{rec}(m, Z)$

$$\Delta M^S(t, \Delta t, Z) = \int_{\mathcal{M}^S(t+\Delta t)}^{\mathcal{M}^S(t)} m f^S_{rec}(m, Z)\phi(m)dm \tag{3.3}$$

To asses the dependence on the metallicity of the new mass ejection rate function, we plot the mass ejection rate of the $\lambda_{ov} = 0.5$ track for two different metallicity values in Fig. 3.10.

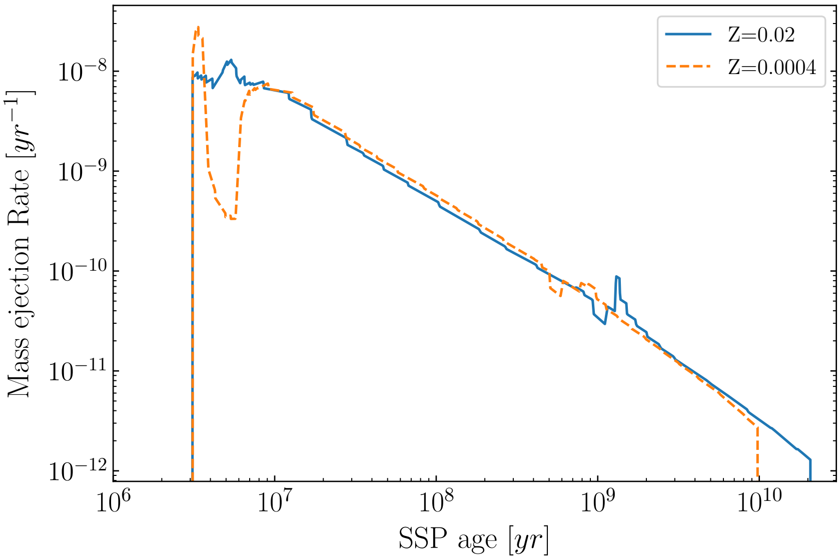


Figure 3.10: Specific mass ejection rate by the entire SSP using SEVN, we used the delayed formalism with the PARSEC $\lambda_{ov} = 0.5$ tracks. We analyze here how the different metallicities used impact the overall mass ejection rate from end-of-life events. Notice how the feature we discussed the efficiency function is present also in the mass ejection rate (i.e. metallicity dependence, a spike in the AGB phase, spike at high mass for BH formation threshold).

f^S_{rec} is responsible for the observed shapes, as we can notice comparing Fig. 3.10 and Fig. 3.7. For high mass (so for early times) the mass ejection rate is higher for lower metallicity. We have a sudden drop that is not present in the solar metallicity case, this is completely analogous to the features we observe in Fig. 3.7. Is also worth noticing how even if the efficiency function for stars with $M < 20\,\mathrm{M}_\odot$ is smaller for lower metallicity, the ejection rate remains higher until $t \sim 3 * 10^9$ yr because lower metallicity stars live less. This effect increases the end-life rate and consequentially the mass ejection rate. The shape of the end-life rate is also responsible for the bump around $\sim 10^9$ yr. This tends to be more prominent for higher metallicities, while for lower ones is shifted to earlier times and is less prominent. After assessing the metallicity dependence, we aim to understand how the mass ejection rate changes if we change the stellar tracks used for the computation. We show the ejection rate for different evolutionary tracks in Fig. 3.11.

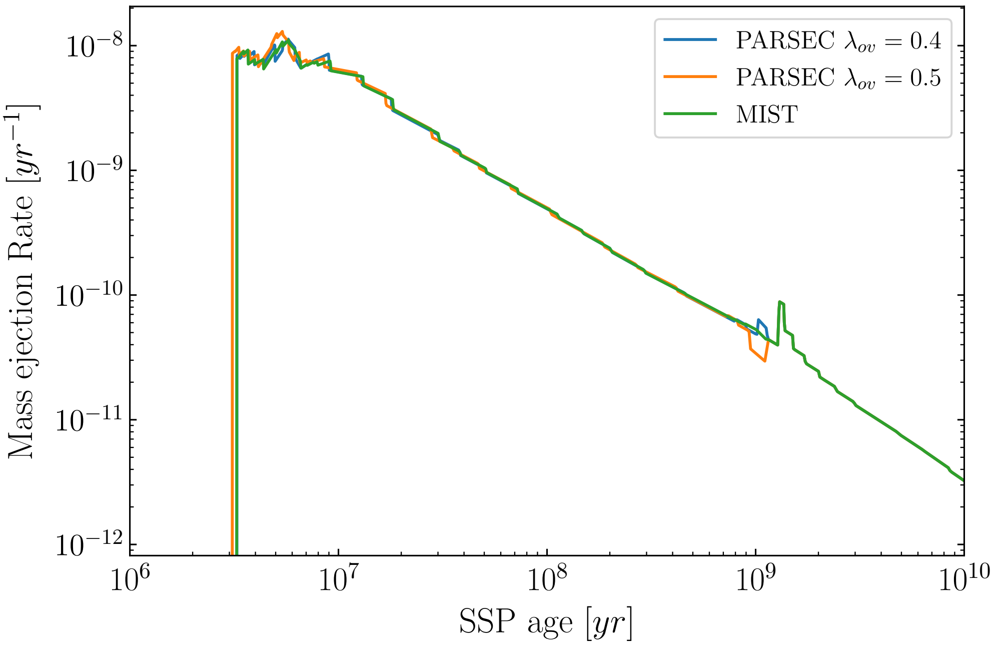


Figure 3.11: Specific mass ejection rate by the entire SSP using SEVN for Z=0.02. We report here how the different tracks used impact the rate. Notice how the features we discussed about the efficiency function are present also in the mass ejection rate.

The three lines corresponding to the three different stellar tracks present a very similar shape, only minor differences can be found in the ejection rate. This similarity between the different tracks used is observed also if we repeat the computation with other SN models. Notice how depending on the track used the behavior of the ejection rate near the bump at $\sim 10^9$ changes. After this time the rate is the same for all tracks since only the MIST tracks can be used. Finally, we plot how the SN formalism affects the mass ejection rate Fig. 3.12.

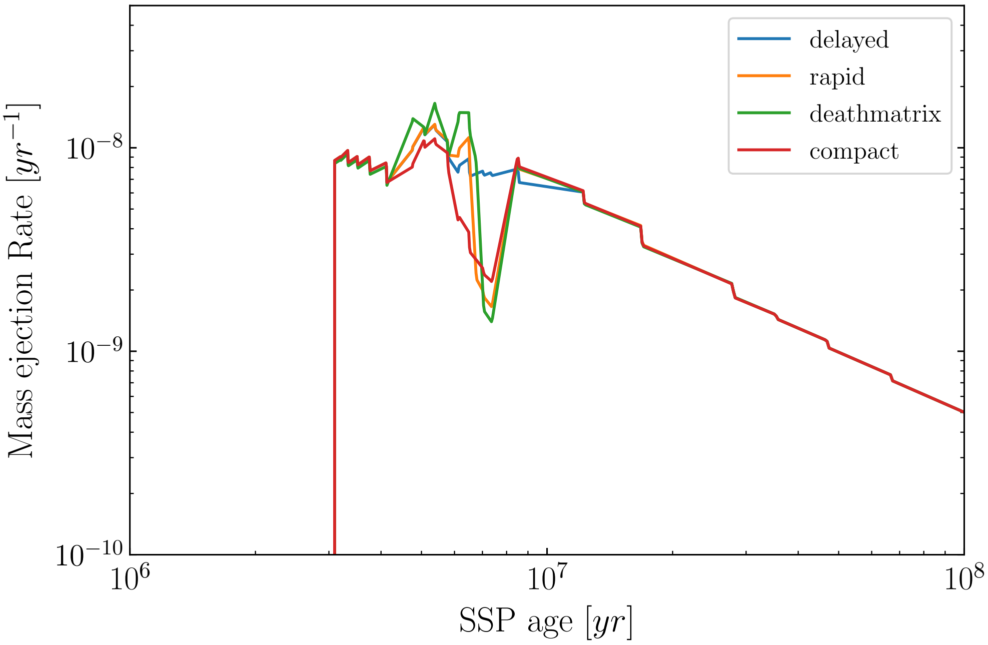


Figure 3.12: We report here how the different SNII formalism used impacts the rate calculated with $\lambda_{ov} = 0.5$ tracks and Z=0.02. We reported only the early time part of the mass ejection rate since it is the only part effected by the SN explosion mechanism.

What we can notice in Fig. 3.12 is that Compact, Rapid, and Deathmatrix present a dip in the mass ejection rate due to the formation of a BH remnant. Instead Delayed model does not show this feature. This difference for the delayed model is due to the different behavior we can observe in the efficiency function in Fig. 3.8. The part of the plot in which we have a strong dependence on the model occurs for ages between $2 * 10^6 - 10^7$ yr. This is the region corresponding to stellar masses in the range $20\,\mathrm{M}_\odot < M < 60\,\mathrm{M}_\odot$, where we have the most uncertainties about the remnant mass and consequently about the mass loss of the star.

For every timestep, the SMUGGLE code calculates for the stellar particle the total mass ejected for the duration of that timestep (i.e. the numerical integral of the ejection rate). To have a complete understanding of the impact of different parameter choices in the model we need to calculate the total (cumulative) mass fraction ejected by the stars. This will be the topic examined in the next subsection.

3.4.4 Total mass ejection fraction

To have a complete understanding of the impact of the different initial parameters of SEVN on the evolution of an SSP, we need to calculate its total ejected mass fraction. In fact, one of the main goals of this work is to understand if an updated stellar evolution model will return a different mass to the ISM. In this subsection, we investigate how changing metallicity, tracks and SN explosion mechanisms affect the cumulative mass fraction function and we quantify the total mass returned in the ISM by the end-life phase of a simple stellar population. We calculate the integral function in time of the mass ejection (i.e. we sum all the ΔM in eq. 3.3 until time t) to get the total mass released by stars in the final stage. To do this we evaluate the following integral using the trapezoid method:

$$M_{tot}(t, Z) = \int_{t_0}^{t} \dot{M}(t, Z)dt, \tag{3.4}$$

Where t_0 is the simulation time at which the SSP is initialized (we take, without loss of generality, $t_0 = 0$ for convenience), t is the age of the SSP and $\dot{M}$ is the mass ejection rate that we discussed in section 3.4.3. $M_{tot}(t, Z)$ is the cumulative function of the mass returned in the ISM.

We first investigate how changing the metallicity leads to different total cumulative mass ejection. We report the results in Fig. 3.13.

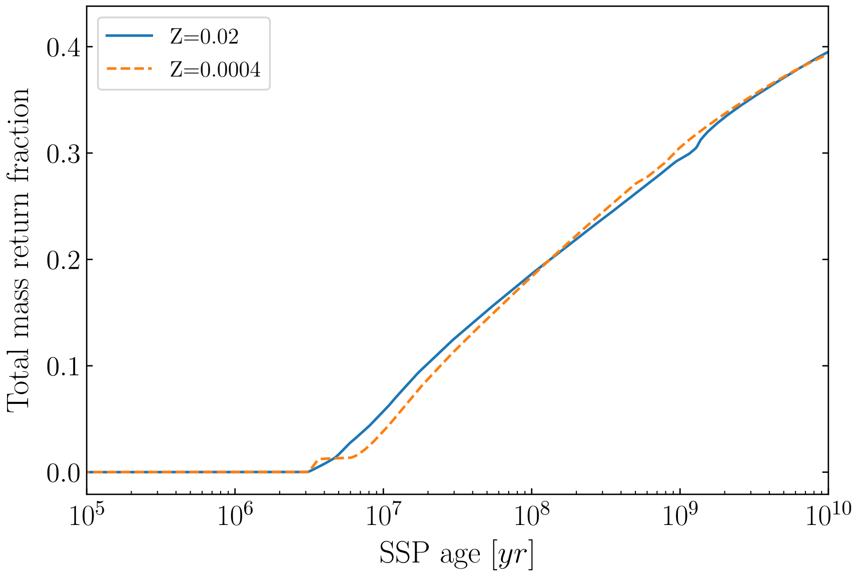


Figure 3.13: Total cumulative mass ejection fraction by the SSP obtained employing SEVN and PARSEC tracks with $\lambda_{ov} = 0.5$ and the delayed formalism for SNII. Two different values for the metallicities are reported, as indicated in the legend.

We notice how at 10^{10} yr the mass fraction function assume basically the same value for different metallicity values. If we observe the efficency function for different metallicity (Fig. 3.7), this result might not be expected. The equivalence between the total mass returned of the two metallicity is caused by the shorter lifetime of lower metallicity stars, resulting in a higher end-life rate (see Fig. 3.6). In fact a shorter lifetime of stars makes for a wider range of stars exploding in the same timestep, enhancing the mass ejected. For this reason in Fig. 3.13 we observe an steeper curve corresponding to Z=0.0004 respect to the Z=0.02. For the lower metallicity $Z = 0.0004$ we have a section of the plot between 4 Myr and 8 Myr where the cumulative function flattens, this correspond to the decrease in the efficiency function observed in Fig. 3.7. We also notice that the bump in the ejection rate at around 10^9 yr correspond to a sudden increase of the cumulative function, this is more visible for the $Z = 0.02$ case. If we compare the results of different metallicities using plots obtained with other tracks and SNII explosion models, we obtain similar dependencies on Z.
We then report in Fig. 3.14 how changing stellar tracks impacts the total mass ejected.

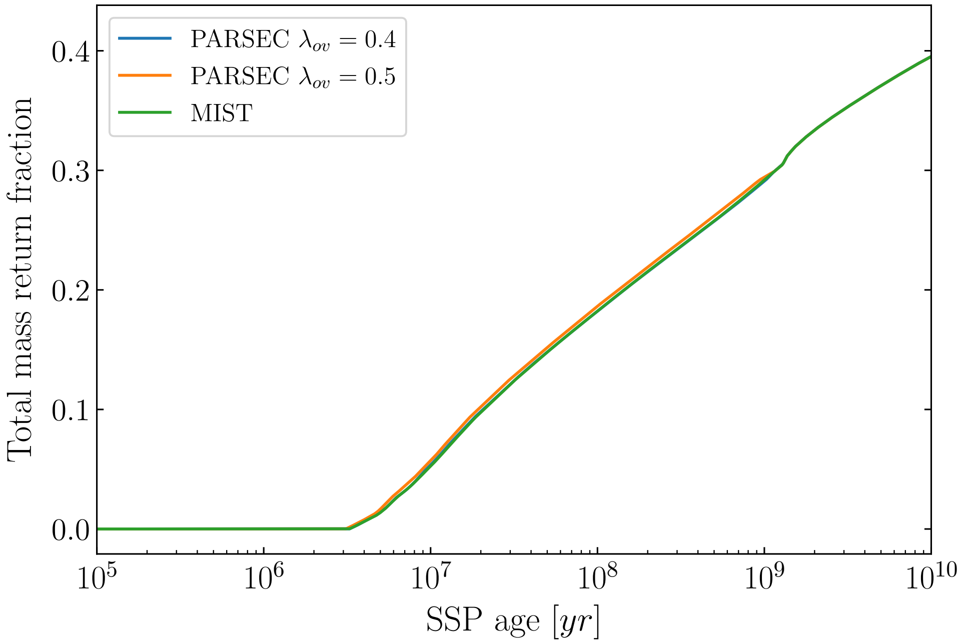


Figure 3.14: Total cumulative mass ejection fraction using SEVN. This plot shows the dependence if the total mass return fraction on the adopted stellar tracks.

In the figure, we see how changing tracks does not significantly influence the total returned mass fraction for end-life stellar mass loss. This is an important result. We showed that with the current implementation of stellar evolution in our MW-type galaxy simulation the overshooting parameter, in the range we are able to investigate, does not seem to influence the total mass return fraction. This imply that for the range investigated in this work, the overshooting parameter do not influence the total mass return fraction. The similarity in the cumulative ejection fraction was expected since the mass ejection rate in Fig. 3.11 does not differ much between stellar evolutionary tracks. The cumulative function smooths the differences between the ejection rates from different tracks even more. As already discussed, for low-mass stars we use only Mesa tracks. All the cumulative functions from different tracks are the same when we reach $\tau^S(2.2\,\mathrm{M}_\odot)$ end-life time, this makes all the cumulative functions identical.

We now analyze how the explosion models for SNII explosions affect the total mass ejected.

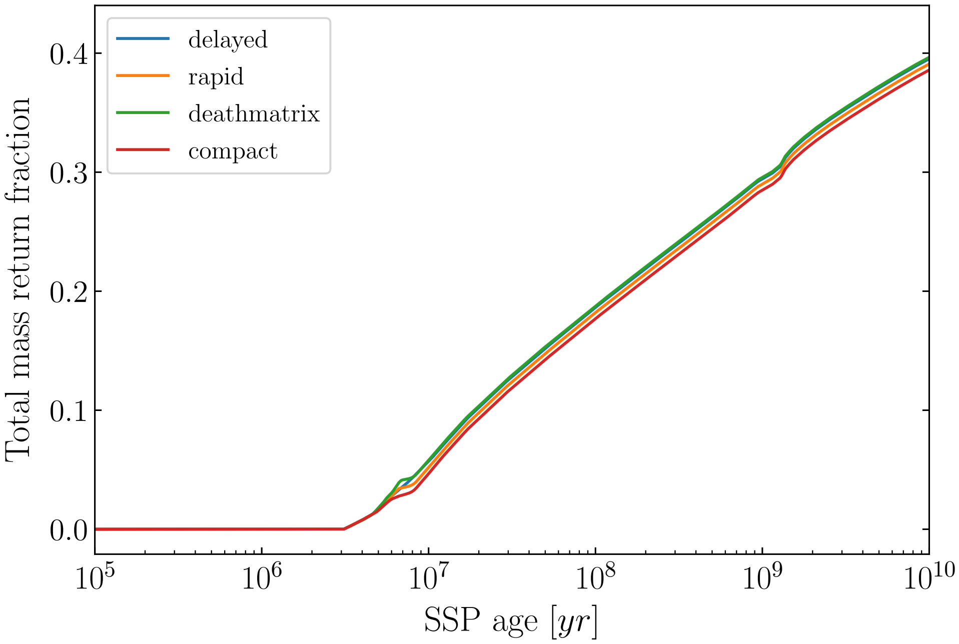


Figure 3.15: Total mass ejection fraction by the SSP using SEVN.For this plot, we used the different formalism for SNII explosion at the solar metallicity using the $\lambda_{ov} = 0.5$ track from PARSEC, as indicated in the legend.

In Fig. 3.15 we see how different models lead to different cumulative mass ejected. The difference is around the 2% of the total mass of the SSP. The discrepancy is much more pronounced with respect to the case in which we use different evolutionary tracks (Fig. 3.14). We notice how the main cause that leads to different results is found in the region between 4 Myr and 8 Myr, the same region where we have the largest discrepancy between the models in the mass ejection rate (Fig. 3.12). Even if the mass ejection rate in the corresponding part of the plot differs by 1 order of magnitude between models, we do not see a prominent change in the total mass ejected. This is due to the fact that most of the mass of the SSP is in stars that have $M \leq 10\,\mathrm{M}_\odot$ (i.e. 80% of the total mass of the population for Chabrier IMF) where we do not have much difference between different tracks or SN explosion models. All these considerations allow us to choose a fiducial track and a fiducial SNII formalism that represent the other possible combinations as well.

3.5 Stellar Winds

The goal of this section is to build a novel model for the continuous mass ejected via stellar winds from the SSP. In order to do this, we need to adopt the following steps. In the output file from SEVN, we select the columns containing the time and the numerical discrete derivative of the mass of the star, i.e. the mass ejection rate from winds of a single star (Fig. 3.1).

We interpolate for each star the output's times with the corresponding mass ejection rate value throughout the main sequence phase. We call the interpolated function describing the mass ejection rate of a single star at every time $W(M,t)$. The adaptive timestep method used in SEVN makes the times saved in the output different for every simulation. We need to characterize the mass ejection rate at some common time for all the stars in the adopted mass sampling. This is a mandatory step if we want to get the total mass ejection rate from the winds of all the stars in the SSP. After initializing a "universal" time t we evaluate for all the sampled stellar masses and for all the selected time values the function $W(M,t)$. Our goal is to obtain the mass ejection rate from winds of all the SSP at any given time t. To get this we need to take the W evaluated at every time, weight it with the adopted IMF to take in account the population of stars at a specific mass and integrate it over a prescribed mass range. This results in

$$\dot{M}_w(t) = \int_{0.1M_\odot}^{\min\{\mathcal{M}(t,Z),100\,\mathrm{M}_\odot\}} W(m,t)\phi(m)dm \qquad (3.5)$$

In eq. (3.5) the integration is performed in the mass interval $0.1\,\mathrm{M}_\odot < M <$ $< \min\{\mathcal{M}(t,Z),100\,\mathrm{M}_\odot\}$. If all the stars are alive we use the range in which the IMF is normalized if instead at time t some stars already left the main sequence, the integration is performed up to the most massive star that is still in the main sequence phase. Adopting this range of masses in the integral 3.5 means that we take the mass loss from stars $M \leq 0.8\,\mathrm{M}_{\odot}$ negligible, this because our sample of masses simply does not have stars with masses lower then $0.8\,\mathrm{M}_{\odot}$. We justify this by saying that stars in the range $0.8\,\mathrm{M}_{\odot} < M < 8\,\mathrm{M}_{\odot}$ gives basically no net contribution to the total mass ejected via winds (see Fig. 3.17). It is then safe to assume that even lower mass stars ($M < 0.8\,\mathrm{M}_{\odot}$) will not contribute to the total mass ejected.

We can integrate with trapezoids eq. (3.5) and get the mass ejection rate produced by stellar winds. We show the rate obtained with this procedure for different metallicities in Fig. 3.16.

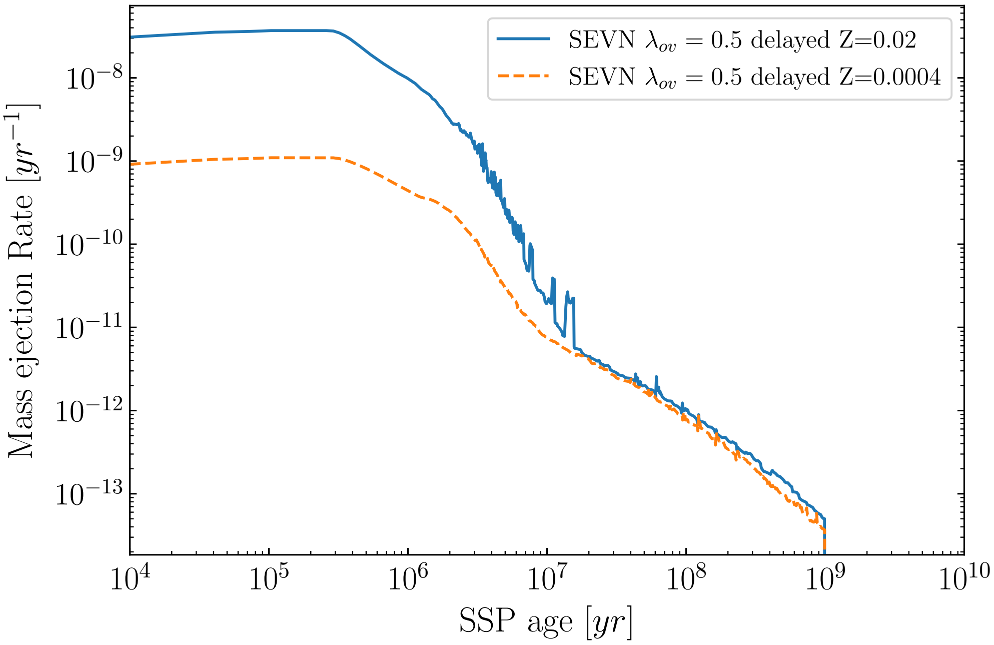


Figure 3.16: Mass ejection rate from the continuous mass losses (winds) for all the SSP. We plotted results for different metallicities as indicated in the legend. This was obtained using the delayed formalism. In this case, we show the results obtained with the PARSEC $\lambda_{ov} = 0.5$ tracks.

Notice how the winds contribute to the mass ejection at approximately the same rate as the SNII until $t \sim 10^7$ yr. Then, very massive stars explode and the wind contribution becomes almost 4 orders of magnitude less than the one at the beginning of the SSP evolution and continues to decrease. We expect winds to be a relevant part of the mass ejected especially in the very first part of the SSP life. For low metallicity the wind contribution is almost 30 times less respect to the solar metallicity case, showing a clear dependence on the metal content. We do not report plots for different SN explosion models and different stellar tracks since in our work we found that this choice does not

have a significant impact on the total mass returned.
Finally, we can integrate the ejection rate eq. (3.5) and get the total mass fraction ejected by winds. We report this in 3.17.

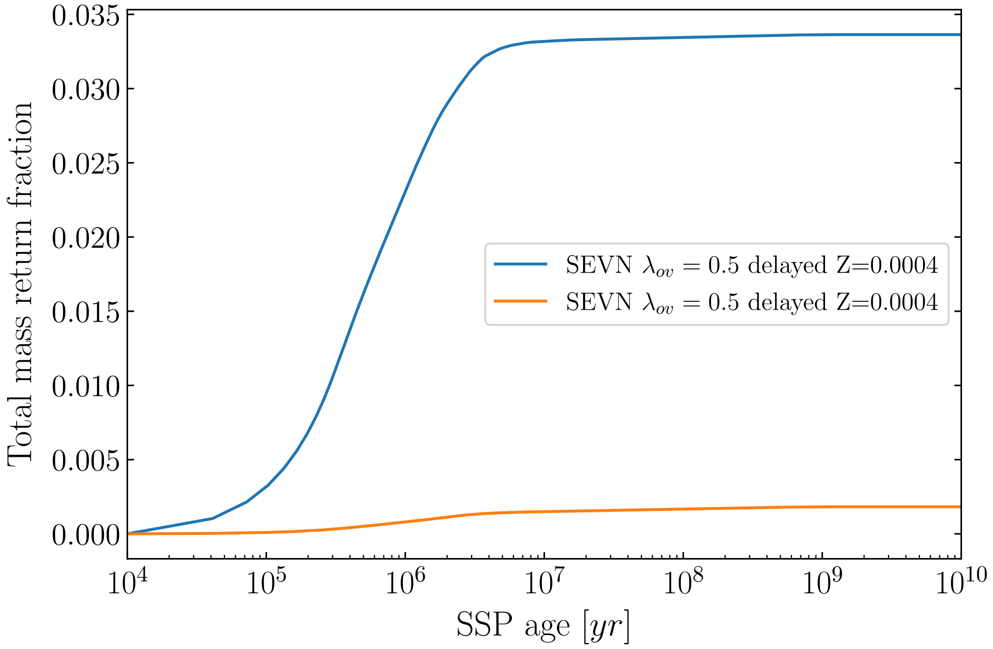


Figure 3.17: Total mass ejection fraction from winds of the SSP. We plotted result for different metallicity, as indicated in the legend. Also in this case we used the PARSEC $\lambda_{ov} = 0.5$ tracks and delayed formalism for SNII explosions; all the other choices give equivalent results.

The trend with metallicity suggests that the mass loss at low metallicity is essentially negligible.

3.5.1 Interpolating Mass Ejection Rate by Stellar Winds

Implementing the procedure described above to calculate the contribution of stellar winds to the mass loss in SMUGGLE code would have the disatvantage of a high computational cost. Therefore, we decide to implement the new model for winds in a more effective way. We sample the $\dot{M}^S(M,t)$ function at ~ 20 representative points. Then we store the sampled data in a hdf5 file that is readable by SMUGGLE. This procedure is similar to the one adopted for the end-life mass ejection, but in this case we sample times and not stellar masses. This a crucial change since sampling times makes the mass return from winds dependent on chosen the stellar evolutionary tracks and the IMF needed to perform integral in eq. (3.5).

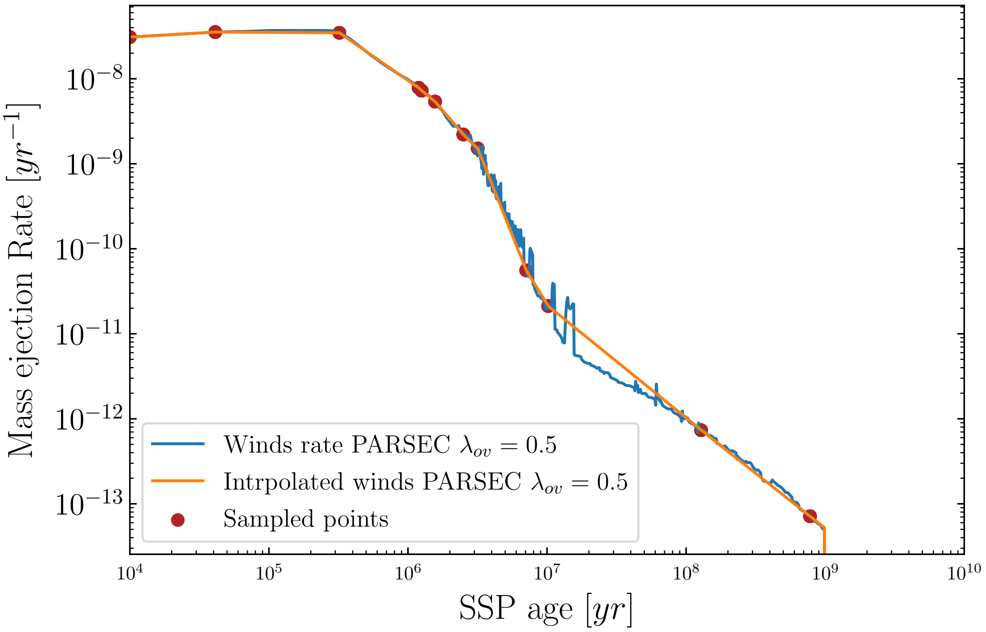


Figure 3.18: Mass ejection rate from winds from the integral 3.5 compared with the sampled one. We used the $\lambda_{ov} = 0.5$ tracks and show the outcome for solar metallicity.

In Fig. 3.18 we see how thissampling was performed. Notice how we choose to not make a finer sampling the range from 10^7 yr to 10^8 yr. This is because the winds ejection rate is mostly a concave function. This implies that the linear interpolation will inevitably under predict the value of the function calculated numerically in eq. (3.5). To understand how much the sampled function differs from the original function obtained numerically, we compare the total mass fraction (i.e. the cumulative integral of the wind mass return) of the two cases (Fig. 3.18).

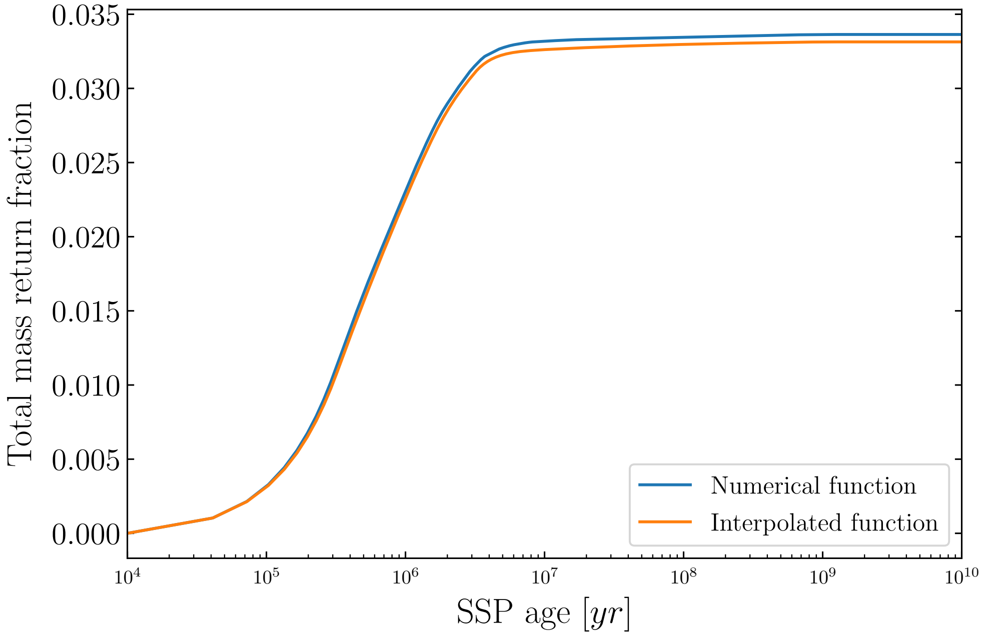


Figure 3.19: Mass ejection rate from winds from the integral 3.5 compared with the sampled one. We used the $\lambda_{ov} = 0.5$ tracks and show the outcome for solar metallicity.

We notice how even if we adopted the caveat of under sampling convex region the cumulative integral of the sampled winds ejection rate is less then the numerical one. This is due to the fact that most of the ejection happens from 10^5 yr to 10^7 yr where the function is concave. In the region where we under sample the ejection rate is 4 order of magnitude less then the initial value for winds, so we expect a very mild effect. Notice that the sampled winds fall short of less then 1/1000 of the calculated winds, which we can consider negligible.

4

Comparison and discussion

4.1 Introduction

In this thesis, we presented various approaches for implementing stellar evolution in galaxy formation simulations. In particular, we analyzed two widely used approaches: the FIRE and the SMUGGLE models. In FIRE-3 (Hopkins et al., 2022) analytic fitting functions were adopted to parameterize stellar evolution and in particular the rate of SNII explosions and mass ejection. Our work, instead, focuses on the approach used in the SMUGGLE model (Marinacci et al., 2019), originally introduced in Vogelsberger et al. (2013), in which pre-computed tables are used to reconstruct the stellar evolution. The goal of this chapter is to compare and discuss these methods and try to interpret how their different outcomes would impact a simulation of a Milky Way type galaxies. This is done by comparing the mass ejection rate and the total mass ejected by the modeled stellar populations in FIRE and SMUGGLE (with both the old approach and the new SEVN tables developed in this work). Then we compare the mass of metals and the single elements returned by the two models. As anticipated in the previous chapter, we need assume fiducial parameters for the new model. In particular, we use PARSEC tracks with overshooting parameter $\lambda_{ov} = 0.5$ and delayed SN formalism for SNII explosions. We chose these specifics for various reasons. The $\lambda_{ov} = 0.5$ tracks are the ones used as a reference in all works that involve the SEVN code (Iorio et al. 2023), while the delayed formalism presented in Fryer et al. (2012) is well corroborated. Also, using the delayed formalism has the advantage of setting an upper limit in the mass fraction returned for all the other models (Fig. 3.15).

4.2 Mass ejection rate

In this section, we compare the mass ejection rates obtained with the different models analyzed in this thesis. We start by comparing the mass ejection rate of Hopkins et al. (2022), Portinari et al. (1998) (used in the original SMUGGLE implementation), and the new model we built with SEVN. We consider all the contributions to the mass ejection, namely stellar winds, SNII explosions, and AGB mass ejection. We do not consider at this stage SNIa for various reasons. Namely, from the previous analysis (see Fig. 2.2.1), we demonstrate that their contribution to total mass return is negligible, and the two parameterizations from Hopkins et al. (2022) and Marinacci et al. (2019) are equivalent.

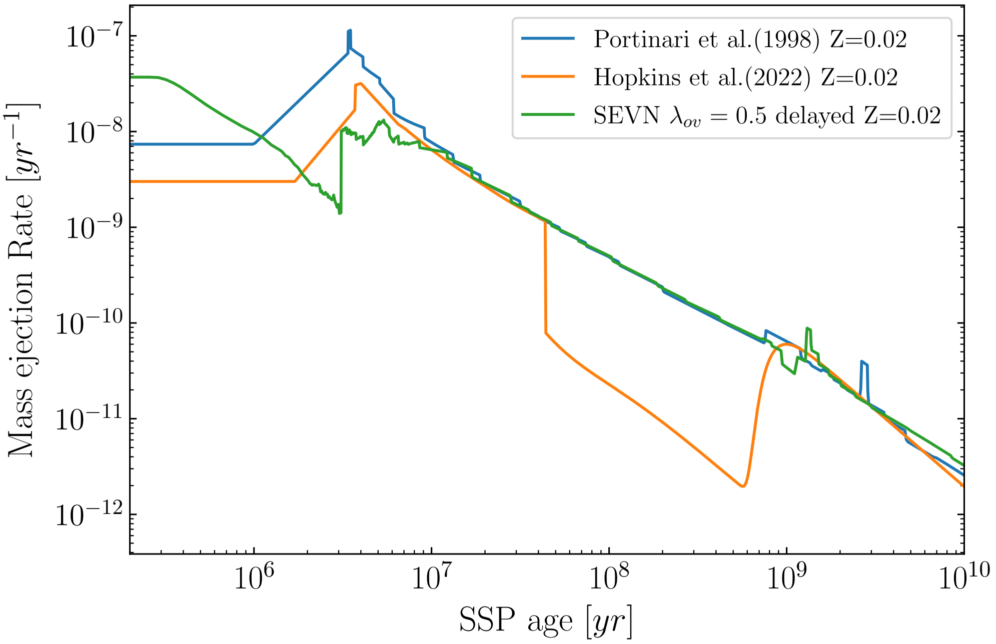


Figure 4.1: Mass ejection rate for the three models analyzed in this work at $Z = 0.02$. We sum all the contributions, so the lines are the combined effect of stellar winds, SNII, and AGB mass loss.

Fig. 4.1 shows very interesting results. For $t < 3 * 10^6$ yr the mass ejected by stellar winds is the only contribution to the mass ejection rate. As we discussed in section 2.3.1 the winds implemented in Marinacci et al. (2019) are taken from the parametrization given in Hopkins et al. (2018), which has a shape similar to the one in Hopkins et al. (2022). They are both constant during early times and then start to grow until SNII begins to explode. The difference between the two parameterizations is that the mass ejection rate by winds from Hopkins et al. (2018) is almost a factor of three larger than the value from the winds in Hopkins et al. (2022) at any given time. While for Hopkins et al. (2022) and Portinari et al. (1998) winds are constant and starts to increase, the trend for the new model is different. The new SEVN model has a mass ejection rate by winds that is almost an order of magnitude higher than the other two for the first 10^5 yr and then it starts to decrease. At $t = 10^6$ yr SEVN wind mass loss rate has the same value as the wind rate from FIRE-2 and then when SNII starts to explode it becomes two orders of magnitude smaller compared to the FIRE models. We expect a higher mass ejection for the new model at the very early evolutionary stages of the SSP. To fully characterize these features, we will compare the total mass ejected by the different models in section 4.3.

When SNII starts to explode the SEVN model has a lower mass ejection rate. This is due to the new efficiency function $f^S_{eff}(M, Z)$ (see subsec. 3.4.2 for a detailed discussion) that presents many new features and a different behavior with respect to the one used in Marinacci et al. (2019) (Fig. 2.14). In particular, for stars that have $M > 20\,\mathrm{M}_\odot$ the new model considers more sophisticated end-life mechanisms, like direct collapse, that make the shape of the mass ejection rate more complex. The new features of the model are evident in the ejection rate in Fig. 4.1, especially in the interval between $3 * 10^6$ yr $< t < 10^7$ yr. It is interesting to notice how in the SEVN model the new and finer sample of stellar masses makes the sudden variations in mass ejection rate (the typical ”step-like” appearance) smaller with respect to the old model obtained with a less refined sample. This justifies even more the choice of a new set of stellar masses. The Hopkins et al. (2022) curve tends to fall in between the new and old model.

All these functions become equivalent from $\sim 10^7$ yr and follow a power law-like behavior. However, a key difference between the interpolated model (both from Portinari et al. (1998) and SEVN) and the FIRE-3 emerges. In fact, after $t \sim 5 * 10^7$ yr, when all SNII have exploded, the rate in the FIRE-3 model drops more than an order of magnitude and stays at this level until $t \sim 10^9$ yr. This drop is unusual, and we could not find a suitable explanation for it. In Hopkins et al. (2022) it is stated that the bump at 10^9 yr is due to the onset of the AGB winds. We were not able to reproduce this behavior from FIRE-3 using both Portinari et al. and SEVN tables. After $t \sim 10^9$ yr the value of FIRE-3 starts again to resemble the other models. Notice how in both SMUGGLE models (namely the model from Portinari and the one obtained with SEVN) we have a bump in the mass ejection rate around $t \approx 10^9$ yr. As we discussed in the previous chapters, this is due to the lifetime function being flatter (see section 3.4.1). The new SEVN model gives a less prominent bump at earlier times.

We also analyze our results at low metallicity (we used $Z = 0.0004$, $Z = 0.02\ Z_\odot$ in terms of solar metallicity) to highlight the different behavior of less evolved stellar systems in the presented models. We report the ejection rate for $Z = 0.0004$ in Fig. 4.2.

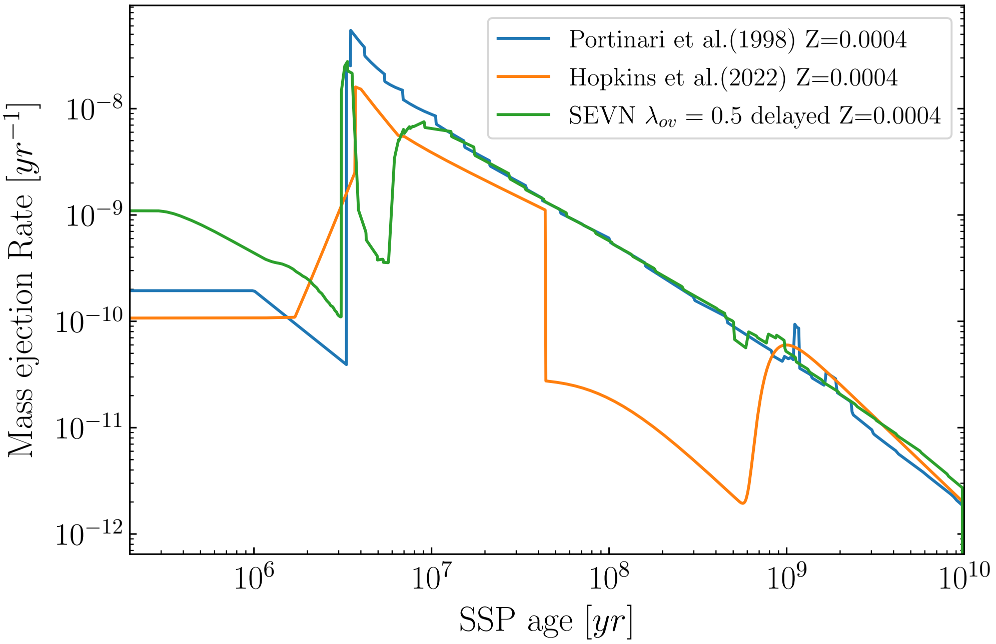


Figure 4.2: The same as Fig. 4.1, but for $Z = 0.0004$.

At lower metallicity, we generally observe a lower ejection rate compared to the solar metallicity case, especially in the part before SNII explosions stop ($t \approx 3 * 10^7$ yr). As mentioned for the $Z = 0.02$ figure, the first part of the plot ($t < 3 * 10^6$ yr) is also dominated by winds. In this case, the SEVN winds remain approximately an order of magnitude above the two other models. We notice how the shape of the winds from the SEVN model resembles much more the one of FIRE-2 but shifted to higher values. Instead, FIRE-3 shows a completely different behavior by staying below the other models and then starting to increase before SNII begins to explode. At times when the mass ejection is dominated by SNII explosions, we have the largest differences between models. The new SEVN model presents a fall in the value of the rate after the SNII starts to explode. As discussed in subsection 3.7, this is caused by the direct collapse of massive stars into black holes. This drop in the ejection rate is absent in both FIRE-3 and

Portinari models. After this dip, the SEVN model assumes the same value as the one based on Portinari. One difference we notice relative to the high metallicity case is that around $t \sim 2 * 10^9$ yr, the new model ejects more mass per unit time with respect to the Portinari one. This is because the new lifetime function from SEVN assumes smaller values for stars with $M < 2\,\mathrm{M}_\odot$, leading to a highr end-of-life rate. The bump for the Helium flash in Portinari is at later times and more prominent with respect to the new SEVN model also at this low metallicity case.

Comparing the ejection rates, we expect differences in how the new SEVN model impacts the galaxy formaiton simulations in which would be implemented. The mass ejection rate from winds show an higher value for the new SEVN model in the initial phase, this implies that more mass is returned in the very early times of the SSP's life. In time intervals where ejection is dominated by stars' end-life events, the new SEVN model predicts an equal or lower value with respect to the old model from Marinacci et al. (2019). In our implementation winds contain the same metal content as the SSP. The total mass ejected is affected by both the winds being substantially different between SEVN and the other two models and the end-life ejection being lower or equal, this does not allow us to clearly state if the new SEVN model will yield a larger or lower total mass return to the ISM with respect to the others. This aspect is analyzed in the next section.

4.3 Total ejected mass fraction

In this section, we compare the total ejected mass fraction across the different models examined in this work. At any timestep of its evolution, every stellar particle expels a quantity of gas given by eq. (2.8), where Δt is the timestep considered. We can recover ΔM by simply taking the cumulative mass ejection fraction function $M_{tot} = M_{tot}(t, Z)$ (see Fig. 2.18 and Fig. 3.13 as examples) and calculate it for t and $t + \Delta t$. In this way $\Delta M = M_{tot}(t + \Delta t) - M_{tot}(t)$. This means that comparing directly the cumulative functions of different models allows us to directly evaluate the mass that a stellar particle is returning to the ISM at any given SSP age. We report in Fig. 4.3 the mass ejection fraction for the different models analyzed here at solar metallicity.

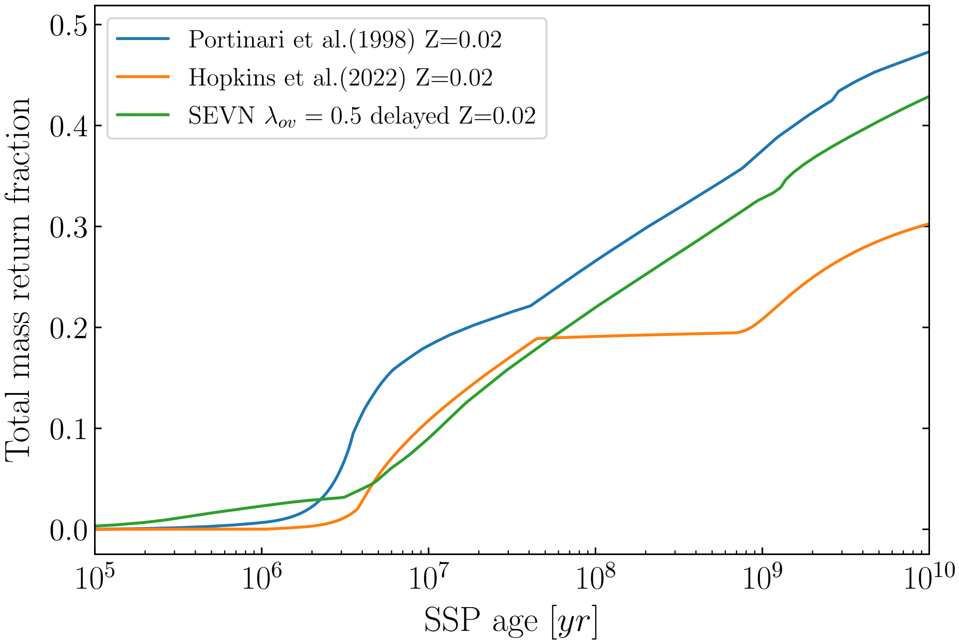


Figure 4.3: The cumulative ejected mass fraction for the three models presented in this work is shown for $Z = 0.02$. We sum all the contributions so the lines encapsulate stellar winds, SNII, and AGB losses.

In Fig. 4.3 we can better quantify the differences between models already discussed in section 4.2. Starting from early times ($t < 10^6$ yr) stellar winds dominate the mass ejection. As expected, the new SEVN model gives an earlier and larger contribution to the mass ejection at this evolutionary stage; however, wind contribution runs out earlier than in the other models considered in this analysis. This is evident in the plateaus displayed by the green curve before SNII starts to explode at $t \approx 2.5 * 10^6$ yr. In Fig. 3.17 we see that the total contribution of winds from SEVN is around 3% of the total mass of the SSP. FIRE-2 O/B winds (Fig. 2.18) arrive instead at 15%, while in FIRE-3 is 8% (Fig. 2.7) of the total mass ejected. This is probably due to our approximation in which the mass expelled in the post main-sequence (i.e. *after the wind phase*) is released during the SNII explosion, making the wind less important in the total mass budget.

SNII starts to explode roughly at the same time for all the models; once this happens both FIRE-3 and Portinari surpass the SEVN model. This is also due to the fact that, differently from the new SEVN model, in which winds start to exhaust before the start of the SNII explosion, in both Portinari and Fire-3 prescriptions winds still inject a relevant quantity of mass during the explosions of SNII, making the ejection mass fraction have a rapid increase when SNII start to explode. Remarkably, the Fire-3 model has a rapid change in the slope after SNII stops to explode, this corresponds to the sudden decrease in the ejection rate that we observe in the orange curve in Fig. 4.1. At the time in which SNII stops to explode ($t \approx 3 * 10^7$ yr), the total mass ejected from the SEVN model and the FIRE-3 model is almost the same. We notice that while for Hopkins et al. (2022) the mass ejected from winds and the mass ejected from SNII explosions are equal (11% for SNII explosions, 9% for stellar winds), for the SEVN model SNII carry much more mass (3% winds, 17% SNII). This is probably another effect of our assumption in constructing the model all the mass in the post main-sequence is expelled in the explosion. This is confirmed also by the average ejecta mass of an SNII event, in Hopkins et al. (2022) they give an average ejecta mass of $M_{SNII}^{avg} \sim 8.72\ \mathrm{M}_{\odot}$ while for SEVN this value is $M_{SNII}^{avg} \sim 12\ \mathrm{M}_{\odot}$. After all SNII have exploded, the new SEVN model and the Portinari model have approximately the same slope. This is expected since the mass ejection rates in Fig. 4.1 are the same after $t \sim 10^7$ yr. Notice how the bump in the rate in both Portinari et al. (1998) and SEVN model makes the cumulative function have a sudden increase.

The old model from Portinari returns roughly 47% of the mass to the ISM at 10^{10} yr. The results of this work show that the new SEVN model predicts at the same time a returned mass fraction of 42%, reducing the total mass ejected by 5% of the total. While this might not seem a relevant change, notice that this difference comes mostly from the different mass returns of SNII. Most of the gas that can be reused after being processed inside a star comes from the first 10^8 yr of the SSP. At $t = 10^8$ yr the cumulative mass returned by the old model is 25% while it is 20% for the SEVN model. This means that the old model has returned almost the 25% more gas with respect to the new one at $t = 10^8$ yr. This gas is available for forming a new star at that time. Therefore, taking into account all the aspects discussed, we expect a (likely mild) reduction in SFR if we adopt this new model inside hydrodynamical simulations that model galaxies.

Even if the SEVN model does predict less mass returned with respect to the old SMUGGLE implementation, the FIRE-3 model returns 10% less mass with respect to SEVN at $t = 10^{10}$ yr, mostly because form $3 * 10^7$ to 10^9 yr the SSP does not lose any mass. This implies that almost all the mass of the stars ending their life from $3 * 10^7$ yr $< t < 10^9$ yr would be stored in the remnant.

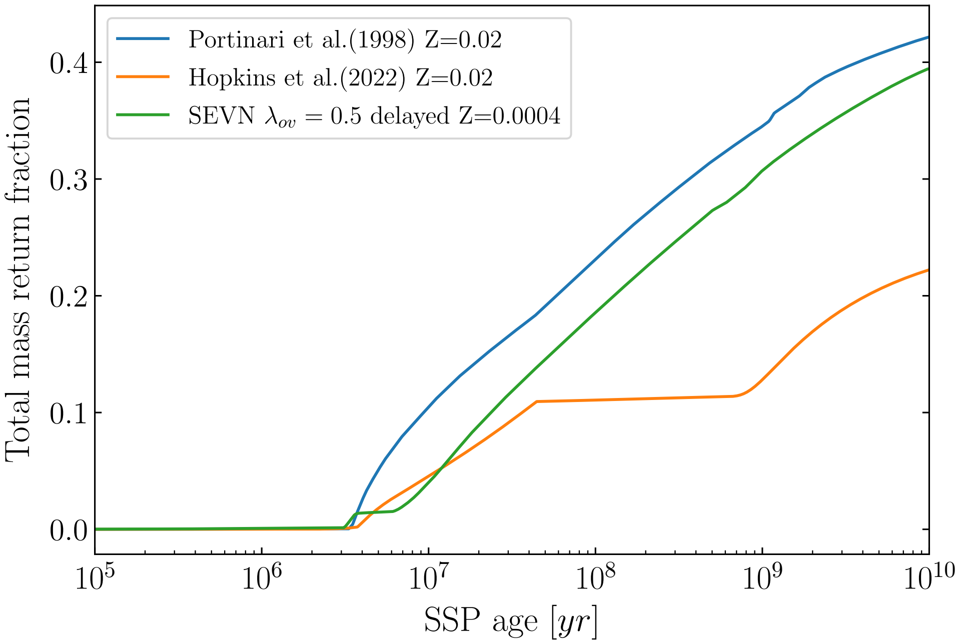


Figure 4.4: The same as in Fig. 4.3, but for $Z = 0.0004$.

We also report the same graph at low metallicity ($Z = 0.0004$) in Fig. 4.4, lower metallicity systems correspond roughly to SSP that formed earlier in the evolution, by comparing the different predictions of the models we can understand how the mass is re-injected in the early stages. Comparing Fig. 4.3 with Fig. 4.4 we notice a decrease in the total mass returned at $t = 10^{10}$ yr at lower metallicity for all the models. In the first part of the plot ($t < 3 * 10^6$ yr), only winds are present, the cumulative mass ejected by these winds is negligible with respect to the mass ejected by SNII. In both Portinari and FIRE-3 models, winds start to inject a relevant quantity of mass when SNII starts to explode (Fig. 2.7). In the Portinari model winds make up 5% of the total mass of the SSP, while in FIRE-3 they only account for 1%. In our SEVN model winds at low metallicity always give a negligible contribution to the mass loss (Fig. 3.17). The dip in the ejection rate for the new SEVN model at $4 * 10^6 - 7 * 10^6$ yr in Fig. 4.2 alters the cumulative function and makes the green curve approximately constant in the interval $4 * 10^6 - 7 * 10^6$ yr. This approximately flat trend in the ejection is the reason why also in the low metallicity case the new model predicts less total mass returned at $t = 10^{10}$ yr compared to the old SMUGGLE model. The difference between the cumulative function of the two models at $t = 10^{10}$ yr is 3% of the total mass ejected. We highlight that after the SNII explosions, the difference between the cumulative function of the two models remains similar to the one that we reported at high metallicity, the discrepancy between

the two starts to shrink for $t > 2 * 10^9$ yr, where the ejection rate of the new model is higher than the old one (Fig. 4.2). This is important since, as we discussed for the high metallicity case, most of the reusable mass for SF is ejected in the first 10^8 yr. The discrepancy of the new model with with FIRE-3 is in this case even bigger respect to the $Z = 0.02$ case, reaching almost the 16% of the total mass. As mentioned in the previous section, we were not able to find a satisfactory explanation for this discrepancy.
We want to emphasize that the SEVN model predicts less mass ejection relative to the Portinari one at all metallicities. Is possible to argue that more updated model follow this trend, since both SEVN and Hopkins et al. (2022) show less mass ejected respect to older model like (Portinari et al., 1998). In the previous chapter we highlighted how the difference between the two models emerges from the different remnant mass estimations after SNII explosions.

4.4 Elemental yields

Elemental yields from stellar evolution are an essential ingredient in galaxy formation simulations. Nuclear fusion alters the chemical composition of the gas, changing its metal content, leading to enriched gas injected in the ISM, that will be used to form stars with higher metal content. We expect different stellar evolution models to produce different quantities of elements during the evolution of an SSP. In this section, we analyze how the new SEVN model compares with FIRE-3 for each element ejected during the SSP lifetime. We emphasize that we will show plots that do not consider the ejection of chemical elements from SNIa. For elements like iron, these contributions are not negligible. Still, the goal of this discussion is to show how the new SEVN stellar evolution model compares to FIRE-3 in the synthesis of new elements. For this reason, we confront only the metal production of stars, excluding SNIa, that we expect to have similar contributions.
In the previous chapter, we did not cover elemental yields from SEVN. This is because SEVN does not currently feature elemental yields as a publicly available output. However, elemental yields are necessary if we aim to build new stellar evolutionary tables to be employed in the future generation of galaxy formation simulations. We circumvent this issue by using the yields from Portinari et al. (1998) re-normalized to the new values of total mass ejected in the SEVN model. We define the new yields as:

$$y_i^S(m, Z) = y_i(m, Z)\frac{M_{tot}^S}{M_{tot}} \tag{4.1}$$

where y_i^S is the new yield for element i, M_{tot}^S is the cumulative mass released by the SEVN model at $t = 10^{10}$ yr and M_{tot} is the one released in the Portinari et al. (1998) model.

We want to compare the new SEVN model with Fire-3 (Hopkins et al., 2022) to analyze potential discrepancies in the element production of the two models. To do this we calculate, for both models, the cumulative mass ejection function of the traced element in the tables. We report the results in Fig. 4.5.

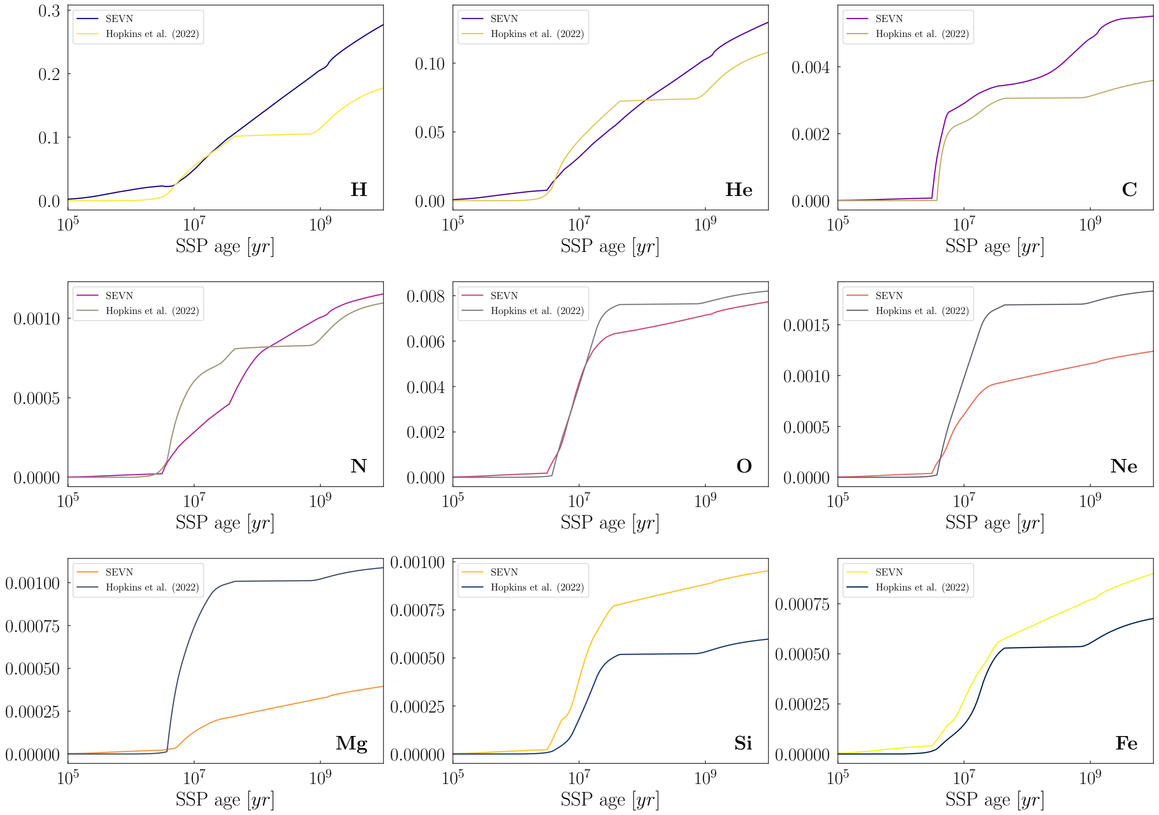


Figure 4.5: Mass ejection fraction for all common chemical elements between SMUGGLE and FIRE-3. The figure compares the cumulative ejection fractions of elements from the new model built with SEVN and the FIRE-3 model, as indicated in the legends of each panel. The lines are the sum of the elements present in stellar winds, SNII explosions, and AGB.

In Fig. 4.5 we can notice various features that distinguish the two models. Hydrogen and Helium follow a similar behavior to the total cumulative function of both models. Other elements like **C**, **Ne** and **Si** follow a very different trend compared to the respective cumulative function, but at $t = 10^{10}$ yr the ratio between the yields of the two models is similar to the ratio of their total cumulative ejected mass. Oxygen and Nitrogen assume very similar values between FIRE-3 and the SEVN model, whereas in FIRE-3 the production of Neon and Magnesium is much more prominent. We need to consider that the total mass returned in FIRE-3 is 10% less than in the new model, in this way, the discrepancy between the element production becomes even more relevant, with the new model underpredicting the percentage of new elements formed, especially Mg, N, O, compared to FIRE-3. These considerations suggest that more self-consistent modeling of elemental yields is needed in the future if we want to create a more accurate model for the chemical enrichment of galaxy formation simulations in the SEVN framework.

4.5 Metal yield

SMUGGLE calculates the cooling rate of the gas by using its total metal fraction (overall metallicity) of the gas. It is thus instructive to evaluate the total metal mass directly by defining a yield for all the metals production happening over the evolution of a SSP. We also want to compare this quantity with the FIRE-3 total metal production. For FIRE-3 we sum all the cumulative functions of elements that are not Helium nor Hydrogen. For the new SEVN model we can use the total metal yield (i.e., the total metal production) of the SSP. We plot the cumulative functions of metals ejected by FIRE-3, (Portinari et al., 1998), and SEVN models in Fig. 4.6.

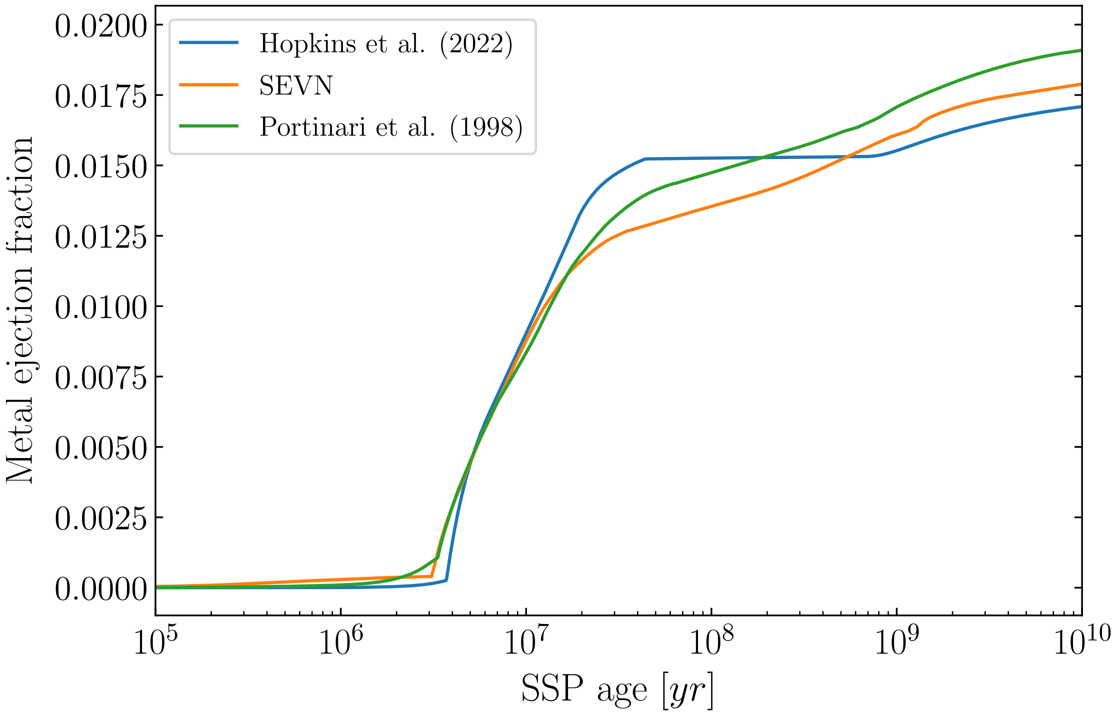


Figure 4.6: Metal ejection fraction of the three models described in this work, as indicated in the legend. The figure compares the cumulative metal ejection fractions calculated at Z=0.02

In Fig. 4.6 some interesting features are noticeable. As expected based on the global mass return (see sec. 4.3), the total mass of metals returned for the model by Portinari is slightly larger than the SEVN one. It is interesting to see how the total metal injected by the FIRE-3 is very similar to the new SEVN model. Even if both (Portinari et al., 1998) and SEVN model predict a higher metal production at $t = 10^{10}$ yr, we immediately notice that the ratio between the total mass in metal and the total mass ejected is much higher in FIRE-3. This might suggest that the current model is underproducing metals relative to FIRE-3. We will test this hypothesis further when elemental yields will be available as output for SEVN. This will be crucial since metals shape the chemical evolution of the ISM, are used as tracers in many cosmological works and also shape the dynamical properties of the gas by changing the rate at which it can cool.

5

Summary and conclusions

In this thesis, we created a new stellar evolution model to be employed in state-of-the-art hydrodynamical simulations of galaxy formation and evolution that make use of the SMUGGLE ISM and stellar feedback framework for the AREPO code.We analyzed its key properties and compared the results with two models from the literature: the original SMUGGLE implementation and the stellar evolution model used in FIRE-3.

We started this work by reproducing the implementation of stellar evolution in FIRE-3 model (Hopkins et al., 2022). In this model, stellar evolution is parameterized via analyitc fitting functions derived from the SSP evolution codes STARBURSTS99 (Leitherer et al., 1999) with updated stellar evolution tracks. We chose the FIRE-3 model as a benchmark for mainly two reasons: it represents one of the most advanced models used in galaxy formation simulations and it employs a standard approach for stellar evolution that is to parameterize it by supplying fits to relevant quantities (lifetimes, stellar mass loss, SN rates, etc...) as a function of time.

We then focused on reproducing all the steps implemented in the SMUGGLE model (Marinacci et al., 2019) to include stellar evolution. We choose SMUGGLE since the code infrastructure for stellar evolution is similar to many other galaxy formation and evolution models (such as Vogelsberger et al. 2013; Pillepich et al. 2018), making the analysis carried in this thesis applicable to many other works in the field. Also, the stellar evolution implementation in SMUGGLE utilizes data from a .hdf5 file, ensuring high portability and ease of code improvement. We found it necessary to update the stellar evolution model currently implemented in SMUGGLE since it is based on the work of Portinari et al. (1998), and also given the differences found between SMUGGLE and the more recent FIRE-3 implementation. The analysis carried out in this work first aimed at having a standalone Python code that performs the same steps as SMUGGLE for stellar evolution. The first step is reading relevant data from the same hdf5 tables. These tables contain an array of masses and metallicity, parameters associated with a single star. For each data entry, the tables give: lifetime, mass ejected at the end of life (considering all the post-main-sequence phases), the overall metal yield, and the elemental yield. By linearly interpolating between these points we derived the functions describing the lifetime, the mass ejected and the yield given a generic mass and metallicity. We then integrated these functions and obtained the rate at which the stars leave the main sequence, the mass ejection rate, and the total mass, total metal mass, and the mass of each tracked element ejected by an SSP. We then proceeded to improve the model by incorporating more recent stellar evolutionary tracks and a self-consistent implementation of stellar winds. The main goal was to create a new set of hdf5 tables that could be used by SMUGGLE and other galaxy formation physics frameworks that share the same software infrastructure for stellar evolution (for instance the Illustris and IllustrisTNG models). To accomplish this we run simulations of single stars using the software SEVN (Iorio et al., 2023). We perform several simulations with SEVN, exploring its parameter space, and changing the formalism for SN explosions, spin, and the stellar tracks available. From these results we constructed several sets of hdf5 tables, that we subsequently analyzed with the same code used for the original SMUGGLE model to get the mass ejection rate, and the total mass returned. We also construct new tables with a self-consistent implementation for the stellar winds of O/B stars based on the outcome of the SEVN simulations. We then compared the three stellar evolution models and discussed their differences.

We firstly noted how the original SMUGGLE model based on Portinari et al. (1998) with stellar winds from Hopkins et al. (2018) had a mass returned at 10^{10} yr equal to almost 50% of the total initial mass of the SSP, while the model of Hopkins et al. (2022) returned just the 30%. The new SEVN model returns 42% of the total mass, confirming that our new model will return less mass. This is important since we expect that with less mass available the SFR of a galaxy simulated with this new version of the SMUGGLE model will decrease. We notice how the new model ejects mass in the ISM at a lower rate, especially in the first 10^7 yr, as it is evident by comparing the mass ejection rates (Fig. 4.1). One of the main features of the new model is that it takes into account the direct collapse for star's end life, leading to less mass released . It is worth noting that the model presented in Hopkins et al. (2022) stops losing mass after the end of SN explosion phase of the SSP, and then the AGB winds start again to inject mass in the ISM at 10^9 yr. We were not able to reproduce this behavior and defer an explanation of this difference to future work.

In this thesis, we also identified and characterized features of the stellar evolution model implemented in SMUGGLE. We showed how the interpolation procedure creates unwanted discontinuities in the rate function, but, because of the integration procedure involved, these jumps are smoothed in the function describing the total mass return. We also noticed that the number of stellar masses used to sample the SSP in the old tables was too small to accurately describe its evolution, eventually leading to those pronounced "jumps" in the rate functions. We then decided to enhance the number of stars on the sample to improve its granularity and get a better description of an SSP in the new model. Moreover, the variety of running options in SEVN allows us to study how changing the running parameters alters the total mass returned by the SSP. We discovered that no difference between the two available tracks with different overshooting is found, suggesting this parameter might not have a large impact on this quantity. We also investigated how different supernova formalisms may affect the mass return. We found a discrepancy of maximum $\sim 2\%$ between models of explosion. Based on these results, we decided to adopt the PARSEC tracks with $\lambda_{ov} = 0.5$ tracks with the delayed formalism as a fiducial model.

We also carried out a comparison between the metal ejection of both FIRE-3 and the old SMUGGLE model. We found that the ratio between metals and total mass is systematically lower in SMUGGLE (4% of the mass ejected is in metals at 10^{10} yr) with respect to Hopkins et al. (2022) (5.5% of the mass ejected is in metals at 10^{10} yr), especially in the production of some metals (such as Mg). To build a table usable by SMUGGLE we need to add the single metal yield. We then renormalized the mass yields of the old model to the new mass ejected for the SEVN model. This leads to less total metal production since the metal-to-mass ratio is defined to be the same as in SMUGGLE.

Given the fact that our new stellar evolution tables predict less mass returned by the SSP, we expect that the SFR of a galaxy simulated with the new model to be reduced with respect to the original one, with a lower overall ISM metal pollution. To unambiguously confirm this hypothesis we would need to run galaxy simulations with our new stellar evolution model. In this way, we would be able to directly test on a galaxy scale the impact of this new implementation of stellar evolution on the SFR, the metal pollution, and other properties such as the multiphase distribution of the ISM, and potential modification of simulated scaling relations. This is a path forward that we intend to explore in the future.

Is also possible to use the codes developed in this work to analyze the results of simulations that implement stellar evolution in the same way as SMUGGLE, but do not take in account metal production. We could use the code to trace back the production of metals and quantify it.

As for the current public release of SEVN, no yields for metals or individual elements are given in the output. When the yields for single elements become available, we will implement the new metal production in the tables, creating a completely self-consistent model for stellar evolution. With the code developed in this work, this would be a straightforward task to accomplish as soon as the simulations with SEVN are performed. Using the complete model with SEVN yields will allow us to carry out a more detailed comparison between simulations with different implemented stellar evolution models.

One key advantage of SEVN is the possibility of adding new sets of stellar tracks to the software to perform new simulations. This feature will allow us to use the code developed in this work to update stellar evolution tables with the most up-to-date stellar evolution tracks very effectively. For this purpose, it would be useful to develop in the future an automatic pipeline that takes as input the array of stellar mass, metallicity, stellar tracks, SNII formalism, etc., and automatically creates new tables ready to use. This will allow us to have a continuously updated implementation of stellar evolution, reducing the time and effort involved in the creation of new tables.

It would also be interesting to adapt the tables for larger simulation efforts, such as IllustrisTNG, to check the impact of this new model on the evolution of the gas (and galaxies) on cosmological scales.

Finally, SEVN is also suited for simulating binary-star systems. Therefore, we could use SEVN to take into account this aspect in our stellar evolution model. We could do this by constructing tables that encode binary evolution in way similar to what it was used for single stars, an array of stellar mass pairs at a given metallicity. This is not straightforward and many assumptions will need to be made, such as the distribution of the distance between stars, the distribution of masses in the binary system, etc. In this way no parameterization for SNIa is in principle needed since their rate of the explosion would be encoded in the tables. This would be true also for BHs mergers, which generate GW that could become another outcome self-consistently derived from the galaxy formation simulations to be compared with the available observational constraints.

In summary, in this work, we created, analyzed, and compared stellar evolution models for hydrodynamic simulation of galaxy formation and evolution. We also laid the foundation for a pipeline that will allow us to create and update such models very effectively and with little effort. We also expect this type of work to be expanded to include binary star evolution to continue improving how stellar physics is taken into account in N-body hydrodynamic simulations of galaxies.

Bibliography

A. Cimatti, F. Fraternali, C. Nipoti 2019, Introduction to Galaxy Formation and Evolution: From Primordial Gas to Present-Day Galaxies. Cambridge University Press

Adams, Robert A. Fournier, John 1975, Sobolev Spaces. Pure and Applied Mathematics, 2nd ed, 2003 edn. Vol. Vol. 140, Academic Press, Boston, MA

Agertz O., Teyssier R., Moore B., 2011, Monthly Notices of the Royal Astronomical Society, 410, 1391

Amard L., Matt S. P., 2020, The Astrophysical Journal, 889, 108

Balogh M. L., Baldry I. K., Nichol R., Miller C., Bower R., Glazebrook K., 2004, The Astrophysical Journal, 615, L101

Barnes J., Hut P., 1986, Nature, 324, 446

Berger M., Colella P., 1989, Journal of Computational Physics, 82, 64

Bode P., Ostriker J. P., 2003, The Astrophysical Journal Supplement Series, 145, 1

Bressan A., Marigo P., Girardi L., Salasnich B., Dal Cero C., Rubele S., Nanni A., 2012, Monthly Notices of the Royal Astronomical Society, 427, 127

Brighenti 2021, Computational Astrophysics and Statistics slides

Burrows A., Vartanyan D., 2021, Nature, 589, 29

Chabrier G., 2001, The Astrophysical Journal, 554, 1274

Chevance M., Krumholz M. R., McLeod A. F., Ostriker E. C., Rosolowsky E. W., Sternberg A., 2022

Choi J., Dotter A., Conroy C., Cantiello M., Paxton B., Johnson B. D., 2016, The Astrophysical Journal, 823, 102

DESI Collaboration Adame A. G., et al., 2024, DESI 2024 VI: Cosmological Constraints from the Measurements of Baryon Acoustic Oscillations, http://arxiv.org/abs/2404.03002

De Vaucouleurs 1959, Springer Berlin Heidelberg, pp 275–310

Donea J., Huerta A., Ponthot J., Rodríguez-Ferran A., 2004, in Stein E., Borst R., Hughes T. J. R., eds, , Encyclopedia of Computational Mechanics, 1 edn, Wiley, doi:10.1002/0470091355.ecm009, https://onlinelibrary.wiley.com/doi/10.1002/0470091355.ecm009

Eleuterio F. Toro 2009, Riemann Solvers and Numerical Methods for Fluid Dynamics. Springer Berlin, Heidelberg

Faucher-Giguère C.-A., Lidz A., Zaldarriaga M., Hernquist L., 2009, The Astrophysical Journal, 703, 1416

Ferland G., Korista K., Verner D., Ferguson J., Kingdon J., Verner E., 1998, Publications of the Astronomical Society of the Pacific, 110, 761

Fraternali F., 2017, pp 323–353, doi:10.1007/978-3-319-52512-9˙14, http://arxiv.org/abs/1612.00477

Fryer C. L., Belczynski K., Wiktorowicz G., Dominik M., Kalogera V., Holz D. E., 2012, The Astrophysical Journal, 749, 91

Grand R. J. J., et al., 2017, Monthly Notices of the Royal Astronomical Society, p. stx071

Groh J. H., et al., 2019, Astronomy & Astrophysics, 627, A24

Guo F., Oh S. P., 2008, Monthly Notices of the Royal Astronomical Society, 384, 251

Hopkins P. F., et al., 2018, Monthly Notices of the Royal Astronomical Society, 480, 800

Hopkins P. F., et al., 2022, Monthly Notices of the Royal Astronomical Society, 519, 3154

Hubble 1925, Vol. 48, p. 139

Høyrup J., 2002, Historia Mathematica, 29, 193

Ikeuchi S., Ostriker J. P., 1986, The Astrophysical Journal, 301, 522

Iorio G., et al., 2023, Monthly Notices of the Royal Astronomical Society, 524, 426

Jørgensen I., Chiboucas K., Flint K., Bergmann M., Barr J., Davies R., 2006, 639

Katz N., Weinberg D. H., Hernquist L., 1996, The Astrophysical Journal Supplement Series, 105, 19

Kroupa P., 2001, Monthly Notices of the Royal Astronomical Society, 322, 231

Krumholz M. R., Tan J. C., 2007, The Astrophysical Journal, 654, 304

Lancaster L., Ostriker E. C., Kim J.-G., Kim C.-G., 2021, The Astrophysical Journal Letters, 922, L3

Leitherer C., et al., 1999, The Astrophysical Journal Supplement Series, 123, 3

Li M., Bryan G. L., Ostriker J. P., 2017, The Astrophysical Journal, 841, 101

Mannucci F., Della Valle M., Panagia N., 2006, Monthly Notices of the Royal Astronomical Society, 370, 773

Maoz D., Graur O., 2017, The Astrophysical Journal, 848, 25

Mapelli M., Spera M., Montanari E., Limongi M., Chieffi A., Giacobbo N., Bressan A., Bouffanais Y., 2020, The Astrophysical Journal, 888, 76

Marconi A., Hunt L. K., 2003, The Astrophysical Journal, 589, L21

Marinacci F., Sales L. V., Vogelsberger M., Torrey P., Springel V., 2019, Monthly Notices of the Royal Astronomical Society, 489, 4233

Martizzi D., Fielding D., Faucher-Giguère C.-A., Quataert E., 2016, Monthly Notices of the Royal Astronomical Society, 459, 2311

Nguyen C. T., et al., 2022, Astronomy & Astrophysics, 665, A126

P. Coles, F. Lucchin 2002, Cosmology, 2nd edition edn. Wiley

Pakmor R., Springel V., Bauer A., Mocz P., Munoz D. J., Ohlmann S. T., Schaal K., Zhu C., 2016, Monthly Notices of the Royal Astronomical Society, 455, 1134

Paxton B., et al., 2013, The Astrophysical Journal Supplement Series, 208, 4

Paxton B., et al., 2019, The Astrophysical Journal Supplement Series, 243, 10

Pillepich A., et al., 2018, Monthly Notices of the Royal Astronomical Society, 473, 4077

Planck Collaboration et al., 2020, Astronomy & Astrophysics, 641, A6

Popesso P., et al., 2022, Monthly Notices of the Royal Astronomical Society, 519, 1526

Portinari L., Chiosi C., Bressan A., 1998

R.W Hockney, J.W Eastwood 1988, Computer Simulation Using Particles, 1st edn. Boca Raton, `https://doi.org/10.1201/9780367806934`

Rahmati A., Pawlik A. H., Raičevic M., Schaye J., 2013, Monthly Notices of the Royal Astronomical Society, 430, 2427

Riess A. G., et al., 1998, The Astronomical Journal, 116, 1009

Spera M., Mapelli M., Bressan A., 2015, Monthly Notices of the Royal Astronomical Society, 451, 4086

Spera M., Mapelli M., Giacobbo N., Trani A. A., Bressan A., Costa G., 2019, Monthly Notices of the Royal Astronomical Society, 485, 889

Spergel D. N., et al., 2003, The Astrophysical Journal Supplement Series, 148, 175

Springel V., 2010a, Annual Review of Astronomy and Astrophysics, 48, 391

Springel V., 2010b, Monthly Notices of the Royal Astronomical Society, 401, 791

Thomas D., Maraston C., Schawinski K., Sarzi M., Silk J., 2010, Monthly Notices of the Royal Astronomical Society

Tully, R. B. Fisher, J. R. 1977, Astronomy and Astrophysics, Vol. 54, p. 661

Vink J. S., De Koter A., Lamers H. J. G. L. M., 2001, Astronomy & Astrophysics, 369, 574

Vogelsberger M., Genel S., Sijacki D., Torrey P., Springel V., Hernquist L., 2013, Monthly Notices of the Royal Astronomical Society, 436, 3031

Vogelsberger M., et al., 2014, Monthly Notices of the Royal Astronomical Society, 444, 1518

Von Neumann 1945

Weinberger R., Springel V., Pakmor R., 2020, The Astrophysical Journal Supplement Series, 248, 32

Wolfire M. G., McKee C. F., Hollenbach D., Tielens A. G. G. M., 2003, The Astrophysical Journal, 587, 278

Woosley S., Sukhbold T., Janka H.-T., 2020, The Astrophysical Journal, 896, 56

Zevin M., Spera M., Berry C. P. L., Kalogera V., 2020, The Astrophysical Journal Letters, 899, L1

van Leer Bram 1984,] 10.1137/0905001, 5

Appendixes

A .hdf5 files

The Hierarchical Data Format version 5 (HDF5), is an open source file format that supports large, complex, heterogeneous data. HDF5 uses a ”file directory” like structure that allows users to organize data within the file in many different structured ways (see https://www.neonscience.org/ for more details). Opeining a hdf5 file we found datasets, lists of data readable by the user, and groups, a list of datasets and other groups. The tables readble by SMUGGLE and used to reconstruct the stellar evolution model are present in this way

- Masses **Dataset** {number_of_masses}
- Metallicities **Dataset** {number_of_metallicities}
- Number of masses **Dataset** {SCALAR}
- Number of metallicities **Dataset** {SCALAR}
- Number of species **Dataset** {SCALAR}
- Reference **Dataset** {STRING}
- Species names **Dataset** {number_of_species}
- Yield names **Dataset** {number_of_metallicities}
- Yields **Group**

Inside the **Group** Yields we find, for each metallicity, a collection of datasets. These datasets are:

- Ejected_mass **Dataset** {number_of_masses}

- Total_Metals **Dataset** {number_of_masses}
- Yield **Dataset** {number_of_masses, number_of_species}

where ejected mass is the mass ejected in the post main sequence phase by a star, the total metals is the total metal yield as defined in eq. 2.3.1, and the yields contain the element variation for each species tracked by the model.

B Interpolation of the lifetime function

The SNII rate obtained using the interpolation technique in SMUGGLE leads to a oscillating function. This shape is rather unlikely to occur in reality, if not unphysical. In this work we propose a new way to address the problem and we show that the main source of this oscillating behavior is the linear interpolation used in SMUGGLE to determine stellar lifetimes. We propose a proof of concept way to eliminate this problem. The idea is to fit the lifetime function with a polynomial of a higher degree. This method can be pretty straightforward to implement and adopt. In fact, we could simple create a new lifetime table where, instead of passing for each sampled mass mass a lifetime, we pass n coefficients of the n-degree polynomial. In this way the new lifetime function becomes:

$$\tau(M) = \sum_{i=0}^{n} a_i M^i \tag{B.0}$$

where a_i is the coefficent of the ith grade coefficient of the polynomial. These coefficents are obtained by using the python function *polyfit* that minimize the the χ^2 by fitting in the data a n-grade polynomial. In Fig. B.1 we show how the results would look like for the fitted lifetime function and the SNII rate of explosion.

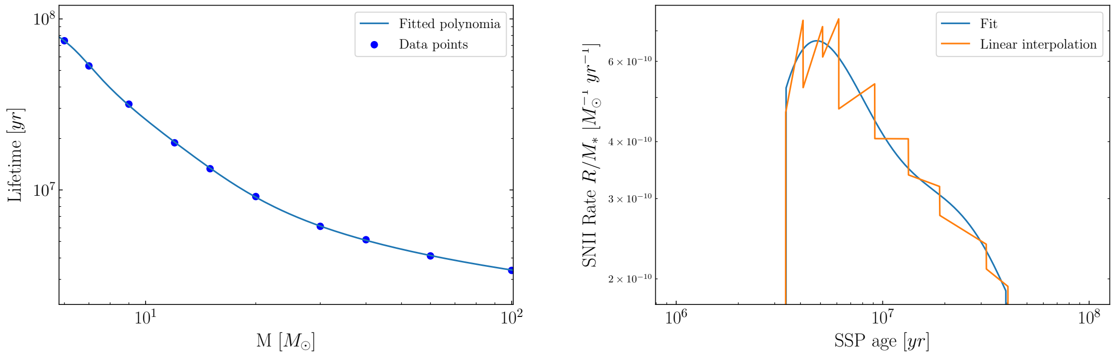


Figure B.1: On the left, the lifetime function for massive stars ($M > 8M_\odot$) obtained by fitting a 10-degree polynomial at solar metallicity. To avoid overfitting, we used a 10-degree polynomial since the number of data points is 11. On the right, we have the comparison between the SNII rate of explosion obtained from the interpolated lifetime function (orange) and the fitted lifetime function (blue).

We notice how the SNII rate is completely smooth. This comes at the cost of making the fitting function not passing precisely by the data points. In order to have a reliable model with this method we would need to quantify the error on the lifetime function, this can be done by comparing the fitted lifetime function with a lifetime function obtained with a very fine sampling of stellar masses. In this way we can quantify the error of both interpolation and fitting lifetime function. We leave this possibility open for future work.

C New DeathMatrix implementation in SEVN

The new Deathmatrix formalism for SN explosions is implemented in SEVN using the results of the table found in Woosley et al. (2020). This table contains the mass of the remnant as a function of the initial mass of the star. SEVN interpolates between points of the Woosley et al. (2020) table in order to find the remnant mass depending as a function of the initial stellar mass. One limitation of this implementation is the fact that pure helium stars are simulated. This implies that the remnant mass is derived by SEVN using the mass of the He nucleus, basically expelling all the hydrogen in the outer layer. Our slight modification to the code makes the hydrogen layer fall inside the remnant if more then 99% of the mass of the helium nucleus is converted in remnant mass. This in fact correspond to the direct collapse of the star into a black hole, so that also the hydrogen outer layer is incorporated in the remnant. In Fig. C.2 we show how this change affects the efficiency function.

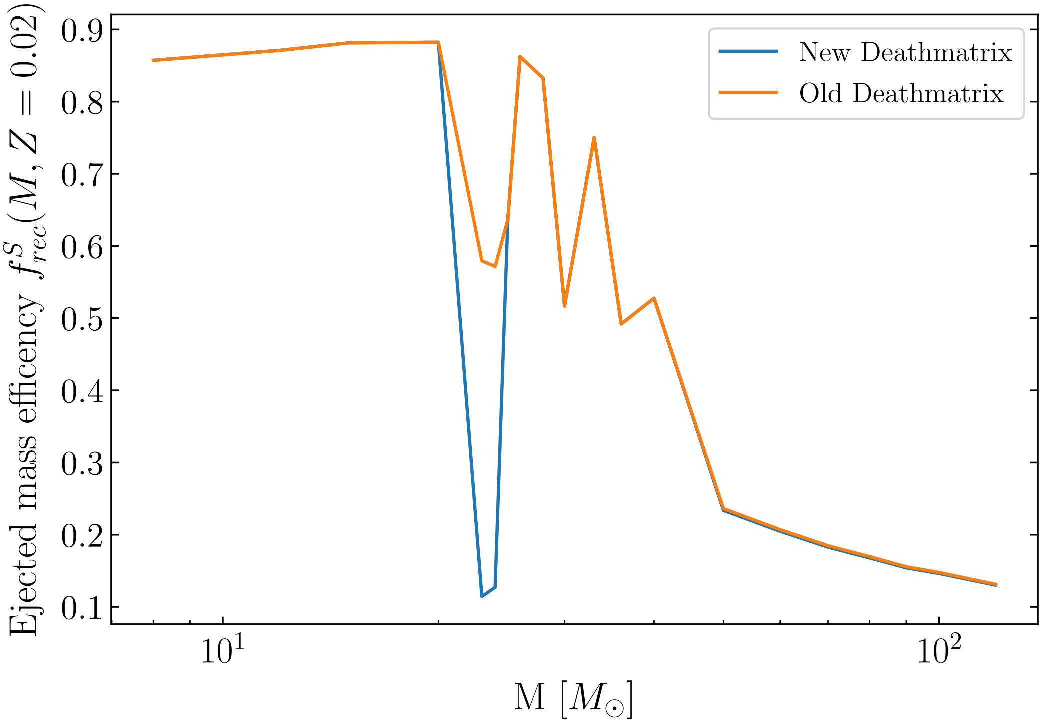


Figure C.2: Comparison between the old (orange) and new (blue) Deathmatrix efficiency functions. Notice how the efficiency function for the new implementation drops when we have a direct collapse ($20M_\odot < M < 30M_\odot$), in fact the efficiency in this rate drops, since all the mass of the star falls in the central black hole. For high stellar masses ($M > 30M_\odot$), the mass of the hydrogen is basically negligible, so the efficiency functions is essentially unchanged.

Ringraziamenti

È consuetudine nei ringraziamenti delle tesi menzionare il relatore per primo. Sebbene io faccia esattamente questo, desidero sottolineare che il mio ringraziamento al professor Marinacci va ben oltre il semplice dovere istituzionale. Durante questo percorso, mi ha guidato, consigliato e supportato (e sopportato) con un impegno davvero eccezionale. Il successo di questo lavoro è in gran parte merito della straordinaria persona e scienziato che è. Ti auguro sinceramente una carriera ricca di altri successi e traguardi.
Ovviamente ringrazio anche Giuliano Iorio, senza il quale il fulcro di questa tesi non sarebbe stato possibile.
In questi due anni e mezzo Bologna è stat la cornice perfetta che ha segnato un punto cardine della mia vita. Ringrazio ogni singola persona con cui ho interagito, anche solo scambiando uno sguardo, sapendo che tutti loro stavano vivendo, creando e plasmando la città che mi ha ospitato durante questo percorso. Come non menzionare i miei leggendari coinquilini che, tra tutto, mi hanno insegnato quanto si possa essere diversi e contemporaneamente simili e come si gioca davvero a Risiko.
Ovviamente, tra tutti i miei coinquilini, sono costretto a ringraziare personalmente Pippo, mio compagno di stanza per più di un anno. Sono sicuro, che in camera tua a Firenze, la notte ti svegli ancora sententomi russare. Te mi hai fatto incazzare, io ti sono entrato nella testa. Sono davvero contento di averti avuto come conquilino e che siamo legati per sempre da questa esperienza unica (l'odio viscerale).
Ringrazio i Guelfi tutti per avermi fatto vivere un sogno e aver assaportao un traguardo che, 10 anni fa mai avrei pensato di raggiungere (letteralmente). Potremo dire la vera astrofisica sono gli amici che ci siamo fatti durante la strada, in questo caso non avrei nessun problema a trovare un Ph.d. (Fig **??**). Idiozie a parte, vorrei davvero ringrazie tutte le persone che hanno segnato la mia vita negli ultimi anni quindi le elencherò. Come non iniziare dal gruppo di laboratorio di Computational astrophysics. Auguro a Leo, Ale , Nico(la) E Nicco una vita meracigliosa e ricca di felicità. Ti auguro a Nicola tutto il meglio per la nuova avventura polacca, sapendo che poche davvero persone sono

più intelligenti di te. Vorrei ringraziare Nicco Passa per avermi dimostrato, da quando non l'ho riconosiuto nel corridoio del navile fino ad oggi, la persona stardinaria che è, ti auguro davvero il meglio. Ringrazio Lorenzo Martinelli che mi ha fatto capire che la geologia è, a quanto pare, interessante (frase meme, non posso elencare tutte le cose a cui mi ha introdutto sono le 2AM voglio andare a letto).

Voglio menzionare Ilaria e Nico(lò), detti Sally e McQueen (io non li ho mai chiamati così ma a quanto pare è lore canonica ormai, credo), due persone che manifestano continuamente il loro essere unici in quanto individui e che invitano e ispirano le persone intorno a loro a fare altrettanto, grazie mille per tutto (ora mi piacciano i tatuaggi btw).

Ringrazio Michele, il mio compagno di graduation, per aver partecipato al concerto più bello della mia vita, me lo ricorderò sempre, sono sicuro che diventeri lo scienziato che ti meriti di essere.

Ringrazio Enrico per aver condiviso con me il terrore e il dolore che consegue naturalmente dall'iscriversi ad una laurea in astrofisica, il progetto coinquilinaggio non morirà mai stai attento sto arrivando ARRIVO.

A special shutout needs to be done (in english of course) to my two group mates of the astro-lab, Ardiana and 彭嘉程 . I shared, with you some beautiful and intense moments that i will never forget, you made this experiece special and memorable (still, non of you two told me why it was so awful working with me, i am still waiting for answers). Vorrei ringraziare il mio dirimpettaio San Donninano, Emmanuel, con cui ho condiviso ore e ore di elucubrazioni inconcludenti e fondamentali che mi portero sempre dietro. Sei una delle persone che credo abbia capito meglio cosa significa essere un umano, ti auguro tutta la felicità del mondo e non credo nenanche sia abbastanza. Letizia, il tuo ringraziamento credo sia il più importante e il più difficile da scrivere. Altre parole sarebbero superflue ormai, quindi ti dico solo grazie dal cuore per essere mia amica. Ringrazio tutti i miei amici di Firenze con cui non ho smesso di sentirmi durante questo periodo di distanza (raga è davvero tardi, vi ho ringraziato nella tesi precedente perdonatemi, vo offro da bere pls).

Ringrazio il Pinza, con cui ogni volta che parlo la mia passione per la materia che ho studiato si rinnova, e ogni volta mi ricorda cosa vuol dire capire a fondo un concetto.

Ringrazio Paolo, che credo sia uno dei modelli più importanti per me nel inseguire l'equilibrio.

Ringrazio Pietro perchè con nessuno le parole hanno lo stesso peso e lo stesso significato.

Ringrazio immensamente la mia famiglia per l'incredibile supporto che mi ha permesso di intraprendere questo viaggio. Ringrazio mia Cugina Martina e Lorenzo, a cui auguro una vita felice con la famiglia meravigliosa che si stanno costruendo, auguro a Leo una

vita meravigliosa.
Ringrazio mio fratello, in questo caso pipponi filosofici a parte, lui sa benissimo perchè lo ringrazio e devi capire che sta roba senza te non si faceva ed era la fine, ti sarò grato a vita.
Non ho bisogno di tante parole per ringraziare i miei genitori, le parole che userò hanno tutta la forza necessaria per compiere il loro lavoro.
Ringrazio mia madre per la pazienza e la comprensione.
Ringrazio mio padre per la forza e la resilienza.