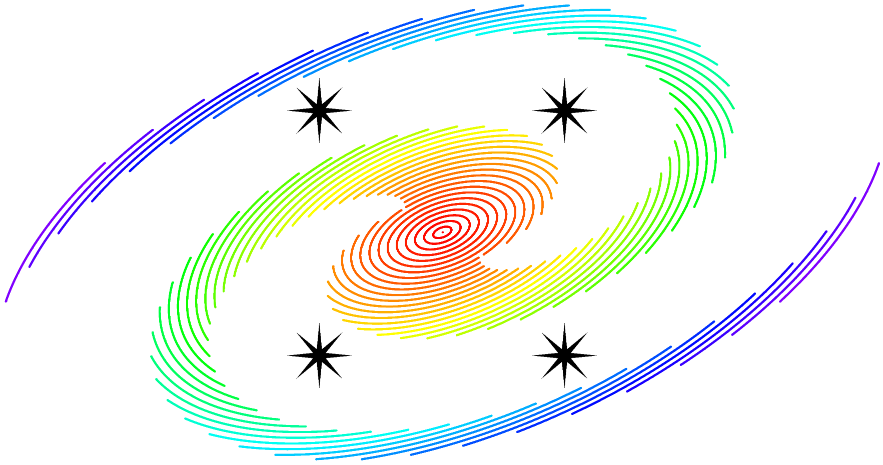
The Pégase code of spectrochemical evolution of galaxies

Version 3.0.1

Pégase



Programme d'Étude des GAlaxies par Synthèse Évolutive

Documentation and complements

Michel Fioc[※] & Brigitte Rocca-Volmerange[※※]

2019-02-22

[※] ‹Michel.Fioc@iap.fr›. Institut d'astrophysique de Paris & Sorbonne université, France.

[※※] ‹Brigitte.Rocca@iap.fr›. Institut d'astrophysique de Paris & université Paris-sud, France.

Appendices

Purpose of the code, reference, license and contact

Pégase.3[※1] is a Fortran 95 code modeling the spectral evolution of galaxies from the far-ultraviolet to submillimeter wavelengths. It also follows the chemical evolution of their stars, gas and dust.

The present document is the user's guide of the code. It also provides complements to the article

__Fioc & Rocca-Volmerange (2019)__, in press in *Astronomy and Astrophysics*

(denoted by "P3AA" hereafter), which should be cited when referring to Pégase.3.

This code is distributed under version 2.1 of the CeCILL license. English and French versions of the latter are provided with this documentation in directory «`doc_dir/`» (see Sec. I.1, p. 3) and are also available at «`https://cecill.info`». In practice, this license is equivalent to the Gnu General Public License and is compatible with it.

To be informed of future developments of the code, ask questions or make comments or suggestions, contact Michel Fioc at

‹`pegase@iap.fr`›.

Overview

For a given *scenario*, i.e. a set of parameters defining the history of mass assembly, the star formation law, the initial mass function, etc., Pégase.3 consistently computes the following:

- the star formation, infall, outflow and supernova rates from 0 to 20 Gyr;
- the stellar metallicity, the abundances of main elements in the gas and the composition of dust;
- the unattenuated stellar spectral energy distribution (SED);
- the nebular SED, using nebular continua and emission lines precomputed with code Cloudy (Ferland et al. 2017);
- the attenuation in star-forming clouds and the diffuse interstellar medium, by absorption and scattering on dust grains, of the stellar and nebular SEDs. For this, the code uses grids of the transmittance for spiral and spheroidal galaxies. We precomputed these grids through Monte Carlo simulations of

[1] Short form for "version 3 of Pégase". "Pégase" is a French acronym for "*__P__rogramme d'__é__tude des __GA__laxies par __s__ynthèse __é__volutive*" ("Program for the study of galaxies by evolutionary synthesis" in English; formerly, the "P" was for "*__P__rojet*"/"Project"). The acronym is pronounced [pegaz] in International Phonetic Alphabet transcription.

The recommended spelling is "`\textsc{P\'egase}`" (in LaTeX 2_{ε} form) or, if small capitals are not available, "`P\'egase`". In contexts where diacritics are problematic (e.g. in filenames), the code's name may be written "`Pegase`", "`PEGASE`"or "`pegase`", in order of decreasing preference.

"Pégase" is also the French name of *Pegasus*, a constellation dominated by an easily recognizable asterism, the Great Square (hence the four stars in the logo).

radiative transfer based on the method of virtual interactions;

- the re-emission by grains of the light they absorbed, taking into account stochastic heating.

The main innovation compared to PÉGASE.2 (Fioc & Rocca-Volmerange 1999, 1997) is the modeling of dust emission and its evolution. While version 2 of the code computed spectra from the far-ultraviolet to the near-infrared ※[2], PÉGASE.3 extends the wavelength range through the mid- and far-infrared up to the submillimetric domain. The computation of nebular emission has also been entirely upgraded to take into account metallicity effects and infrared lines.

Other major differences are that complex scenarios of evolution (derived for instance from cosmological simulations), with several episodes of star formation, infall or outflow, may now be implemented, and that the detailed evolution of the most important elements —not only the overall metallicity— is followed. More details are provided in P3AA and Sec. II (p. 8 *et seq.*).

I. Getting started

1. Installation

The code is freely available at

«http://www.iap.fr/users/fioc/Pegase/Pegase.3/» ※[3]

and «www.iap.fr/pegase/». Download file «Pegase.3.0.1.tar.gz» and type in a terminal ※[4]

“tar xvf Pegase.3.0.1.tar.gz”

in the directory where the archive has been saved. This extracts all the files in directory «Pegase.3.0.1/».

In addition to the file «Read_me.txt.html», which gives some instructions to compile and test the code, the directory «Pegase.3.0.1/» contains the following subdirectories:

- **«ages_dir/»**: files determining the ages at which executables print their outputs;
- **«bin_dir/»**: «Makefile», executable files, log files and all the files created by the compilation;
- **«calib_dir/»**: filter passbands, spectra of reference stars and output file of code «calib»;
- **«Cloudy_dir/»**: nebular continua and a selection of emission lines computed by CLOUDY (Ferland et al. 2017). Emission lines are listed in file «list_neb_lines.txt»;
- **«colors_dir/»**: files produced by code «colors»;
- **«doc_dir/»**: this documentation, the article in *Astronomy & Astrophysics* (file «article_Pegase.3.pdf»), changes with respect to previous versions (file «Changes.txt.html»), and French and English versions of the CeCILL license (in HTML and text format; files «License_CeCILL_V2.1-*»);
- **«dust_dir/»**: input files related to dust properties;
- **«grain_SED_dir/»**: files of grain SEDs optionally produced by code «spectra»;
- **«grain_temp_dir/»**: files of grain temperatures optionally produced by code «spectra»;
- **«IMFs_dir/»**: initial mass functions;
- **«RT_dir/»**: input files related to radiative transfer;
- **«scenarios_dir/»**: files created by the user and containing the scenarios used as input by code «spectra»;
- **«source_dir/»**: Fortran source files specifically related to PÉGASE;
- **«spectra_dir/»**: main output files produced by code «spectra»;
- **«SSPs_dir/»**: files containing properties of single stellar populations and produced by code «SSPs»;
- **«stel_lib_dir/»**: libraries of stellar spectra;
- **«tracks_dir/»**: stellar evolutionary tracks;
- **«util_dir/»**: Fortran source files of general interest;
- **«yields_dir/»**: yields of elements produced by stars.

All the text and Fortran files are in UTF-8 format.

[2] In the presence of dust, PÉGASE.2 energy distributions were not significant at longer wavelengths.

[3] See App. A, p. 47, for a description of the typographical conventions used in this documentation.

[4] We assume hereafter that the user works with a command-line interface on a LINUX/UNIX operating system.

2. Compilation

a. Executables and their input and output data files

Go in subdirectory «bin_dir/» of «Pegase.3.0.1/» and type "make" in a terminal to compile the code according to the instructions given in file «Makefile» ※[5]. This creates the following executable files in «bin_dir/»:

- **«SSPs»**: code computing several evolving properties of single stellar populations ("SSPs") —but not their spectral energy distribution. «SSPs» creates a file, called hereafter a ***set of SSPs***, listing the names of the other output files produced during the run;
- **«spectra»**: code processing one or more scenarios of galaxy evolution and computing, for each of them, the SED and various other properties of the modeled galaxy as a function of time.

  For each scenario, the main output of «spectra» is a ***file of spectra*** containing aforementioned data. Optionally, one or two auxiliary files are also produced for stochastically heated dust grains:

  - a ***file of grain temperatures***, which provides the temperature probability distribution of individual grains of various species and sizes;
  - a ***file of grain SEDs***, which contains their SEDs.

  As input, «spectra» needs a ***file of scenarios*** written by the user and defining, for each of the scenarios it contains, the parameters of the modeled galaxy ※[6];
- **«calib»**: code computing several properties for a series of filters, in particular the in-band fluxes of reference stars used to calibrate magnitudes;
- **«colors»**: code computing the colors and other properties of a galaxy for a given scenario of evolution. The output is written in a ***file of colors***.

  Input: file of spectra produced by «spectra» for this scenario. Code «colors» also uses the output file of «calib»;
- **«plot_spectra»**: code plotting the evolving SED of a galaxy for one or more scenarios.

  Inputs: files of spectra produced by «spectra» for these scenarios;
- **«plot_grain_temp»**: code plotting the temperature probability distribution of stochastically heated dust grains for a given scenario.

  Input: file of grain temperatures produced by «spectra» for this scenario;
- **«plot_grain_SED»**: code plotting the SED of individual grains for a given scenario.

  Input: file of grain SEDs produced by «spectra» for this scenario.

b. Compiler

The default compiler is GFortran. To use IFort instead, comment the line "COMPILER = gfortran" in «Makefile» (i.e., prefix this line with a "#") and uncomment the line "# COMPILER = ifort".

To use another Fortran 95 compiler, comment the lines "COMPILER = gfortran" and "COMPILER = ifort", add a line such as "COMPILER = *compil_command*" and modify the options of compilation (variable "OPTIONS") in accordance. (Actually, the options defined for GFortran and IFort are for debugging, and removing them might speed up the execution.)

If you have run ‹make› for some Fortran compiler already and you want to switch to another one, once you have changed ‹COMPILER› in «Makefile», type first "make clean" in a terminal, then "make". (The command ‹make clean› deletes all the object and module files, as well as the executables; otherwise, ‹make› will not recompile the code.)

c. Graphical library for «plot_spectra», «plot_grain_temp» and «plot_grain_SED»

The codes «plot_spectra», «plot_grain_temp» and «plot_grain_SED» require the PGPlot graphical library (Pearson 2002). The location of this library on the author's system is set by the variable ‹PGPLOT_LIB› of «Makefile», both for IFort and for GFortran. To use «plot_spectra», «plot_grain_temp» or «plot_grain_SED» on your system, you will most certainly have to redefine ‹PGPLOT_LIB›. You may also replace PGPlot instructions (lines beginning with "call pg") in «plot_spectra.f90», «plot_grain_temp.f90» or «plot_grain_SED.f90» (all in directory «source_dir/») by equivalents in your favorite graphical language.

[5] This file is written for the Gnu ‹make› command.

[6] This file points in particular to a set of SSPs.

d. Home directory and directory separator

To be able to refer to your home directory through the symbol "~", you need to provide the path from the root directory down to the home directory[7]. For this, edit file «`util_dir/mod_dir_access.f90`» and follow the instructions in this file to assign parameter ‹`home_dir`› to the correct value for this path.

On non-LINUX/UNIX systems, you may also need to change the value of the directory separator (parameter ‹`dir_sep`› in the same file.)

If you change any of these parameters, recompile the code with ‹`make`›.

e. Real numbers

Except for plots with PGPLOT, the range and precision used for real numbers are set by parameter ‹`CDR`›. By default, ‹`CDR`› gives double precision reals. If memory is missing, you may select single precision reals by replacing "`DPR`" with "`SPR`" in the statement "`integer, parameter :: CDR = DPR`" of file «`util_dir/mod_types.f90`». Recompile the code with ‹`make`› afterwards.

3. Execution

We recommend that, to become acquainted with the code, the user first run once «`SSPs`», «`spectra`» and «`colors`» as in the examples below.

a. Running «SSPs»

«`SSPs`» computes from stellar evolutionary tracks of various metallicities the evolution of an SSP for a given initial mass function. It calculates isochrones (i.e. the locus of stars having the same age but different masses) in the Hertzsprung & Russell diagram and apportions them among the elements of a library of stellar spectra. «`SSPs`» also computes from a set of stellar yields the ejection rate in the interstellar medium, by an SSP, of gas enriched in metals (globally and for each main element) and of circumstellar dust, as well as the number of supernovae and the mass locked in compact stellar remnants.

⟶ Sec. III, p. 16, for details on the inputs to «`SSPs`».

To run «`SSPs`», just type "`./SSPs`" in «`bin_dir/`» and answer the questions.

Here is a minimal example[8]:

```
► | [═══]
  | Initial mass function? [═══]                               ;
◄ | ↵
► | Lower mass of the IMF (in solar masses)? [═══]             ;
◄ | ↵
► | Upper mass of the IMF (in solar masses)? [═══]             ;
◄ | ↵                                                          ;
► | [═══]
  | Set of evolutionary tracks? [═══]                          ;
◄ | ↵                                                          ;
► | [═══]
  | Set of yields for high-mass stars? [═══]                   ;
◄ | ↵                                                          ;
► | [═══]
  | Set of stellar libraries? [═══]                            ;
◄ | ↵                                                          ;
► | [═══]
  | File defining the output ages of code "SSPs"? [═══]        ;
◄ | ↵                                                          ;
```

[7] Some compilers, e.g. GFORTRAN, do not associate "~" to the home directory.

[8] The symbol [═══] denotes lines written on the terminal but skipped here; ↵ means that the "Return"/"Enter" key has been pressed, which selects the default answer; ► and ◄ distinguish outputs to the terminal and inputs from the keyboard.

```
► [═══]
  Identifier prefixed to the output files of code "SSPs"?
```

(provide a string of characters. For the first run, we recommend to type "`example`" because this is the identifier used in the file input to «`spectra`» in Sec. I.3.*b*);

```
◄ example
```

.

Once «`SSPs`» has finished running, several output files have been created in «`SSPs_dir/`». The master file —the "set of SSPs"— is named "*`prefix`*`_SSPs.txt`", where ‹*`prefix`*›[9] is the value of the identifier requested in the last question above[10]. The set of SSPs («`example_SSPs.txt`» in the example) lists the names of the associated output files (one per file of stellar evolutionary tracks, each one corresponding to an initial metallicity; their name begins also with "*`prefix`*`_`"). These files contain all the information needed by «`spectra`» to compute the evolving SED of a single stellar population at any reasonable metallicity; they are not meant to be read by the user, but are a prerequisite for «`spectra`».

All the sets of SSPs produced by «`SSPs`» are listed in «`SSPs_dir/list_SSPs_sets.txt`».

b. Running «spectra»

For each galaxy, a scenario of evolution must be given. Scenarios are read from a file containing the parameters of one or more scenarios[11]. These parameters are organized in trees; many have a default value.

⟶ Sec. IV, p. 19, for a complete description of the contents of a file of scenarios;

⟶ App. E, p. 52, to consult the trees of parameters and the tables providing their type, possible values and, if any, their default value.

As an example, let us use the file «`example_scenarios.txt`»[12] (this requires that «`example_SSPs.txt`» has first been created by «`SSPs`»; see Sec. I.3.*a*). Type "`./spectra`" in «`bin_dir/`», and when asked

```
► Name of the file of scenarios? [═══]
```

,

provide this name[13]:

```
◄ example_scenarios.txt
```

.

For each scenario, «`spectra`» creates a "file of spectra" containing, among other data, the SED of the galaxy as a function of time (file "`example_spectra1.txt`" for the first scenario in the example considered here).

If requested (see Sec. IV.3.*j*.*β*, p. 31), «`spectra`» also outputs for each scenario the temperature probability distribution and the spectral energy distribution of individual dust grains in a "file of grain temperatures" and a "file of grain SEDs", respectively.

Unless another directory is specified, all the files of spectra (resp. grain temperatures, grain SEDs) are written in «`spectra_dir/`» (resp. «`grain_temp_dir/`», «`grain_SED_dir/`»). If their names are not given in the file of scenarios, or if they are already attributed, «`spectra`» automatically assigns names to them. The files of spectra (resp. grain temperatures, grain SEDs) are listed in the file «`spectra_dir/list_spectra.txt`» (resp. «`grain_temp_dir/list_grain_temp.txt`», «`grain_SED_dir/list_grain_SED.txt`»).

A ***log file*** is also created in subdirectory «`log_spectra_dir/`» of «`bin_dir/`». Its name is built by prefixing the time-stamp mentioned in Sec. IV.3.*j*.*α*.ii, p. 30 and the character "`_`" to the name of the file of scenarios.

c. Running «calib»

Code «`calib`» produces the file "`calib.txt`" in «`calib_dir/`», which «`colors`» needs to compute in-band fluxes and magnitudes.

«`calib`» is automatically run whenever you compile all the codes with a simple "`make`" command. You

[9] We use an italic font to distinguish placeholders.

[10] Creating a specific directory would clutter less «`SSPs_dir/`», but it is impossible in standard Fortran 95 to do this from within the code. The Fortran 2008 ‹`execute_command_line`› subroutine or system-specific procedures are needed for this; we plan to implement them in future versions of PÉGASE. Nonetheless, if the prefix ends with a directory separator (symbol "`/`" on LINUX/UNIX or "`\`" on WINDOWS), the code will ask you to create the directory by yourself; once done, il will write the files of SSPs (without the leading "*`prefix`*`_`" in their names) in this directory.

[11] In PÉGASE.2, this file was created by code «`scenarios`» through an interactive dialog. Because of the large number of parameters, this would have been very inconvenient in PÉGASE.3, and the file of scenarios must now be written as a series of statements with a text editor. Although the user is less guided, it makes it much easier to correct mistakes and to update the file.

[12] This file is available in «`scenarios_dir/`» and is described in Sec. IV.2, p. 19.

[13] If the file of scenarios is not in «`scenarios_dir/`», either its path relative to «`scenarios_dir/`» or its absolute path must be given.
The same holds for the files of spectra input to «`colors`» and «`plot_spectra`» with respect to directory «`spectra_dir/`», the files of grain temperatures and SEDs input to «`plot_grain_temp`» and «`plot_grain_SED`» with respect to «`grain_temp_dir/`» and «`grain_SED_dir/`», and for the output file of «`colors`» with respect to «`colors_dir/`».

will also need to run it again every time you add filters or change them (all filters are listed in «calib_dir/list_filters.txt»); to do this, type "./calib" in «bin_dir/».

d. Running «colors»

Code «colors» computes colors, mass-to-light ratios, equivalent widths of emission lines, etc., for a given scenario. Type "./colors" in «bin_dir/» to run it, and when asked

```
▶ Name of the input file (file of spectra)? [══]
```

follow the instructions to select the name of the file of spectra corresponding to the scenario. For instance, type

```
◀ example_spectra1.txt
```

Code «colors» then asks

```
▶ Name of the output file? [══]
```

Answer

```
◀ ↵
```

to select the default name. This creates the file "colors_*name*" ("colors_example_spectra1.txt" in the example) in «colors_dir/», where "*name*" is the name of the file of spectra (stripped from its path, if present).

You may choose another name than the default one for the output file of «colors». Unless a path is specified, it will be written in «colors_dir/».

e. Running «plot_spectra»

Code «plot_spectra» reads one or more files of spectra. For each age, it plots or overplots the SEDs on a log-log plot, with the wavelength λ (in Å) on the x-axis and $\lambda\,\widehat{L}_\lambda$ on the y-axis, where $\widehat{L}_\lambda$ (in erg s^{-1} Å$^{-1}/M_{\text{sys}}$) is the continuum monochromatic luminosity (spectral power), per unit wavelength, normalized to the mass M_{sys} of the system (see Sec. II.1.*b*, p. 8).

Pégase does not compute the profile of emission lines, only their integrated luminosity. For the plots it creates, «plot_spectra» considers that all lines have a Gaussian profile. The standard deviation σ_ℓ, in wavelength, of the ℓ-th line is related to the full width at half-maximum[14] of the velocity distribution, W, by

$$\sigma_\ell = \frac{\lambda_\ell\,W}{2\,\sqrt{2\ln 2}\;c}\,, \tag{1}$$

where λ_ℓ is the wavelength of the line and c is the speed of light. Emission lines are represented as spikes of height

$$\frac{\lambda_\ell\,\widehat{\mathcal{L}}_\ell}{\sqrt{2\,\pi}\;\sigma_\ell}\,, \tag{2}$$

where $\widehat{\mathcal{L}}_\ell$ is the normalized integrated luminosity of the line (in erg s$^{-1}/M_{\text{sys}}$). Note that the line profiles themselves are not shown.

To run «plot_spectra», type "./plot_spectra" in «bin_dir/» and follow the instructions to select the files of spectra, change the value of W and define the bounds of the plot.

f. Running «plot_grain_temp» and «plot_grain_SED»

Code «plot_grain_temp» plots the temperature probability distribution of individual dust grains for a given scenario; «plot_grain_SED» plots their spectral energy distribution.

To run «plot_grain_temp» (resp. «plot_grain_SED»), type "./plot_grain_temp" (resp. "./plot_grain_SED") in «bin_dir/», provide the name of a file of grain temperatures (resp. of grain SEDs) when requested and follow the instructions.

[14] The default value of W is the parameter ‹FWHM_v_def› in «plot_spectra.f90»; if you change it, recompile with ‹make›. You may also enter another value of W while running «plot_spectra».

II. Modeling

The modeling of galaxies by Pégase.3 is described in P3AA. We repeat here only what is required for the consistency of this documentation and what the user needs to know to run the code. We also add a few technical details, especially on the meaning of some parameters.

1. The system: zones

a. Zones

The ***system*** considered in code «spectra» is made of several distinct zones:

- the ***galaxy*** proper, modeled as one zone, where all the processing occurs, and from where all the light comes. Initially, the galaxy contains only interstellar gas (or nothing if entirely formed by subsequent infall), but no stars or compact stellar remnants ※[15]; this gas is primordial by default. The galaxy contains initially no dust either, unless the gas metallicity has been set to some non-null value and the basic model of dust evolution is adopted (Sec. II.5.$b.\alpha$, p. 12);
- zero, one or more ***reservoirs***. These contain only gas (primordial by default), which may fall onto the galaxy via ***infall***. As modeled, nothing may enter them;
- the part of the ***intergalactic medium*** (IGM) produced by ***outflows*** (= galactic winds) of interstellar matter (gas and dust) from the galaxy only. Nothing may exit from the IGM.

The mass assembly history of the galaxy is determined by the relative initial masses of the galaxy and of the reservoirs, and by the infall and outflow rates. One has

$$M_{\mathrm{gal}}(t) = M_{\mathrm{gal}}(0) + \int_{t'=0}^{t} \left(\dot{M}_{\mathrm{in}}[t'] - \dot{M}_{\mathrm{out}}[t']\right) \mathrm{d}t', \tag{3}$$

where $M_{\mathrm{gal}}(t)$ is the mass of the galaxy at age t, and $\dot{M}_{\mathrm{in}}$ and $\dot{M}_{\mathrm{out}}$ are the total infall and outflow rates. One or more instantaneous or extended, overlapping or consecutive ***episodes of infall*** or ***outflow*** may occur.

⟶ Sec. IV.3.*d*, p. 22, and App. E.4, p. 53, to set the parameters for reservoirs and infall episodes;

⟶ Sec. IV.3.*f*, p. 25, and App. E.6, p. 57, to set the parameters for outflow episodes.

b. System's mass and normalized quantities

Many quantities are ***normalized*** to the constant ***mass of the system***,

$$M_{\mathrm{sys}} := M_{\mathrm{gal}}(0) + \sum_{j=1}^{n_{\mathrm{res}}} M_{\mathrm{res},\,j}(0) = M_{\mathrm{gal}}(t) + \sum_{j=1}^{n_{\mathrm{res}}} M_{\mathrm{res},\,j}(t) + \int_{t'=0}^{t} \dot{M}_{\mathrm{out}}(t')\,\mathrm{d}t', \tag{4}$$

where n_{res} is the number of reservoirs, $M_{\mathrm{res},\,j}$ is the mass of the j-th reservoir and $\int_{t'=0}^{t} \dot{M}_{\mathrm{out}}(t')\,\mathrm{d}t'$ is the mass increment of the IGM due to outflows from the galaxy. *Normalized quantities are denoted by a hat accent hereafter.* For instance, $\widehat{M}_{\mathrm{gal}}(t) = M_{\mathrm{gal}}(t)/M_{\mathrm{sys}}$. See App. D.1, p. 50, for explanations on how to relate normalized quantities computed by the code to observed ones.

c. Components

The galaxy contains ***stars***, an ***interstellar medium*** (ISM) and ***compact stellar remnants*** (i.e. white dwarfs, neutron stars and black holes, but excluding stellar ejecta). By "stars", we usually mean both stars proper (i.e. regulated by thermonuclear reactions), called ***live stars*** hereafter, and ***inert*** (e.g. substellar) ***objects***. Note that, although the latter are assumed to be dark and unevolving, they lock some matter during star formation, which affects the history of the galaxy and thus, indirectly, its spectral energy distribution. We also neglect the emission of light by late white dwarfs, neutron stars and the surroundings of black holes.

By ***gas*** and ***dust***, unless otherwise specified, we mean the gas and the dust in the ISM only: not that in stars, their circumstellar envelopes, the reservoirs or the IGM. The stars and the interstellar gas are mainly made of hydrogen and helium. As is traditional in astrophysics, all other elements are called ***metals***. Except for a negligible fraction of hydrogen atoms, dust consists only in metals. We give the abundances of elements in terms of mass fraction, not of relative numbers of atoms; in particular, what we call ***metallicity*** of a component

[15] See Sec. II.1.*c* for definitions of the various components.

is the overall mass fraction of metals in that component,

$$Z = \frac{M(\text{metals})}{M(\text{H}) + M(\text{He}) + M(\text{metals})}\,. \tag{5}$$

d. Regions

To compute the attenuation of the light by dust grains and the re-emission of the latter, we distinguish two ***regions*** in the interstellar medium:

- ***the diffuse ISM*** (DISM);
- ***star-forming clouds*** (SFC).

2. Single stellar populations

⟶ Sec. III, p. 16.

The stellar content of a galaxy is a mixture of ***single stellar populations*** (SSP), i.e. ensembles of stars formed instantaneously, simultaneously and with the same initial chemical composition. The monochromatic luminosity of an SSP (per unit of initial mass of the SSP) at age t' and with an initial chemical composition $\{\chi_0\} := \{\chi_0(\text{H}), \chi_0(\text{He}), \dots\}$ is

$$L_\lambda^{\text{ssp}}(t', \{\chi_0\}) = \int L_\lambda^{\text{star}}(m, t', \{\chi_0\})\,\mathrm{d}n, \tag{6}$$

where $L_\lambda^{\text{star}}(m, t', \{\chi_0\})$ is the monochromatic luminosity at age t' of a star of initial mass m and initial composition $\{\chi_0\}$, and $\mathrm{d}n$ is the number of stars, per unit of initial mass of the SSP, born with a mass in $[m, m + \mathrm{d}m[$ [※16]. This number is computed from the ***initial mass function*** (IMF) ϕ as

$$\mathrm{d}n = \phi(m)\,\mathrm{d}(\ln m), \tag{7}$$

where ϕ is normalized, i.e.

$$\int m\,\phi(m)\,\mathrm{d}(\ln m) = 1. \tag{8}$$

⟶ Sec. III.1, p. 16, to choose the IMF used in «SSPs»;

⟶ Sec. III.5, p. 18, to choose the output ages of «SSPs».

In the code, L_λ^{star} is computed from ***stellar evolutionary tracks*** and a ***library of stellar spectra*** as

$$L_\lambda^{\text{star}}(m, t', \{\chi_0\}) = \mathcal{L}_{\text{star}}(m, t', \{\chi_0\})\,\ell_\lambda^{\text{star}}(\{\chi_{\text{surf}}\}, T_{\text{eff}}, g), \tag{9}$$

where $\mathcal{L}_{\text{star}}$ is the bolometric luminosity of the star, i.e. the *amplitude* of the stellar spectrum, and the function $\lambda \mapsto \ell_\lambda^{\text{star}}$ is the *shape* of this spectrum. This shape depends only on the surface composition $\{\chi_{\text{surf}}\}$ of the star, its effective temperature T_{eff} and its surface gravity g [※17].

$\mathcal{L}_{\text{star}}$, T_{eff}, g and $\{\chi_{\text{surf}}\}$ are given by stellar evolutionary tracks as a function of m and t'. They also depend on the initial composition $\{\chi_0\}$, but because tracks are computed only as a function of the initial metallicity Z, with hydrogen and helium abundances determined by Z and fixed relative ratios of abundances for metals [※18], we use Z as a substitute for $\{\chi_0\}$.

⟶ Sec. III.2, p. 17, to choose the stellar evolutionary tracks used in «SSPs».

The relative monochromatic luminosities $\ell_\lambda^{\text{star}}(\{\chi_{\text{surf}}\}, T_{\text{eff}}, g)$ are obtained by interpolating the shapes $\lambda \mapsto \ell_{\lambda,i}^{\text{lib}}$ of the elements of a library of stellar spectra as a function of the chemical composition [※19], of T_{eff} and g: $\ell_\lambda^{\text{star}} = \sum_i \alpha_i\,\ell_{\lambda,i}^{\text{lib}}$, where the weights α_i are independent of λ, non negative and $\sum_i \alpha_i = 1$ [※20].

[16] Binary stars are not taken into account in the spectral evolution (but they have an impact on the chemical evolution because of the model assumed for type Ia supernovae; see Sec. II.3, p. 10).

[17] We neglect the effect of rotation and stellar winds on the spectrum.

[18] We consider only tracks at $[\alpha/\text{Fe}] = 0$.

[19] Such libraries usually provide spectra only as a function of Z (or, equivalently, [Fe/H]). Because the evolution of the surface composition is negligible, except maybe during the latest phases, we take the initial value of Z instead of $\{\chi_{\text{surf}}\}$.

[20] We never extrapolate the spectra because it might lead to negative fluxes: when the metallicity of the star is outside the range of the library's spectra, we take the closest one; we do the same for the gravity and the temperature, but give a larger weight to the latter. In the best case, we keep the eight nearest points bracketing the point (Z, T_{eff}, g) in the $(Z^{\text{lib}}, T_{\text{eff}}^{\text{lib}}, g^{\text{lib}})$ space, so at most eight of the α_i are non null.

The relative monochromatic luminosity $\ell^{\text{lib}}_{\lambda,i}$ of an element of the library is calculated as

$$\ell^{\text{lib}}_{\lambda,i} = L^{\text{lib}}_{\lambda,i} \Big/ \int L^{\text{lib}}_{\lambda,i}\,\mathrm{d}\lambda\,, \tag{10}$$

where $L^{\text{lib}}_{\lambda,i}$ is the absolute monochromatic luminosity of the element, and $\int L^{\text{lib}}_{\lambda,i}\,\mathrm{d}\lambda$ is its bolometric luminosity [※21].

⟶ Sec. III.4, p. 18, to choose the libraries of stellar spectra used in «SSPs».

L^{SSP}_{λ} is computed by code «SSPs» from Eq. (6). To do this, «SSPs» derives from the evolutionary tracks the ***isochrone*** of an SSP at age t, i.e. the locus of all the stars in the $(\mathcal{L}_{\text{star}}, T_{\text{eff}}, g)$ space.

3. Star formation and chemical evolution

The unattenuated stellar monochromatic luminosity of a galaxy at age t and wavelength λ is

$$L^{\star,0}_{\lambda}(t) = \int_{t'=0}^{t} \psi(t-t')\, L^{\text{SSP}}_{\lambda}\big(t', Z[t-t']\big)\,\mathrm{d}t', \tag{11}$$

where $\psi(t-t')$ is the ***star formation rate*** (including inert objects) at age $t-t'$ and $Z(t-t')$ is the metallicity of the interstellar medium at that age. One or more instantaneous or extended, overlapping or consecutive ***episodes of star formation*** may occur.

⟶ Sec. IV.3.*e*, p. 23, and App. E.5, p. 55, to set the parameters of star formation episodes.

Stars eject matter in the ISM through stellar winds and when they explode as supernovae. In low-mass stars, stellar winds occur mainly during the Red (or “First”) Giant Branch (RGB) and the Asymptotic Giant Branch (AGB) phases; the final remnant is a white dwarf [※22]. High-mass stars undergo strong stellar winds during their whole life and usually end as core-collapse supernovae (type II, Ib or Ic); the final remnant is a neutron star or a black hole. In all cases, when computing the chemical evolution of a galaxy, we assume that the ejection happens only at the end of the life of stars: this is justified by the short life of high-mass stars compared to the age of a galaxy and by the late onset of intense winds in low-mass stars.

According to the favorite model for type Ia supernovae, these occur in close binaries where the primary star becomes a CO white dwarf: because the mass ejected by the secondary falls on the primary, the latter may reach the Chandrasekhar mass and become a type Ia supernova; no compact remnant is then left. We use the prescriptions of Matteucci & Greggio (1986) to model the rate of type Ia supernovae and the ejecta of close binaries [※23]. Note that, beside this, all stars are considered as isolated. The fraction of close binaries is set by parameter ‹close_bin_frac› (see Sec. IV.3.*c*, p. 21).

The chemical evolution of a galaxy is determined by the history of the rates ψ, $\dot{M}_{\text{in}}$ and $\dot{M}_{\text{out}}$ of star formation, infall and outflow. In particular, the evolution of the mass M_{ISM} of the interstellar medium is given by

$$\frac{\mathrm{d}M_{\text{ISM}}}{\mathrm{d}t} = -\psi(t) + \dot{M}_{\text{in}}(t) - \dot{M}_{\text{out}}(t) + \int_{t'=0}^{t} \psi(t-t')\,\dot{M}^{\text{SSP}}_{\text{ej}}\big(t', \{\chi_0\}[t-t']\big)\,\mathrm{d}t', \tag{12}$$

where $\dot{M}^{\text{SSP}}_{\text{ej}}(t', \{\chi_0\}[t-t'])$ is the mass ejection rate of matter into the ISM at age t' by the stars of an SSP formed with the initial composition $\{\chi_0\}(t-t')$, per unit of initial mass of the SSP.

For the evolution of the mass of metals in the ISM, we have

$$\frac{\mathrm{d}(M_{\text{ISM}}\,Z_{\text{ISM}})}{\mathrm{d}t} = -(\psi\,Z_{\text{ISM}})(t) + \sum_{j=1}^{n_{\text{res}}} \dot{M}_{\text{in},j}(t)\,Z_{\text{in},j} - (\dot{M}_{\text{out}}\,Z_{\text{ISM}})(t) + \int_{t'=0}^{t} \psi(t-t')\,\dot{M}^{\text{SSP}}_{\text{ej},Z}\big(t', \{\chi_0\}[t-t']\big)\,\mathrm{d}t', \tag{13}$$

where Z_{ISM} is the mass fraction of metals in the ISM, $Z_{\text{in},j}$ is the same quantity in reservoir j, $\dot{M}_{\text{in},j}$ is the infall rate from reservoir j, and $\dot{M}^{\text{SSP}}_{\text{ej},Z}$ is the mass ejection rate of metals by an SSP. The code computes the evolution of the abundances of He, C, N, O, Ne, Mg, Si, S, Ca and Fe in the ISM with similar equations. In these, we assume that the composition of galactic outflows is the same as that of the ISM and that stellar ejecta are instantaneously and homogeneously mixed with the ISM.

[21] For spectra available on too narrow a wavelength range, in particular observed ones, the integral in Eq. (10) underestimates the bolometric luminosity. One has then to rely on bolometric corrections (see App. D.2, p. 50), but these may be inconsistent with the spectra.

[22] In the evolutionary tracks based on Groenewegen & de Jong (1993) that we use for the thermally pulsing AGB (TPAGB) phase, a core-collapse may however happen if the core mass exceeds the Chandrasekhar mass.

[23] The constants ‹min_mass_SNIa› and ‹close_bin_gamma› correspond, respectively, to the parameters M_{Bm} and γ defined in sec. 2.1 of Matteucci & Greggio (1986). They are set in «source_dir/mod_SSPs_constants.f90» and may be modified. To take a modification into account, recompile the code (with ‹make›; see Sec. I.2, p. 4) and run «SSPs» again (cf Sec. I.3.*a*, p. 5).

4. Stellar yields

⟶ Sec. III.3, p. 17, to choose the yields used in «SSPs».

The ejection rate in Eq. (12) is given by

$$\dot{M}_{\rm ej}^{\rm SSP}(t', \{\chi_0\}) = \int \dot{m}_{\rm ej}(m, t', \{\chi_0\})\, \phi(m)\, {\rm d}(\ln m), \tag{14}$$

where $\dot{m}_{\rm ej}(m, t', \{\chi_0\})$ is the mass ejection rate at age t' of a star of initial mass m and initial composition $\{\chi_0\}$. As mentioned in previous section, we assume that the ejection occurs at once, at the end of the life of the star, i.e. at the end of the TPAGB phase for low-mass stars or when they explode for those becoming supernovae:

$$\dot{m}_{\rm ej}(m, t', \{\chi_0\}) = m_{\rm ej}(m, \{\chi_0\})\, \delta(t' - t_{\rm end}[m, \{\chi_0\}]), \tag{15}$$

where $t_{\rm end}(m, \{\chi_0\})$ is the life duration of the star and δ is the Dirac delta “function”. Note that *recycling is not instantaneous*. As usual and for want of better, we use Z as a substitute for $\{\chi_0\}$ and interpolate the total ejected mass $m_{\rm ej}$ as a function of m and Z from a ***set of yields***.

Sets of yields also provide either the ***gross yield*** or the ***net yield*** for various elements: the gross yield of any element i is the total mass $m_{{\rm ej},\,i}$ of i ejected by a star in the ISM; the net yield $m_{{\rm ej},\,i}^{\rm net}$ is related to the gross yield by

$$m_{{\rm ej},\,i} = m_{{\rm ej},\,i}^{\rm net} + \chi_{0,\,i}\, m_{\rm ej}\,, \tag{16}$$

where $\chi_{0,\,i}$ is the initial abundance of i in the star, $\chi_{0,\,i}\, m_{\rm ej}$ is the portion of the initial mass released in the ISM and originally in the form of i, and $m_{{\rm ej},\,i}^{\rm net}$ is the additional ejected mass of i. (The same holds for all the metals together: $m_{{\rm ej},\,Z} = m_{{\rm ej},\,Z}^{\rm net} + Z\, m_{\rm ej}$.)

The ejection rate of element i in the ISM by an SSP is

$$\dot{M}_{{\rm ej},\,i}^{\rm SSP}(t', \{\chi_0\}) = \int \dot{m}_{{\rm ej},\,i}(m, t', \{\chi_0\})\, \phi(m)\, {\rm d}(\ln m). \tag{17}$$

Because sets of yields are only available for fixed initial compositions, which cannot be interpolated (for all elements simultaneously) to the composition $\{\chi_0\}$ of the ISM when stars form, it is more accurate, in order to compute $\dot{M}_{{\rm ej},\,i}^{\rm SSP}$, to interpolate the net yields rather than the gross yields as a function of the initial stellar metallicity Z, and to use $\{\chi_0\}$ for the original contribution. Using the assumption expressed by Eq. (15), i.e. $\dot{m}_{{\rm ej},\,i}(m, t', \{\chi_0\}) = m_{{\rm ej},\,i}(m, \{\chi_0\})\, \delta(t' - t_{\rm end}[m, \{\chi_0\}])$, one has then

$$\dot{M}_{{\rm ej},\,i}^{\rm SSP}(t', \{\chi_0\}) \approx \int m_{{\rm ej},\,i}^{\rm net}(m, Z)\, \delta(t' - t_{\rm end}[m, Z])\, \phi(m)\, {\rm d}(\ln m) + \chi_{0,\,i}\, \dot{M}_{\rm ej}^{\rm SSP}(t', Z). \tag{18}$$

5. Dust grains

a. Grain size distribution

Two ***families*** of dust grains are considered: silicates and carbonaceous grains. The latter family is itself subdivided in three ***grain species***: graphites, neutral PAHs (Polycyclic Aromatic Hydrocarbons) and ionized PAHs. The silicate family is a grain species by itself.

The size distributions of grains and the relative masses of the various species within a given family are constant in the code. Three models are provided currently for these quantities:

- the “BARE_GR_S” model of Zubko et al. (2004) (file «ZDA.txt» in «dust_dir/»);
- the model number 7 of table 1 of Weingartner & Draine (2001) used in Li & Draine (2001) (file «LWD.txt»);
- the outdated model, without PAHs, of Mathis et al. (1977) (file «MRN.txt»).

⟶ Sec. IV.3.*h*.*β*, p. 28, and App. E.8, p. 61.

The files containing the size distributions also point to files giving the optical properties (absorption and scattering opacities, asymmetry parameter) of individual grains as a function of wavelength (see Sec. II.5.*c*, p. 13). The wavelenth range is [10 Å, 1 mm], with resolutions $\lambda/\Delta\lambda \approx 17$ for graphites and silicates and $\lambda/\Delta\lambda \approx 90$ for PAHs.

b. Dust evolution

⟶ Sec. IV.3.*g*, p. 27, and App. E.7, p. 59.

While size distributions of grains and relative masses of dust species within a given family are constant, the overall masses in the ISM of the silicate and carbonaceous dust families evolve with time. Two mutually exclusive models are implemented in «`spectra`»:

α. Basic model

In this model, the mass of dust is simply proportional to the mass of its constituents in the ISM: one does not consider the processes by which dust is produced and destroyed. The mass of carbonaceous grains in the ISM is given by

$$M^{\text{ISM}}_{\text{carb}}(t) = \delta^{\text{ISM}}_{\text{carb}}\, M^{\text{ISM}}_{\text{C}}(t), \tag{19}$$

where $M^{\text{ISM}}_{\text{C}}$ is the mass of carbon in the ISM (including dust grains) and $\delta^{\text{ISM}}_{\text{carb}}$, the depletion factor for carbon atoms, is constant. (The mass of hydrogen atoms in carbonaceous grains is neglected.)

The mass of silicate grains in the ISM is given by

$$M^{\text{ISM}}_{\text{sil}}(t) = \delta^{\text{ISM}}_{\text{sil}} \sum_{i=1}^{n_{\text{sil}}} M^{\text{ISM}}_{i}(t)\,(1 + \Xi\, A_{\text{O}}/A_i), \tag{20}$$

where the n_{sil} elements referred to by index i are Mg, Si, S, Ca and Fe; M^{ISM}_{i} is the mass of element i in the ISM (including dust grains); A_i, its atomic mass; A_{O}, that of oxygen; and Ξ is the number of atoms of oxygen in silicate dust per atom of Mg, etc. Two assumptions are made in this equation:

- the same constant depletion factor $\delta^{\text{ISM}}_{\text{sil}}$ applies to each of the n_{sil} elements[※24];
- the number of oxygen atoms in the ISM is always sufficient to keep Ξ constant[※25].

β. Dwek's model

This model is based on Dwek (1998) and tries to describe the formation of dust in the late phases of stellar evolution, its destruction in the ISM by the blast waves of supernovae and the accretion of dust constituents on grains already present in the ISM.

i. Circumstellar dust production

Here, dust is produced in the following environments:

- **In the winds of mass-losing stars**. Carbon and oxygen atoms are assumed to be locked preferentially in CO molecules. If the number of carbon atoms in the overall[※26] ejecta of a star is larger than the number of oxygen atoms, only carbonaceous dust is produced. The mass of carbonaceous grains formed by a star is then

$$m_{\text{ej, carb}} = \delta^{\text{HMW/LMW}}_{\text{carb}}\,(m_{\text{ej, C}} - m_{\text{ej, O}}\,A_{\text{C}}/A_{\text{O}}), \tag{21}$$

  where $m_{\text{ej, O}}\,A_{\text{C}}/A_{\text{O}}$ is the mass locked in CO, $\delta^{\text{HMW}}_{\text{carb}}$ is the depletion factor in the winds of high-mass stars and $\delta^{\text{LMW}}_{\text{carb}}$ the one in low-mass stars.

  If the number of carbon atoms is less than that of oxygen ones, only silicate dust is produced. The mass of silicate dust produced by a star is then

$$m_{\text{ej, sil}} = \delta^{\text{HMW/LMW}}_{\text{sil}} \sum_{i=1}^{n_{\text{sil}}} m_{\text{ej},\,i}\,(1 + \Xi\, A_{\text{O}}/A_i), \tag{22}$$

  where i designates the same elements as in Eq. (20);
- **In the ejecta of supernovae**. No CO is formed (among other reasons, because C and O are in separate shells), so

$$m_{\text{ej, carb}} = \delta^{\text{CC SN/SN Ia}}_{\text{carb}}\, m_{\text{ej, C}}\,, \tag{23}$$

[24] The same assumption is made for the various δ_{sil} used in Dwek's model.

[25] The same assumption is made regarding dust accretion in Dwek's model. For circumstellar dust production, we assume that the number of free (not locked in CO) oxygen atoms in the environment is sufficient to keep Ξ constant.

[26] It would be more physical to compare the ejection *rates* $\dot{m}_{\text{ej, C}}$ and $\dot{m}_{\text{ej, C}}$ of carbon and oxygen rather than the integrated quantities $m_{\text{ej, C}}$ and $m_{\text{ej, C}}$…

Note also that, to compute the amount of dust produced in circumstellar environments, we need the gross yields for each stellar mass, not the net yields weighted by the number of dying stars; Eq. (18) is therefore not appropriate here.

where $\delta_{\mathrm{carb}}^{\mathrm{CC\,SN}}$ is the depletion factor in the ejecta of core-collapse supernovae and $\delta_{\mathrm{carb}}^{\mathrm{SN\,Ia}}$ the one in type Ia supernovae. Similarly,

$$m_{\mathrm{ej,\,sil}} = \delta_{\mathrm{sil}}^{\mathrm{CC\,SN/SN\,Ia}} \sum_{i=1}^{n_{\mathrm{sil}}} m_{\mathrm{ej},\,i}\,(1 + \Xi\,A_{\mathrm{O}}/A_i). \tag{24}$$

As for all other stellar ejecta, we assume that dust grains produced in circumstellar environments are formed instantaneously when the star dies. They are then immediately scattered through the whole ISM[27].

ii. Dust destruction in the ISM

Dust is also destroyed by supernovae. The destruction rate of the mass $M_{\mathrm{carb}}^{\mathrm{ISM}}(t)$ of carbonaceous dust in the ISM is computed as

$$\left(\frac{\mathrm{d}M_{\mathrm{carb}}^{\mathrm{ISM}}}{\mathrm{d}t}\right)_{\mathrm{destr}} = -\dot{n}_{\mathrm{SN}}(t)\,m_{\mathrm{swept}}\,\frac{M_{\mathrm{carb}}^{\mathrm{ISM}}(t)}{M_{\mathrm{ISM}}(t)}\,, \tag{25}$$

where $\dot{n}_{\mathrm{SN}}(t)$ is the number rate of supernovae, and the constant m_{swept} is the mass of ISM swept by the blast of a single supernova. Idem for silicate dust.

iii. Dust accretion in the ISM

The accretion rate of carbon on carbonaceous grains already in the ISM is modeled as

$$\left(\frac{\mathrm{d}M_{\mathrm{carb}}^{\mathrm{ISM}}}{\mathrm{d}t}\right)_{\mathrm{accr}} = \left(1 - \frac{M_{\mathrm{carb}}^{\mathrm{ISM}}(t)}{M_{\mathrm{C}}^{\mathrm{ISM}}(t)}\right)\frac{M_{\mathrm{carb}}^{\mathrm{ISM}}(t)}{\tau_{\mathrm{carb}}^{\mathrm{accr}}}\,, \tag{26}$$

where $\tau_{\mathrm{carb}}^{\mathrm{accr}}$ is a constant timescale and $1 - M_{\mathrm{carb}}^{\mathrm{ISM}}(t)/M_{\mathrm{C}}^{\mathrm{ISM}}(t)$ is the fraction of ISM carbon atoms in the gaseous phase, thus able to accrete on dust grains. Similarly, for silicates,

$$\left(\frac{\mathrm{d}M_{\mathrm{sil}}^{\mathrm{ISM}}}{\mathrm{d}t}\right)_{\mathrm{accr}} = \left(1 - \frac{M_{\mathrm{sil}}^{\mathrm{ISM}}(t)}{\sum_{i=1}^{n_{\mathrm{sil}}} M_i^{\mathrm{ISM}}(t)\,(1 + \Xi\,A_{\mathrm{O}}/A_i)}\right)\frac{M_{\mathrm{sil}}^{\mathrm{ISM}}(t)}{\tau_{\mathrm{sil}}^{\mathrm{accr}}}\,. \tag{27}$$

While the model where dust is produced in circumstellar environments, destroyed and accreted is more physical, it involves a lot of parameters, the value of which is rather uncertain. The basic model is therefore the default one. The parameters for these two models are listed in Sec. IV.3.*g*, p. 27.

c. Optical properties

The optical properties of dust species are computed as a function of wavelength from their size distributions (see Sec. II.5.*a*, p. 11) and from the optical properties of individual grains as follows. The absorption opacity in surface per unit mass (i.e. the mass absorption coefficient) of dust species i at wavelength λ is

$$\kappa_{i,\,\lambda}^{\mathrm{abs}} = \int_a \pi\,a^2\,Q_{i,\,\lambda}^{\mathrm{abs}}(a)\,\frac{\mathrm{d}n_i}{\mathrm{d}a}\,\mathrm{d}a, \tag{28}$$

where $\pi\,a^2\,Q_{i,\,\lambda}^{\mathrm{abs}}(a)$ is the absorption cross-section of a grain with radius a, and $\mathrm{d}n_i/\mathrm{d}a$ is the number of grains per unit size per unit mass of dust (i.e.,

$$\int_a \frac{4}{3}\,\pi\,a^3\,\varrho_i\,\frac{\mathrm{d}n_i}{\mathrm{d}a}\,\mathrm{d}a = 1, \tag{29}$$

where ϱ_i is the inner mass density of a grain). Similarly for the scattering opacity $\kappa_{i,\,\lambda}^{\mathrm{sca}}$ (with $Q_{i,\,\lambda}^{\mathrm{abs}}(a)$ replaced by $Q_{i,\,\lambda}^{\mathrm{sca}}(a)$). The extinction opacity is given by

$$\kappa_{i,\,\lambda}^{\mathrm{ext}} = \kappa_{i,\,\lambda}^{\mathrm{abs}} + \kappa_{i,\,\lambda}^{\mathrm{sca}}\,, \tag{30}$$

the albedo by

$$\omega_{i,\,\lambda} = \kappa_{i,\,\lambda}^{\mathrm{sca}}\big/\kappa_{i,\,\lambda}^{\mathrm{ext}} \tag{31}$$

and the asymmetry parameter by

$$g_{i,\,\lambda} = \frac{\int_a \pi\,a^2\,Q_{i,\,\lambda}^{\mathrm{sca}}(a)\,c_{i,\,\lambda}(a)\,(\mathrm{d}n_i/\mathrm{d}a)\,\mathrm{d}a}{\kappa_{i,\,\lambda}^{\mathrm{sca}}}\,, \tag{32}$$

[27] We therefore neglect the effect of circumstellar dust grains on the light of the stars they surround.

where $c_{i,\lambda}(a)$ is the average cosinus of the scattering angle. The quantities $Q^{\rm abs}_{i,\lambda}(a)$, $Q^{\rm sca}_{i,\lambda}(a)$ and $c_{i,\lambda}(a)$ are taken from Laor & Draine (1993), Draine & Lee (1984) and Li & Draine (2001) (files «`opt_prop_graphites.txt`» for graphites, «`opt_prop_silicates.txt`» for silicates, «`opt_prop_neutral_PAHs.txt`» for neutral PAHs and «`opt_prop_ionized_PAHs.txt`» for ionized ones, all in «`dust_dir/`»).

6. Attenuation by grains in the diffuse ISM

⟶ Sec. IV.3.*h*, p. 28, and App. E.8, p. 61.

The attenuation of the galaxy SED by grains in the diffuse medium is extensively described for ***geometries*** appropriate to spiral and spheroidal galaxies in sec. 4.1 and 4.2, respectively, of P3AA; all the notations used in Sec. II.6.*a* and II.6.*b* below are defined therein.

In addition to these two realistic geometries, a third one is still available for reasons of compatibility with previous versions of the code: the ***infinite slab*** in which stars and dust are well mixed.

See Sec. IV.3.*h*.γ, p. 28 to select the geometry.

a. Spiral galaxies

Grids of the transmittance for the spiral disk and bulge are provided in the files «`disk_transmit.txt`» and «`bulge_transmit.txt`» of directory «`RT_dir/`», both as a function of the viewing angle (inclination) toward the galaxy and averaged over all inclinations.

The default values of the bulge-to-total mass ratio Γ and of $M^{\rm spir}_{\rm sys}/R_{\rm d}^2$[28], where $M^{\rm spir}_{\rm sys}$ is the mass of the system and $R_{\rm d}$ is the characteristic radius of the dust disk, are given in Table 1. They are derived from the values provided in table I-2 of Binney & Tremaine (2008; BT) for the Milky Way and may be changed (see parameters ‹`bulge_tot_ratio`›, ‹`M_sys_spiral`› and ‹`expo_radius`›, Sec. IV.3.*h*, p. 28).

Table 1. Default structural parameters for spirals.

$\Gamma = 1/7$	From the values for L_V(disk) ($1.2 \times 10^{10}\,L_V(\odot)$) and L_V(bulge) ($2 \times 10^9\,L_V(\odot)$) in BT's model of the Milky Way.
$M^{\rm spir}_{\rm sys}/R_{\rm d}^2 = 2915\,M_\odot/{\rm pc}^2$	From Γ, the value of $R_{\rm d}/R_\star^{\rm disk}$ in table 1 of P3AA and the values for $M_{\rm tot}$(disk) ($6 \times 10^{10}\,M_\odot$) and $R_\star^{\rm disk}$ (3.5 kpc) in BT's model of the Milky Way.

b. Spheroidal galaxies

Grids of the transmittance for spheroidal galaxies are provided in file «`RT_dir/King_transmit.txt`».

For the mass of the system, $M^{\rm sph}_{\rm sys}$, we use the current mass of stars in model *b* of Tsai & Mathews (1995), neglecting thus the matter expelled by the galaxy in the intergalactic medium since it formed. The mass of the system is then

$$M^{\rm sph}_{\rm sys} = 4\,\pi\,R_{\rm c}^3\,\mu^{\rm sph}_{\star,0} \int_{s=0}^{s_{\rm t}} s^2\,(1+s^2)^{-3/2}\,{\rm d}s = 4\,\pi\,R_{\rm c}^3\,\mu^{\rm sph}_{\star,0} \left(\operatorname{arsinh} s_{\rm t} - s_{\rm t}/\sqrt{1+s_{\rm t}^2}\right), \tag{33}$$

where $s_{\rm t} := R_{\rm t}/R_{\rm c}$, $R_{\rm c}$ and $R_{\rm t}$ are the core and truncation radii, and $\mu^{\rm sph}_{\star,0}$ is the central stellar mass density. The default value of $M^{\rm sph}_{\rm sys}/R_{\rm c}^2$[28] is $7.73 \times 10^6\,M_\odot\,{\rm pc}^{-2}$; it may be changed (see parameters ‹`M_sys_spher`› and ‹`core_radius`›, Sec. IV.3.*h*, p. 28).

c. Infinite slab

The column mass density of dust through the slab and perpendicularly to it is derived from the column number density of hydrogen, computed as

$$\sigma^{\rm slab}_{\rm H} = \alpha_{\rm slab}\,\widehat{M}_{\rm ISM}. \tag{34}$$

Grids of the transmittance for this geometry are provided as a function of inclination in file «`RT_dir/slab_transmit.txt`». The default value of $\alpha_{\rm slab}$ may be changed (see parameter ‹`slab_factor`›, Sec. IV.3.*h*, p. 28).

[28] $M^{\rm spir}_{\rm sys}/R_{\rm d}^2$, $M^{\rm sph}_{\rm sys}/R_{\rm c}^2$ and $\alpha_{\rm slab}$ are the only unnormalized quantities in the code.

7. Dust emission by grains in the diffuse medium

⟶ Sec. IV.3.*h*, p. 28, and App. E.8, p. 61.

a. Interstellar radiation field

The computation of the mean interstellar radiation field in the diffuse ISM, $\langle u_\lambda \rangle$, is described in sec. 5.1 of P3AA. Note that, dividing the numerator and denominator of eq. P3AA (25) by M_{sys}, we may express $\langle u_\lambda \rangle$ exclusively in terms of the normalized unattenuated luminosity $\widehat{L}^0_\lambda$ and of the normalized dust mass $\widehat{M}_{\mathrm{d}}$ provided by the code:

$$\langle u_\lambda \rangle = \frac{\widehat{L}^0_\lambda \,(1 - \overline{\Theta}_\lambda)}{c\, \kappa^{\mathrm{abs}}_\lambda \, \widehat{M}_{\mathrm{d}}}\,, \tag{35}$$

where $\overline{\Theta}_\lambda$ is the inclination-averaged transmittance, c is the speed of light and $\kappa^{\mathrm{abs}}_\lambda$ is the absorption opacity per unit mass of dust. (Note, however, that $\overline{\Theta}_\lambda$ depends on the unnormalized optical depth $\tau^{\mathrm{ext}}_\lambda$ through the ratio $M^{\mathrm{spir}}_{\mathrm{sys}}/R^2_{\mathrm{d}}$ or $M^{\mathrm{sph}}_{\mathrm{sys}}/R^2_{\mathrm{c}}$ (Footnote 28, p. 14), so $\langle u_\lambda \rangle$ and the dust emission spectrum too.)

b. Stochastic heating

Because of their large cross section, big grains are permanently hit by a huge number of photons and always have temperatures close to their equilibrium value. On the other hand, small grains are seldom hit and undergo strong temperature fluctuations: they spend most of the time at low temperatures but sometimes reach very high temperatures for short durations. The computation of the temperature probability distribution of grains due to stochastic heating and of the resulting emission is explained in sec. 5.2 of P3AA. To switch off stochastic heating, see Sec. IV.3.*h*.ζ, p. 29, of this documentation.

As we do not evolve the size distribution of grains, we do not take into account the destruction of grains which may occur during temperature spikes. For information only, we however count separately the energy emitted by grains at temperatures larger than the ***sublimation temperature*** (one for all carbonaceous grains, one for silicates, whatever their size and species; see Sec. IV.3.*h*.η, p. 29 to change the sublimation temperatures).

c. Self-absorption

By default, we neglect ***self-absorption***, i.e. we assume that the ISM, whether the diffuse one or the star-forming clouds, is transparent to the emission of dust grains in the same medium[29]. The monochromatic luminosity leaving the region (the galaxy, if the region is the diffuse medium) is then

$$L^{-\textsc{sa}}_\lambda = \overline{\Theta}_\lambda\, L^0_\lambda + L^{\mathrm{d},-\textsc{sa}}_\lambda, \tag{36}$$

where $\overline{\Theta}_\lambda$ is the inclination-averaged transmittance, through dust in the medium, of the unattenuated luminosity L^0_λ (stellar, mainly), and $L^{\mathrm{d},-\textsc{sa}}_\lambda$ is the emission by dust if self-absorption is neglected.

To assess the effects of self-absorption, we propose to model it in the following way:

- The transmittance for dust-emitted photons is parametrized as

$$\Theta^{\mathrm{d}}_\lambda = \overline{\Theta}_\lambda^{\,\gamma}, \tag{37}$$

  where the power γ (parameter ‹`self_abs_power`›; see Sec. IV.3.*h*, p. 28) is a constant in $[0, 1]$. The default case, $\gamma = 0$, corresponds to no self-absorption ($\Theta^{\mathrm{d}}_\lambda = 1$). On the other hand, for very dusty environments, the mean free path of the original photons absorbed by dust is so small that one may consider that they are absorbed "on the spot"; in this approximation, dust-emitted photons are thus transmitted with the same factor $\overline{\Theta}_\lambda$ than original photons at the same wavelength, so taking $\gamma = 1$ should be appropriate[30];
- Moreover, since most of the photons re-emitted by grains are in the infrared, the fraction of this dust-emitted energy which is absorbed again by grains is much smaller than for the original pho-

[29] To be clearer, grains in the diffuse ISM (DISM) absorb and scatter the light from stars in the DISM, but also the light emitted and processed in star-forming clouds (SFCs), since these are embedded in the DISM. On the other hand, because SFCs are small compared to the DISM, grains in clouds process only the stellar and nebular photons emitted within SFCs, but not those coming from the DISM.

[30] Note that using $\gamma = 1$ for environments with little dust does not change the results by much, compared to the no self-absorption case: since $\overline{\Theta}_\lambda \approx 1$, one has then $\Theta^{\mathrm{d}}_\lambda \approx 1$ too, especially at the wavelengths where dust grains emit.

tons. We therefore assume that the temperature distribution of dust grains, and thus the *shape* of the dust-emitted SED, is unchanged by the subsequent processing.

With these assumptions, Eq. (36) becomes

$$L_\lambda^{+\text{SA}} = \overline{\Theta}_\lambda\, L_\lambda^0 + \alpha\, \overline{\Theta}_\lambda^{\,\gamma}\, L_\lambda^{\text{d},\,-\text{SA}}, \tag{38}$$

where $L_\lambda^{+\text{SA}}$ is the monochromatic luminosity, with "self-absorption", emerging from the region, and α is a constant (i.e., independent of wavelength) introduced to ensure the conservation of the energy: indeed, self-absorption redistributes the SED of dust grains at other wavelengths but conserves their overall bolometric luminosity. The factor α is thus fixed by the equation

$$\int \alpha\, \overline{\Theta}_\lambda^{\,\gamma}\, L_\lambda^{\text{d},\,-\text{SA}}\,\text{d}\lambda = \int L_\lambda^{\text{d},\,-\text{SA}}\,\text{d}\lambda. \tag{39}$$

See also App. D.3, p. 51.

8. Star-forming regions and nebular emission

⟶ Sec. IV.3.*i*, p. 29, and App. E.9, p. 63.

The modeling of star-forming regions and of the effects of dust on the stellar and nebular light they emit, in particular in the Lyman continuum, as well as the computation with version c17.01 of Cloudy (Ferland et al. 2017) of the nebular emission in star-forming clouds and in the diffuse ISM, are extensively described in sec. 6 and app. 1 of P3AA. Let us just remind that the fraction $\varphi(t')$ of stars aged t' still in their birth cloud is modeled as

$$\varphi(t') = \varphi_0\,(1 - t'/\theta)^\beta, \tag{40}$$

that all stars form in clusters and that all these clusters have the same initial stellar mass.

III. Computing the properties of single stellar populations with «SSPs»

Code «SSPs» computes as a function of age the isochrones of single stellar populations for several initial metallicities; it then apportions their light among the elements of a library of stellar spectra. The code also outputs the mass of compact stellar remnants produced by an SSP, the amount of matter ejected by stars into the ISM, the number of supernovae, and the radiation rate of Lyman continuum photons※[31].

To produce these outputs, «SSPs» needs the inputs described below.

1. Initial mass function

⟶ Eq. (7) and Eq. (8), p. 9, for the definition and normalization of the initial mass function ϕ.

The initial mass functions (IMF) mentioned in Table 2, p. 17, are provided with the code. The files associated to these IMFs are gathered in «IMFs_dir/» and are listed in «list_IMFs.txt»; the first IMF in the latter file is the default one. References are given in the headers of the IMF files.

The form of IMFs marked as "analytical" is actually not provided in the corresponding file, but is hard-coded in «source_dir/mod_IMF.f90». In particular, the log-normal IMF is given (see eq. (30) in Miller & Scalo (1979)) by

$$\phi(m) = C_0 \exp\Big(-C_1\,[\log_{10}(m/M_\odot) - C_2]^2\Big). \tag{41}$$

If you select this IMF when running «SSPs», you will be requested to enter values for C_1 and C_2 (C_0 is fixed by the normalization). You may also use the default values, $C_1 = 1.09$ and $C_2 = -1.02$, corresponding to Miller & Scalo (1979)'s fit of the solar neighborhood at 12 Gyr with a constant star formation rate.

All other IMFs are modeled as *continuous* piecewise power-law functions of m (i.e., for m in the bin $[m_i, m_{i+1}[$, $\phi(m) \propto m^{s_i}$, where the slope s_i is constant in the bin). In all cases, ϕ is automatically normalized by code «SSPs».

To add other IMFs, see App. C.1, p. 49.

When running «SSPs», it is possible to supersede the default values for the lower and upper masses of the IMF defined in «source_dir/mod_SSPs_constants.f90» or the «IMF_*.txt» files. Note also that the IMF used in «SSPs» relates only to *live stars*: it is possible to add a population of *inert* (i.e., dark and unevolving) *objects* to the evolutionary scenario of «spectra» via the parameter ‹SF_inert_frac› (see Sec. IV.3.*e*, p. 23).

[31] Although they are in text format, the files produced by «SSPs» are not aimed to be read by the user and do not contain spectral energy distributions. To compute these quantities for an SSP, first run «SSPs», then «spectra» for an episode of instantaneous star formation.

Table 2. Initial mass functions provided with the code.

File	Form of the IMF	Reference
«IMF_Rana_Basu.txt»	Analytical	Rana & Basu (1992).
«IMF_Chabrier_2003.txt»	Analytical	Chabrier (2003).
«IMF_Chabrier_2005.txt»	Analytical	Chabrier (2005).
«IMF_Ferrini.txt»	Analytical	Ferrini et al. (1990, p. 520).
«IMF_Kennicutt.txt»	Piecewise power-law	Kennicutt (1983).
«IMF_Kroupa.txt»	Piecewise power-law	Kroupa et al. (1993).
«IMF_log_normal.txt»	Analytical	Log-normal IMF (see below).
«IMF_Miller_Scalo.txt»	Piecewise power-law	Miller & Scalo (1979).
«IMF_Salpeter.txt»	Piecewise power-law	Salpeter (1955).
«IMF_Scalo86.txt»	Piecewise power-law	Scalo (1986).
«IMF_Scalo98.txt»	Piecewise power-law	Scalo (1998).
«IMF_Kroupa1.5.txt»	Piecewise power-law	Kroupa et al. (1993), but with a steeper slope at high mass.

As implemented, the IMF is independent of time. It is however possible to have an IMF evolving with the initial metallicity of SSPs, and thus indirectly with time. See App. C.2, p. 50, for this.

2. Stellar evolutionary tracks

In the code, ***files of stellar evolutionary tracks*** provide the evolution of the bolometric luminosity, effective temperature and surface gravity of a star as a function of its age for a range of initial masses and a single initial metallicity. ***Sets of stellar evolutionary tracks*** gather files with various metallicities. Sets of tracks are in «tracks_dir/» and are listed in «list_tracks_sets.txt»; the first set in this file is the default one.

The default set with current settings, «tracks_set+.txt», is based on the tracks computed by the Padova group in the 1990's and is described in sec. 2.2.2 of P3AA. At $Z = 0.1$, pseudo-tracks for masses larger than 9 $M_\odot$ have been computed from the corresponding masses in the $Z = 0.02$ and $Z = 0.05$ sets. For stars undergoing the helium flash, the zero-age main sequence (ZAMS) tracks are connected to the zero-age horizontal branch tracks with the same core mass, assuming a Reimers law (Reimers 1975) for the mass loss along the first giant branch with $\eta = 0.4$ (Renzini 1981).

Hydrogen burning post-AGB and CO white dwarf tracks from Bloecker (1995) ($m/M_\odot \in \{0.605,\ 0.625,\ 0.696,\ 0.836,\ 0.940\}$), Schoenberner (1983) ($m/M_\odot \in \{0.546,\ 0.565\}$; extrapolated with the 0.456 $M_\odot$-track from Koester & Schoenberner (1986)) and Paczyński (1971) ($m = 1.2\ M_\odot$) are then connected.

3. Stellar yields

⟶ Sec. II.4, p. 11.

The sets of yields we use are described in sec. 2.2.3 of P3AA.

The ejecta of low-mass stars are from Marigo (2001). The gross and net yields are given, respectively, in files «gross_ejecta_LMW.txt» and «net_ejecta_LMW.txt» of directory «yields_dir/», as are all the files mentioned in this section.

For high-mass stars, the default yields, from Portinari et al. (1998), are in the files «*_ejecta_HMW_P.txt» for the wind phase and «*_ejecta_CCSN_P.txt» for the core-collapse phase. The explosive yields of model B of Woosley & Weaver (1995) may also be selected; they are in the files «*_ejecta_CCSN_WW.txt» (the mass loss before the supernova is not taken into account in these).

To compute the net yields (files «net_*») from the gross ones (files «gross_*») or the reverse, we need the initial abundances in the star. These are provided for hydrogen, helium and all the metals together by Marigo (2001) and Portinari et al. (1998), but not for individual metals. For these, we scale the solar abundances of Anders & Grevesse (1989) (file «Anders_Grevesse_1989.txt») to Z.

We do the same for the metal yields of Woosley & Weaver (1995). For $Z < Z_\odot$, however, these authors do not provide the initial abundances of hydrogen and helium either. We thus compute the initial mass fraction of helium, Y, assuming a constant relative enrichment of helium with respect to metals:

$$\frac{Y - Y_{\text{prim}}}{Z - Z_{\text{prim}}} = \frac{Y_{\odot} - Y_{\text{prim}}}{Z_{\odot} - Z_{\text{prim}}}, \tag{42}$$

where $Y_{\text{prim}} = 0.23$ and $Z_{\text{prim}} \approx 0$ are the primordial abundances of helium and metals (just after the Big Bang), and the solar abundances $Y_{\odot}$ and $Z_{\odot}$ are taken from Anders & Grevesse (1989). The initial mass fraction X of hydrogen is then given by $X = 1 - Y - Z$.

The gross yields of model W7 of Thielemann et al. (1986), used to compute the ejecta of type Ia supernovae, are provided in file «`W7.txt`». The net yields are computed by «`SSPs`» with the same procedure as for Woosley & Weaver (1995).

4. Stellar spectra

In the code, ***files of stellar spectra*** provide the SEDs of stars as a function of wavelength for a range of effective temperatures and surface gravities and for a single initial metallicity. The emission rates of Lyman continuum photons, derived from these SEDs and used to model nebular emission, are also supplied. The SEDs and rates are normalized to the bolometric luminosity of the star.

Files of spectra are gathered in ***sets of stellar spectra*** to cover the Hertzsprung & Russell diagram at the metallicities possibly occurring during the evolution of a galaxy. All the sets are in «`stel_lib_dir/`» and are listed in «`list_stel_lib_sets.txt`»; the first set in the latter file is the default one («`BaSeL2.2_Rauch_¬ set1.txt`» currently).

Each set of stellar spectra is made of two libraries:

- BaSeL's spectra for stars with an effective temperature $T_{\text{eff}} < 50\,000$ K. The wavelength range is [91 Å, 160 µm], with a variable resolution decreasing from a few Å in the far-UV to 20 Å in the visible and to 20 µm in the far-IR/submm;
- the spectra from Rauch (2003), rebinned to the wavelengths of BaSeL, for hotter stars. These spectra are only available at [Fe/H] $\in \{-1, 0\}$.

The BaSeL library is based on theoretical spectra (Kurucz (1979), mostly). These are corrected with wavelength-dependent factors to fit observed colors. The library comes in two versions:

- v2.2 ※[32] (Lejeune et al. 1998), where the same correction factors, obtained from solar-metallicity calibrations of observed colors as a function of effective temperature and surface gravity, are applied to all metallicities;
- v3.1 WLBC99 (Westera et al. 2002), where correction factors are, somewhat indirectly, derived from fits to observed color - magnitude diagrams of star clusters with subsolar metallicities.

As recognized by Westera et al. (2002), when combined with theoretical isochrones, the WLBC99 library does not improve the fits to observed color - magnitude diagrams, compared to BaSeL-2.2. Because the latter covers a larger range of metallicities, we use it as our default.

5. Output ages of «SSPs»

The span in initial masses, phases and metallicities of the default set of stellar evolutionary tracks allows to follow the evolution of a galaxy from 0 to 20 Gyr.

The output ages of «`SSPs`» are determined from a file. The default file, «`SSPs_ages.txt`», has a time resolution degrading from 1 Myr for ages less than 30 Myr to 100 Myr above 10 Gyr. Other files may be used instead, to enhance the time resolution for instance, but all of them must be listed in «`ages_dir/¬ list_SSP_ages.txt`» and be located in the same directory. Look at the comments in «`SSP_ages.txt`» and «`list_SSP_ages.txt`» to define another file of output ages for «`SSPs`» and add it to the list. Note that the first number in the file of output ages is the ***convolution time-step***, i.e. the time-step used in «`spectra`» to evolve the system (in particular, to convolve the properties of SSPs with the star formation history).

[32] Several variants and combinations of the BaSeL-2.2 library are actually provided, for reasons explained in the headers of the «`BaSeL2.2_Rauch_set*.txt`» files.

IV. Defining scenarios of evolution for «spectra»

1. Preliminaries

a. Types of statements

A file of scenarios describes one or more scenarios and consists in a list of statements written following the syntax detailed in App. B.1, p. 47. There are two kinds of statements:

- **assignments**, i.e. all the statements which explicitely set (with "=") the values of the ***parameters*** defining a scenario.

  The basic form of an assignment is "*key* = *val*", where ‹*key*› is the name of a single scalar parameter and ‹*val*› its value. To assign the *k*-th element of an array of parameters ‹*key*›, use the syntax "*key*(*k*) = *val*". For the first element, you may more conveniently write "*key* = *val*" instead of "*key*(1) = *val*".

  For more details, in particular about complex assignments involving arrays of parameters, see App. B.2, p. 47;
- **commands**, i.e. all other statements.

  Some commands implicitely assign (without "=") parameters to their default values (the ‹reset_*› statements, for instance); other ones control the back and forth between the reading of scenarios and the computation of the evolution for the last read scenario.

  In particular, any scenario must be separated from the previous one in the file of scenarios by the command ‹return›: this returns control to the main procedure in «spectra», which then processes the current scenario and computes the corresponding evolution and spectra.

All the parameters and commands are described in Sec. IV.3.*b* to IV.5, p. 21 to 32.

b. General rules applying to parameters

Complete lists of the parameters, with their default value if any, are given in Tables 14 to 22, p. 53 to 65.

Related parameters are organized in trees (see Fig. 1 to 7, p. 53 to 64). Detailed explanations on how to read these trees are given at the beginning of App. E, p. 52.

Parameters obey two main rules:

1. **Explicitely setting the value of a parameter implicitely assigns all its "ancestors" to consistent values.**

   There is therefore no need to provide all the parameters in a tree.

   As an example, let us consider the tree related to star-forming clouds and nebular emission (see Fig. 6, p. 63). If you write "neb_emis_const_frac = 0.5" in the file of scenarios, then the parent of ‹neb_emis_const_frac›, ‹neb_emis_type›, is automatically set to ‹"constant"›, which in turn sets its grandparent, ‹nebular_emission›, to ‹.true.›. You still have to provide the values of other parameters (‹cloud_init_frac›, etc.), unless you are satisfied with their default values;
2. **Parameters which are not assigned in a scenario, whether explicitely or implicitely, retain the values they had in the previous scenario or remain undefined if they have no default value and have never been assigned.**

   This rule holds for all parameters※[33], including those not relevant for the current scenario.

   For example (see Fig. 1, p. 53), if ‹infall_expo_timescale› was defined in previous scenario (the initial value of this parameter is undefined), its value is irrelevant in the current scenario if ‹infall_type› is now set to ‹"constant"›, but will be used again in the next scenario if ‹infall_type› is then set to ‹"exponential"›.

If you are lost at some point, you can use one of the ‹reset_*› commands (see Sec. IV.5.*j*, p. 33), where ‹*› stands for ‹reserv_infall›, ‹SF›, etc., to reset all the parameters related to ‹*› to their default values, if they exist, or to undefined otherwise.

You can also write ‹echo› at any point in the file of scenarios to write the values of all the parameters at that point, both on the screen and in the log file (see Sec. I.3.*b*, p. 6).

2. An example of file of scenarios

Before we describe in detail how to write a file of scenarios and the meaning of all parameters, let us look at an example, file «example_scenarios.txt» (available in «scenarios_dir/»). This file contains three

[33] The only exceptions are the names of the output files of «spectra» for the current scenario (‹spectra_file› and, if required, ‹grain_temp_file› and ‹grain_SED_file›).

scenarios.

a. First scenario (instantaneous burst of star formation)

```
1  SSPs_set = "example_SSPs.txt"
2
3  ! Instantaneous burst of star formation:
4  SF_type = "instantaneous"
5  spectra_file = "example_spectra1.txt"
6  return
```

Explanations:

- **Line 1**: the IMF, yields, evolutionary tracks and stellar spectra defined in «example_SSPs.txt» will be used;
- **Lines 2 and 3**: skipped (blank lines and comments);
- **Line 4**: the scenario consists in a single instantaneous burst of star formation;
- **Line 5**: the name of the file of spectra produced for this scenario is «*example_spectra1.txt*»;
- **Line 6**: reading of the scenario is done. Return to «spectra» and compute the evolution of the modeled galaxy for the first scenario.

All other parameters have their default values: the whole mass is initially in the galaxy and in the form of zero-metallicity gas; the burst occurs at age 0 and consumes all this gas to form zero-metallicity stars; there is neither infall nor outflow; nebular emission and dust effects are not considered.

b. Second scenario (≈ spiral galaxy)

The parameters of the second scenario are defined by the following lines:

```
 8  ! Spiral galaxy:
 9  reserv_init_mass = 1.
10  infall_expo_timescale = 1000
11  SF_ISM_timescale = 3000
12  nebular_emission = .true.
13  extinction = .true.
14  inclin_averaged = .true.
15  dust_emission = .true.
16  spectra_file = "example_spectra2.txt"
17  return
```

Explanations:

- **Line 9**: the galaxy accretes gas from a reservoir (the first one by default) with a mass ‹reserv_init_mass› × M_{sys}. As ‹reserv_Z› = 0 by default and ‹reserv_init_mass› = 1, the galaxy forms entirely from the zero-metallicity gas in the reservoir;
- **Line 10**: this explicitely sets ‹infall_expo_timescale› to 1000 Myr and, implicitely, all the parameters above it, along the path to the top of the tree, to consistent values: ‹infall_type› is therefore set ‹"exponential"›; parameters in other branches, e.g. ‹infall_begin_time›, are untouched and keep their value if it has been defined previously, explicitely or implicitely, or if there is a default one. Consequently, the infall rate is exponentially decreasing with a timescale of 1 Gyr from 0 to 20 Gyr;
- **Line 11**: this sets the star formation rate to proportional to the mass of ISM with a timescale of 3 Gyr. The implicit ‹SF_type› = ‹"ISM_mass"› activated by "SF_ISM_timescale = 3000" supersedes the "SF_type = "instantaneous"" of the previous scenario;
- **Line 12**: nebular emission will be computed;
- **Line 13**: extinction by dust is taken into account;
- **Line 14**: dust attenuation for a disk galaxy, averaged over all inclinations, is applied to the SEDs;
- **Line 15**: dust emission spectrum is added.

Except ‹spectra_file›, all the parameters defined previously in the file, e.g. ‹SSPs_set›, retain their value, unless modified explicitely or implicitely.

c. Third scenario (spiral galaxy + late burst)

```
19 ! Spiral galaxy with a late burst:
20 SF_type(2) = "constant"
21 SF_const_mass(2) = 1.e-1
22 SF_begin_time(2) = 10000
23 SF_end_time(2) = 11000
24 spectra_file = "example_spectra3.txt"
```

These lines define a second star formation episode occuring from 10 Gyr (line 22) to 11 Gyr (line 23), with a constant rate (line 20) and involving a mass of 0.1 M_{sys} (line 21). This episode is *added* to the one defined by scenario number 2. Note that the first episode of star formation could have been defined with the statement "`SF_ISM_timescale(1) = 3000`" instead of "`SF_ISM_timescale = 3000`" on line 11.

3. Description of main parameters

Default values are indicated for some parameters only. For others, see App. E, p. 52.

a. Parameters related to cosmology

⟶ App. E.2, p. 52.

α. **‹Omega_m›**, **‹H_0›**, **‹form_redshift›**

‹Omega_m› is the current value of the ratio Ω_{m} of the mean density of matter in the Universe to the critical density of the latter; ‹H_0› is the current value of the Hubble constant in km s^{-1} Mpc^{-1}.

These cosmological parameters are used to compute the redshift of the galaxy as a function of its age and of ‹form_redshift›, its formation redshift (Peebles 1993, p. 317). A flat universe with a cosmological constant Ω_Λ is assumed in this calculation ($\Omega_{\mathrm{m}} + \Omega_\Lambda = 1$); the density of radiation is neglected with regard to that of matter. Redshifts are printed in the main output file of «spectra».

β. **‹CBR›**

The redshift is also used to compute the temperature of the cosmic blackbody. The heating of dust by the cosmic background radiation field is considered, in addition to the radiation field emitted by the galaxy, if and only if ‹CBR› is ‹.true.›.

b. Single stellar populations: **‹SSPs_set›**

⟶ Sec. I.3.*a*, p. 5, and App. E.3, p. 53.

‹SSPs_set› holds the name of the set of SSPs[34], the master file produced by «SSPs». (This file points to auxiliary files containing the properties of SSPs with a single initial metallicity —one per file of evolutionary tracks.)

c. Chemical evolution parameters: **‹ISM_init_Z›**, **‹close_bin_frac›**

⟶ Sec. II.3, p. 10, and App. E.3, p. 53.

‹ISM_init_Z› is the initial value of the metallicity of the interstellar medium in the galaxy.

‹close_bin_frac› is the fraction of close binaries[34]. This parameter is used to compute the rate and ejecta of type Ia supernovae.

[34] For computational reasons, it is recommended to set the parameters ‹SSPs_set›, ‹close_bin_frac› and ‹grains_file› (or ‹grains_file_SFC› and ‹grains_sizes_DISM›; see Sec. IV.3.*h*.*β*, p. 28) at most once, in the first scenario. Note that ‹SSPs_set› must be provided at least one since there is no default value for this parameter.

d. Reservoir and infall parameters

⟶ Sec. II.1.*a*, p. 8, and App. E.4, p. 53.

α. **‹reserv_init_mass(*j*)›**, **‹reserv_Z(*j*)›**

‹`reserv_init_mass(`*j*`)`› is the initial normalized mass $\widehat{M}_{\mathrm{res},\,j}$ of reservoir j[35]; ‹`reserv_Z(`*j*`)`› is its constant metallicity.

The normalized initial mass of the galaxy is computed as

$$\widehat{M}_{\mathrm{gal}}(t=0) = 1 - \sum_j \widehat{M}_{\mathrm{res},\,j}(t=0). \tag{43}$$

By default, $\widehat{M}_{\mathrm{res},\,j}(t=0) = 0$ for all reservoirs, so $\widehat{M}_{\mathrm{gal}}(t=0) = 1$.

If the values of the parameters ‹`reserv_init_mass(`*j*`)`› are such that $\sum_j \widehat{M}_{\mathrm{res},\,j}(t=0) > 1$, then $\widehat{M}_{\mathrm{gal}}(t=0)$ is set to 0, and the values used (*in this scenario only*) for the $\widehat{M}_{\mathrm{res},\,j}(t=0)$ are scaled by a common factor to ensure this[36].

β. Infall episodes, **‹infall_source(*k*)›**

More than one episode of infall[35] on the galaxy can occur, and this from any reservoir. The parameter ‹`infall_source(`*k*`)`› is the index of the reservoir from which the infalling gas in episode *k* comes. By default, all infall episodes remove gas from the first reservoir.

The mass removal rate from reservoir *j* at age *t* is

$$\dot{M}_{\mathrm{res},\,j}(t) = -\sum_{\substack{\text{all } k \text{ such}\\ \text{that } r_k=j}} \dot{M}_{\mathrm{in},\,k}(t), \tag{44}$$

where $\dot{M}_{\mathrm{in},\,k}(t)$ is the infall rate of episode *k*, and r_k is the value of ‹`infall_source(`*k*`)`›.

If the mass removed from reservoir *j* in one convolution time-step, $-\dot{M}_{\mathrm{res},\,j}(t)\,\Delta t$, is larger than $M_{\mathrm{res},\,j}(t)$, the infall rates of all the episodes drawing from reservoir *j* are scaled down at *t* by the same factor so that $-\dot{M}_{\mathrm{res},\,j}(t)\,\Delta t = M_{\mathrm{res},\,j}(t)$[36].

For each *k*, the value of $\dot{M}_{\mathrm{in},\,k}(t)$ is computed using the values of the parameters defined in the following paragraphs.

γ. **‹infall_begin_time(*k*)›**, **‹infall_end_time(*k*)›**

Whatever the value of $\dot{M}_{\mathrm{in},\,k}(t)$ computed using other parameters,

$$\forall\ t \notin [t_{\mathrm{b},\,k}, t_{\mathrm{e},\,k}[, \quad \dot{M}_{\mathrm{in},\,k}(t) = 0, \tag{45}$$

where $t_{\mathrm{b},\,k}$ and $t_{\mathrm{e},\,k}$ are the values of ‹`infall_begin_time(`*k*`)`› and ‹`infall_end_time(`*k*`)`›.

δ. **‹infall_type(*k*)› = ‹"none"›**

$$\forall\ t, \quad \widehat{\dot{M}}_{\mathrm{in},\,k}(t) = 0. \tag{46}$$

ε. **‹infall_type(*k*)› = ‹"instantaneous"›: ‹infall_inst_mass(*k*)›**

$$\forall\ t \in [t_{\mathrm{b},\,k}, t_{\mathrm{e},\,k}[, \quad \widehat{\dot{M}}_{\mathrm{in},\,k}(t) = f_k\,\delta(t - t_{\mathrm{b},\,k}), \tag{47}$$

where f_k is the value of ‹`infall_inst_mass(`*k*`)`›, $t_{\mathrm{b},\,k}$ is that of ‹`infall_begin_time(`*k*`)`›, and δ is the Dirac distribution. (In practice, the infall event occurs at the nearest convolution time.)

[35] The maximal numbers of reservoirs and of infall, star formation and outflow episodes are set (to 10 by default) in «`source_¬ dir/mod_spectra_constants.f90`» by parameters ‹`max_dim_reserv`›, ‹`max_dim_infall_epis`›, ‹`max_dim_SF_epis`› and ‹`max_dim_¬ outflow_epis`›, respectively. If needed, change these values and recompile the code with ‹`make`› (see Sec. I.2, p. 4).

[36] A warning will be printed at the end of the file of spectra, in the log file and, if ‹`verbosity`› ⩾ 0 (see Sec. IV.5.*g*, p. 32), on the screen. See also Table 10, p. 39.

ζ. **‹infall_type(k)› = ‹"constant"›: ‹infall_const_mass(k)›**

$$\forall\ t \in [t_{\mathrm{b},k}, t_{\mathrm{e},k}[,\quad \widehat{\dot{M}}_{\mathrm{in},k}(t) = \frac{f_k}{t_{\mathrm{e},k} - t_{\mathrm{b},k}}\,, \tag{48}$$

where f_k is the value of ‹infall_const_mass(k)›.

η. **‹infall_type(k)› = ‹"exponential"›: ‹infall_expo_timescale(k)›, ‹infall_expo_mass(k)›**

$$\forall\ t \in [t_{\mathrm{b},k}, t_{\mathrm{e},k}[,\quad \widehat{\dot{M}}_{\mathrm{in},k}(t) = \frac{f_k}{|\tau_k|}\,\mathrm{e}^{-(t-t_{\mathrm{b},k})/\tau_k}, \tag{49}$$

where f_k is the value of ‹infall_expo_mass(k)› and τ_k that of ‹infall_expo_timescale(k)›. A positive value of τ_k corresponds to a decreasing infall rate; a negative value, to an increasing one.

θ. **‹infall_type(k)› = ‹"reserv_mass"›: ‹infall_reserv_timescale(k)›, ‹infall_reserv_power(k)›**

$$\forall\ t \in [t_{\mathrm{b},k}, t_{\mathrm{e},k}[,\quad \widehat{\dot{M}}_{\mathrm{in},k}(t) = \frac{\left(\widehat{M}_{\mathrm{res},\,j}[t]\right)^{\alpha_k}}{\tau_k}\,, \tag{50}$$

where j = ‹infall_source(k)›, $\widehat{M}_{\mathrm{res},\,j}(t)$ is the normalized mass of reservoir j at age t, τ_k is the value of ‹infall_reserv_timescale(k)›, and α_k is that of ‹infall_reserv_power(k)›.

ι. **‹infall_type(k)› = ‹"file"›: ‹infall_file(k)›**

For all $t \in [t_{\mathrm{b},k}, t_{\mathrm{e},k}[$, $\widehat{\dot{M}}_{\mathrm{in},k}(t)$ is interpolated from the values read in a file. The name of the latter is stored in parameter ‹infall_file(k)›[※37].

The first lines of the file may be blank lines or comments beginning with a “ ! ”. On the following lines, the first column will be the age in Myr, and the second one, the normalized infall rate at that time.

e. Star formation parameters

⟶ Sec. II.3, p. 10, and App. E.5, p. 55.

More than one episode of star formation[※35] can occur. Each episode is characterized by its rate at age t, $\psi_k(t)$. The total star formation rate is

$$\psi(t) = \sum_k \psi_k(t). \tag{51}$$

For each k, the value of $\psi_k(t)$ is computed using the values of the parameters defined in this section.

If the mass of stars formed in one convolution time-step, $\psi(t)\,\Delta t$, is larger than the mass of the ISM at that age, $M_{\text{ISM}}(t)$, all the $\psi_k(t)$ are scaled down by the same factor so that $\psi(t)\,\Delta t = M_{\text{ISM}}(t)$[※36].

α. **‹SF_begin_time(k)›, ‹SF_end_time(k)›**

Whatever the value of $\psi_k(t)$ computed using other parameters,

$$\forall\ t \notin [t_{\mathrm{b},k}, t_{\mathrm{e},k}[,\quad \psi_k(t) = 0, \tag{52}$$

where $t_{\mathrm{b},k}$ and $t_{\mathrm{e},k}$ are the values of ‹SF_begin_time(k)› and ‹SF_end_time(k)›.

β. **‹SF_type(k)› = ‹"none"›**

$$\forall\ t,\quad \widehat{\psi_k}(t) = 0. \tag{53}$$

γ. **‹SF_type(k)› = ‹"instantaneous"›: ‹SF_inst_mass(k)›**

$$\widehat{\psi_k}(t) = f_k\,\delta(t - t_{\mathrm{b},k}), \tag{54}$$

where f_k is the value of ‹SF_inst_mass(k)›, $t_{\mathrm{b},k}$ is that of ‹SF_begin_time(k)›, and δ is the Dirac distribution. (In practice, the star formation event occurs at the nearest convolution time.)

[37] If this file is not in «scenarios_dir/», its absolute path or its path relative to «scenarios_dir/» must be given.

δ. ‹SF_type(k)› = ‹"constant"›: ‹SF_const_mass(k)›

$$\forall\ t \in [t_{\mathrm{b},k}, t_{\mathrm{e},k}[,\quad \widehat{\psi}_k(t) = \frac{f_k}{t_{\mathrm{e},k} - t_{\mathrm{b},k}}\ , \tag{55}$$

where f_k is the value of ‹SF_const_mass(k)›.

ε. ‹SF_type(k)› = ‹"exponential"›: ‹SF_expo_timescale(k)›, ‹SF_expo_mass(k)›

$$\forall\ t \in [t_{\mathrm{b},k}, t_{\mathrm{e},k}[,\quad \widehat{\psi}_k(t) = \frac{f_k}{|\tau_k|}\exp\left(-\frac{t - t_{\mathrm{b},k}}{\tau_k}\right), \tag{56}$$

where f_k is the value of ‹SF_expo_mass(k)› and τ_k that of ‹SF_expo_timescale(k)›. A positive value of τ_k corresponds to a decreasing star formation rate; a negative value, to an increasing one.

ζ. ‹SF_type(k)› = ‹"peaked"›: ‹SF_peaked_timescale(k)›, ‹SF_peaked_mass(k)›

$$\forall\ t \in [t_{\mathrm{b},k}, t_{\mathrm{e},k}[,\quad \widehat{\psi}_k(t) = f_k\,\frac{(t - t_{\mathrm{b},k})}{\tau_k^2}\exp\left(-\frac{t - t_{\mathrm{b},k}}{\tau_k}\right), \tag{57}$$

where f_k is the value of ‹SF_peaked_mass(k)› and τ_k that of ‹SF_peaked_timescale(k)›. (The star formation rate increases during τ_k after $t_{\mathrm{b},l}$, and then decreases with a timescale of τ_k, as in Sandage (1986).)

η. ‹SF_type(k)› = ‹"ISM_mass"›: ‹SF_ISM_timescale(k)›, ‹SF_ISM_power(k)›, ‹SF_ISM_threshold(k)› (≋ Schmidt-Kennicutt law)

$$\forall\ t \in [t_{\mathrm{b},k}, t_{\mathrm{e},k}[,\quad \widehat{\psi}_k(t) = \frac{\left(\widehat{M}_{\mathrm{ISM}}[t] - \sigma_k\right)^{\alpha_k}}{\tau_k}\ , \tag{58}$$

where $\widehat{M}_{\mathrm{ISM}}(t)$ is the normalized mass of the interstellar medium at age t, τ_k is the value of ‹SF_ISM_timescale(k)›, α_k that of ‹SF_ISM_power(k)› and σ_k that of ‹SF_ISM_threshold(k)›. If $\widehat{M}_{\mathrm{ISM}}(t) < \sigma_k$, $\widehat{\psi}_k(t) = 0$.

θ. ‹SF_type(k)› = ‹"infall"›: ‹SF_infall_factor(k)›

$$\forall\ t \in [t_{\mathrm{b},k}, t_{\mathrm{e},k}[,\quad \widehat{\psi}_k(t) = f_k\,\widehat{\dot{M}}_{\mathrm{in}}(t), \tag{59}$$

where f_k is the value of ‹SF_infall_factor(k)›.

ι. ‹SF_type(k)› = ‹"file"›: ‹SF_file(k)›

For all $t \in [t_{\mathrm{b},k}, t_{\mathrm{e},k}[$, $\widehat{\psi}_k(t)$ is interpolated from the values read in a file. The name of the latter is stored in parameter ‹SF_file(k)›[37].

The first lines of the file may be blank lines or comments beginning with a “ ! ”. On the following lines, the first column will be the age in Myr, and the second one, the normalized star formation rate at that time.

κ. ‹SF_stochastic(k)›: ‹SF_stoch_fluc(k)›, ‹SF_stoch_timescale(k)›

If the value of ‹SF_stochastic(k)› is ‹.true.›, the star formation rate of the k-th episode is stochastically modulated by a log-normal function:

$$\psi_k(t) = \exp(\sigma_k\,\xi - \sigma_k^2/2)\,\psi_{k,0}(t), \tag{60}$$

where

$$\sigma_k = \sqrt{\ln(1 + \text{‹SF\_stoch\_fluc}(k)\text{›}^2)} \tag{61}$$

and $\psi_{k,0}$ is the star formation rate in the absence of stochastic modulation. This expression ensures that $\psi_k(t) \geqslant 0$, that the expectation of $\psi_k(t)$ is $\psi_{k,0}(t)$ and that the standard deviation of $\psi_k(t)$ is ‹SF_stoch_fluc(k)› × $\psi_{k,0}(t)$. The number ξ is a Gaussian random deviate with mean 0 and variance 1; it is constant during a duration drawn from a Poissonian distribution of mean given by‹SF_stoch_timescale(k)›[38].

[38] See Sec. IV.4, p. 31, to define the sequence of random numbers used for this purpose.

λ. ‹**SF_Z_type(*k*)**› = ‹**"consistent"**›

The metallicity of new stars formed in episode k is the same as that of the ISM.

μ. ‹**SF_Z_type(*k*)**› = ‹**"constant"**›: ‹**SF_Z_const_val(*k*)**›

The metallicity of new stars formed in episode k is a constant given by the value ‹SF_Z_const_val(*k*)›.

ν. ‹**SF_Z_type(*k*)**› = ‹**"file"**›: ‹**SF_Z_file(*k*)**›

The metallicity of new stars formed in episode k is interpolated from the values read in a file. The name of the latter is stored in parameter ‹SF_Z_file(*k*)›※37.

The first lines of the file may be blank lines or comments beginning with a "!". On the following lines, the first column will be the age in Myr, and the second one, the metallicity at that time.

ξ. ‹**SF_inert_frac(*k*)**›

For the purpose of the code, inert objects are objects, such as brown dwarfs or planets, which formed from the ISM with live stars but are not taken into account in the IMF provided to «SSPs»※39. Contrary to live stars, inert objects are assumed to be dark and unevolving since their birth: they just lock mass. The formation rate in episode k of inert objects is $f_k\,\psi_k(t)$, where f_k is the value of ‹SF_inert_frac(*k*)›. This parameter changes the mass-to-light ratio and the chemical evolution. (The formation rate of live stars is $(1 - f_k)\,\psi_k(t)$.)

f. Outflow parameters

⟶ Sec. II.1.*a*, p. 8, and App. E.6, p. 57.

More than one episode of outflow※35 from the galaxy into the intergalactic medium can occur. Each episode is characterized by its rate at age t, $\dot{M}_{\text{out},\,k}(t)$. Except as mentioned in Sec. IV.3.*f*.γ, the total outflow rate is

$$\dot{M}_{\text{out}}(t) = \sum_k \dot{M}_{\text{out},\,k}(t). \tag{62}$$

For each k, the value of $\dot{M}_{\text{out},\,k}(t)$ is computed using the values of the parameters defined in this section.

If the mass expelled from the galaxy in one convolution time-step, $\dot{M}_{\text{out}}(t)\,\Delta t$, is larger than the mass $M_{\text{ISM}}(t)$ of the ISM at that age, all the $\dot{M}_{\text{out},\,k}(t)$ are scaled down by the same factor so that $\dot{M}_{\text{out}}(t)\,\Delta t = M_{\text{ISM}}(t)$※36.

α. ‹**outflow_begin_time(*k*)**›, ‹**outflow_end_time(*k*)**›

Whatever the value of $\dot{M}_{\text{out},\,k}(t)$ computed using other parameters,

$$\forall\ t \notin [t_{\text{b},\,k}, t_{\text{e},\,k}[, \quad \dot{M}_{\text{out},\,k}(t) = 0, \tag{63}$$

where $t_{\text{b},\,k}$ and $t_{\text{e},\,k}$ are the values of ‹outflow_begin_time(*k*)› and ‹outflow_end_time(*k*)›.

β. ‹**outflow_type(*k*)**› = ‹**"none"**›

$$\forall\ t, \quad \dot{M}_{\text{out},\,k}(t) = 0. \tag{64}$$

γ. ‹**outflow_type(*k*)**› = ‹**"radical"**›

For all $t \in [t_{\text{b},\,k}, t_{\text{e},\,k}[$, all the ISM present in the galaxy is instantaneously expelled.

δ. ‹**outflow_type(*k*)**› = ‹**"instantaneous"**›: ‹**outflow_inst_mass(*k*)**›

$$\forall\ t \in [t_{\text{b},\,k}, t_{\text{e},\,k}[, \quad \overset{\frown}{M}_{\text{out},\,k}(t) = f_k\,\delta(t - t_{\text{b},\,k}), \tag{65}$$

where f_k is the value of ‹outflow_inst_mass(*k*)›, $t_{\text{b},\,k}$ is that of ‹outflow_begin_time(*k*)›, and δ is the Dirac distribution. (In practice, the outflow event occurs at the nearest convolution time.)

[39] Note that objects already included in the IMF are considered as live stars even if their mass is in fact lower than the upper mass for a brown dwarf.

ε. ‹outflow_type(*k*)› = ‹"constant"›: ‹outflow_const_mass(*k*)›

$$\forall\ t \in [t_{\mathrm{b},k}, t_{\mathrm{e},k}[, \quad \widehat{\dot{M}}_{\mathrm{out},k}(t) = \frac{f_k}{t_{\mathrm{e},k} - t_{\mathrm{b},k}}\ , \tag{66}$$

where f_k is the value of ‹outflow_const_mass(*k*)›.

ζ. ‹outflow_type(*k*)› = ‹"SF"›: ‹outflow_SF_factor(*k*)›, ‹outflow_SF_power(*k*)›

$$\forall\ t \in [t_{\mathrm{b},k}, t_{\mathrm{e},k}[, \quad \dot{M}_{\mathrm{out},k}(t) = f_k\, \frac{\psi(t)}{\left(\widehat{M}_{\text{ISM}}[t]\right)^{\alpha_k}}\ , \tag{67}$$

where f_k is the value of ‹outflow_SF_factor(*k*)› and α_k that of ‹outflow_SF_power(*k*)› ($\alpha_k = 0$ by default[40], as in most simulations (Dalla Vecchia & Schaye 2008), but Sharma & Nath (2012) recommend to take $\alpha_k \approx 1$). This rate may be appropriate if outflows in the IGM are due not only to supernovae but also to the stellar winds of high-mass stars.

η. ‹outflow_type(*k*)› = ‹"SN"›: ‹outflow_SN_mass(*k*)›, ‹outflow_SN_power(*k*)› /
‹outflow_type(*k*)› = ‹"CCSN"›: ‹outflow_CCSN_mass(*k*)›, ‹outflow_CCSN_power(*k*)› /
‹outflow_type(*k*)› = ‹"SNIa"›: ‹outflow_SNIa_mass(*k*)›, ‹outflow_SNIa_power(*k*)›

If ‹outflow_type(*k*)› = ‹"SN"›, then

$$\forall\ t \in [t_{\mathrm{b},k}, t_{\mathrm{e},k}[, \quad \dot{M}_{\mathrm{out},k}(t) = f_k\, \frac{\dot{n}_{\text{SN}}(t)}{\left(\widehat{M}_{\text{ISM}}[t]\right)^{\alpha_k}}\ , \tag{68}$$

where $\dot{n}_{\text{SN}}$ is the number rate of supernovae (both core-collapse and type Ia), f_k = ‹outflow_SN_mass(*k*)› is the *unnormalized* mass of ISM expelled from the galaxy by one supernova and α_k is the value of ‹outflow_SN_power(*k*)›.

With "CCSN" (resp. "SNIa") instead of "SN" in the value of ‹outflow_type› and the names of parameters ‹outflow_SN_mass› and ‹outflow_SN_power›, the number rate of core-collapse (resp. type Ia) supernovae replaces the number rate of all supernova types in Eq. (68).

θ. ‹outflow_type(*k*)› = ‹"ejecta"›: ‹outflow_ejec_threshold(*k*)›, ‹outflow_ejec_factor(*k*)›

$$\forall\ t \in [t_{\mathrm{b},k}, t_{\mathrm{e},k}[, \quad \dot{M}_{\mathrm{out},k}(t) = f_k \max\left(0, 1 - \frac{M_{\text{ISM}}(t)}{\sigma_k \int_{t'=0}^{t} \psi(t')\,\mathrm{d}t'}\right) \dot{M}_{\mathrm{ej}}(t), \tag{69}$$

where f_k is the value of ‹outflow_ejec_factor(*k*)›, σ_k that of ‹outflow_ejec_threshold(*k*)› and $\dot{M}_{\mathrm{ej}}$ is the mass ejection rate of stars in the ISM.

Note that, although the outflow rate is related to the mass ejection rate of stars in the ISM, the outflow affects all the matter in the ISM, not only stellar ejecta.

In early-type galaxies, old low-mass stars eject enough matter in the ISM to feed a residual star formation at late ages, and this with a rate nearly independent from the gas-to-stars conversion factor if the mass of the ISM decreases (see sec. 3.2.1 in Fioc & Rocca-Volmerange (1997) and chap. 3, sec. 4 in Fioc (1997)). Conversely, if stellar ejecta are not used to form stars, they accumulate in the galaxy and cool down. As neither substantial star formation nor large amounts of cold gas are observed in early-type galaxies, most of the late stellar ejecta must be expelled in the IGM through galactic winds (or be heaten up)[41]. The purely phenomenological modeling implemented through Eq. (69), inspired from Larson (1974)[42], is aimed to rid the galaxy from its ISM, and thus to starve star formation, when the ISM content drops below some threshold (typically $\lesssim 0.1$): the lower the threshold, the later the age when "galactic winds" occur and the larger the mean stellar metallicity. A more recent paper by Conroy et al. (2015) also proposes that the ejecta of dying low-mass stars prevent residual star formation in quiescent galaxies.

[40] So, $\dot{M}_{\mathrm{out},k}(t) = f_k\,\psi(t)$, and this, with the convention $0^0 = 1$, even if $M_{\text{ISM}}(t) = 0$.

[41] Another possibility is that only low-mass stars are formed (Fabian et al. 1984).

[42] We however use the mass of stars ever formed at t, $\int_{t'=0}^{t} \psi(t')\,\mathrm{d}t'$, instead of the mass in stars.

ι. **‹outflow_type(*k*)› = ‹"file"›: ‹outflow_file(*k*)›**

For all $t \in [t_{\mathrm{b},k}, t_{\mathrm{e},k}[$, $\widehat{\dot{M}}_{\mathrm{out},k}(t)$ is interpolated from the values read in a file. The name of the latter is stored in parameter ‹outflow_file(*k*)›[※37].

The first lines of the file may be blank lines or comments beginning with a “ ! ”. On the following lines, the first column will be the age in Myr, and the second one, the normalized outflow rate at that time.

g. Dust evolution parameters

⟶ App. E.7, p. 59.

The meaning of the code parameters used to model the evolution of dust is given in Table 3, p. 27.

α. **‹dust_evolution› = ‹"basic"›: ‹ISM_carb_deplet›, ‹ISM_sil_deplet›**

If ‹dust_evolution› = ‹"basic"›, the dust evolution model described in Sec. II.5.*b*.*α*, p. 12, is used.

β. **‹dust_evolution› = ‹"Dwek"›: ‹HMW_carb_deplet›, ‹LMW_carb_deplet›, ‹CCSN_carb_deplet›, ‹SNIa_carb_deplet›, ‹HMW_sil_deplet›, ‹LMW_sil_deplet›, ‹CCSN_sil_deplet›, ‹SNIa_sil_deplet›, ‹SN_swept_mass›, ‹carb_accr_timescale›, ‹sil_accr_timescale›**

If ‹dust_evolution› = ‹"Dwek"›, the dust evolution model described in Sec. II.5.*b*.*β*, p. 12 is used.

γ. **‹O_sil_ratio›**

Common to both models of dust evolution.

Table 3. Meaning of code parameters for dust evolution models.

Code parameter	Notation used in the modeling	Equations where the quantity appears
If ‹dust_evolution› = ‹"basic"›:		
‹ISM_carb_deplet›	$\delta^{\mathrm{ISM}}_{\mathrm{carb}}$	Eq. (19), p. 12;
‹ISM_sil_deplet›	$\delta^{\mathrm{ISM}}_{\mathrm{sil}}$	Eq. (20), p. 12.
If ‹dust_evolution› = ‹"Dwek"›:		
‹HMW_carb_deplet›	$\delta^{\mathrm{HMW}}_{\mathrm{carb}}$	Eq. (21), p. 12;
‹LMW_carb_deplet›	$\delta^{\mathrm{LMW}}_{\mathrm{carb}}$	Idem;
‹CCSN_carb_deplet›	$\delta^{\mathrm{CC\,SN}}_{\mathrm{carb}}$	Eq. (23), p. 12;
‹SNIa_carb_deplet›	$\delta^{\mathrm{SN\,Ia}}_{\mathrm{carb}}$	Idem;
‹HMW_sil_deplet›	$\delta^{\mathrm{HMW}}_{\mathrm{sil}}$	Eq. (22), p. 12;
‹LMW_sil_deplet›	$\delta^{\mathrm{LMW}}_{\mathrm{sil}}$	Idem;
‹CCSN_sil_deplet›	$\delta^{\mathrm{CC\,SN}}_{\mathrm{sil}}$	Eq. (24), p. 13;
‹SNIa_sil_deplet›	$\delta^{\mathrm{SN\,Ia}}_{\mathrm{sil}}$	Idem;
‹SN_swept_mass›	m_{swept}	Eq. (25), p. 13;
‹carb_accr_timescale›	$\tau^{\mathrm{accr}}_{\mathrm{carb}}$	Eq. (26), p. 13;
‹sil_accr_timescale›	$\tau^{\mathrm{accr}}_{\mathrm{sil}}$	Eq. (27), p. 13.
For both dust evolution models:		
‹O_sil_ratio›	Ξ	Eq. (20), p. 12.

h. Dust attenuation and emission parameters

⟶ App. E.8, p. 61.

α. ‹extinction›, ‹extinction_SFC›, ‹extinction_DISM›

⟶ Sec. II.6, p. 14.

The extinction due to dust grains in clouds (resp. the diffuse medium) is computed if and only if the value of parameter ‹extinction_SFC› (resp. ‹extinction_DISM›) is ‹.true.›.

The values of ‹extinction_SFC› and ‹extinction_DISM› may be given at once through the explicit assignment of parameter ‹extinction› ※43.

β. ‹grains_file›, ‹grains_file_SFC›, ‹grains_file_DISM›

⟶ Sec. II.5.*a*, p. 11

‹grains_file_SFC› (resp. ‹grains_file_DISM›) ※43 holds the name of the ***file of grains*** providing the size distribution in clouds (resp. the diffuse medium) of the various species of dust grains ※34. This file also points to files (one for each species) giving the optical properties of individual grains. Currently available files of grains are ‹"ZDA.txt"›, ‹"LWD.txt"› and ‹"MRN.txt"›.

If the value of ‹extinction_SFC› (resp. ‹extinction_DISM›) is ‹.false.›, assigning ‹grains_file_SFC› (resp. ‹grains_file_DISM›) has no effect.

The values of ‹grains_file_SFC› and ‹grains_file_DISM› may be given at once through the explicit assignment of parameter ‹grains_file› ※34 ※43.

γ. ‹geometry›

If ‹geometry› = ‹"spiral"›, the spatial distribution of stars and dust described in Sec. II.6.*a*, p. 14, is used.

If ‹geometry› = ‹"spheroidal"›, the spatial distribution of stars and dust described in Sec. II.6.*b*, p. 14, is used.

If ‹geometry› = ‹"slab"›, the spatial distribution of stars and dust described in Sec. II.6.*c*, p. 14, is used.

Table 4. Meaning of code parameters related to the spatial distribution of stars and dust.

Code parameter	Notation used in the modeling	Sections or equations where the quantity appears
	If ‹geometry› = ‹"spiral"›:	
‹bulge_tot_ratio›	Γ	Eq. (17) of P3AA;
‹M_sys_spiral›	$M_{\mathrm{sys}}^{\mathrm{spir}}$	Sec. 4.1 of P3AA;
‹expo_radius›	R_{d}	Eq. (13) of P3AA.
	If ‹geometry› = ‹"spheroidal"›:	
‹M_sys_spher›	$M_{\mathrm{sys}}^{\mathrm{sph}}$	Sec. 4.2 of P3AA;
‹core_radius›	R_{c}	Eq. (18) of P3AA.
	If ‹geometry› = ‹"slab"›:	
‹slab_factor›	α_{slab}	Eq. (34), p. 14.

δ. ‹inclin_averaged›, ‹inclination›

These parameters are not relevant if ‹geometry› = ‹"spheroidal"›.

If ‹inclin_averaged› = ‹.true.›, the attenuation is averaged over all inclinations. Otherwise, the attenuation is computed for the inclination specified by parameter ‹inclination›. The inclination is in degrees and relative to face-on (0° for face-on, 90° for edge-on).

[43] For instance, setting ‹extinction› to ‹.true.› or ‹.false.› immediately sets ‹extinction_SFC› and ‹extinction_DISM› to this same value.

Note that the parameters ‹extinction›, ‹grains_file›, ‹dust_emission›, ‹stoch_heating›, ‹self_abs_power›, ‹carb_sublim_¬ temp›, ‹sil_sublim_temp›, ‹nebular_emission›, ‹neb_emis_type› and ‹neb_emis_const_frac› are not used in the calculations. Only the ‹*_SFC› and ‹*_DISM› variants of these parameters matter and are written in the output file or are printed by command ‹echo›.

ε. **‹dust_emission›, ‹dust_emission_SFC›, ‹dust_emission_DISM›**

⟶ Sec. II.7, p. 15.

The emission produced by dust grains in clouds (resp. the diffuse medium) is computed if and only if the value of parameter ‹dust_emission_SFC› (resp. ‹dust_emission_DISM›) is ‹.true.› *and* the value of ‹extinction_SFC› (resp. ‹extinction_DISM›) is ‹.true.› too.

The values of ‹dust_emission_SFC› and ‹dust_emission_DISM› may be given at once through the explicit assignment of parameter ‹dust_emission› ※43.

ζ. **‹stoch_heating›, ‹stoch_heating_SFC›, ‹stoch_heating_DISM›**

⟶ Sec. II.7.*b*, p. 15.

Stochastic heating of dust grains in clouds (resp. the diffuse medium) is considered if and only if the value of parameter ‹stoch_heating_SFC› (resp. ‹stoch_heating_DISM›) is ‹.true.›.

If the value of ‹extinction_SFC› (resp. ‹extinction_DISM›) or ‹dust_emission_SFC› (resp. ‹dust_emission_DISM›) is ‹.false.›, assigning ‹stoch_heating_SFC› (resp. ‹stoch_heating_DISM›) has no effect.

The values of ‹stoch_heating_SFC› and ‹stoch_heating_DISM› may be given at once through the explicit assignment of parameter ‹stoch_heating› ※43.

η. **‹carb_sublim_temp›, ‹carb_sublim_temp_SFC›, ‹carb_sublim_temp_DISM›, ‹sil_sublim_temp›, ‹sil_sublim_temp_SFC›, ‹sil_sublim_temp_DISM›**

⟶ Sec. II.7.*b*, p. 15.

The values of ‹carb_sublim_temp_SFC› (resp. ‹carb_sublim_temp_DISM›) and ‹sil_sublim_temp_SFC› (resp. ‹sil_sublim_temp_DISM›) are the sublimation temperatures of carbonaceous grains and silicates in clouds (resp. the diffuse medium). Note that grains do not actually sublimate in the code, but the SEDs of grains at temperatures above the sublimation temperature are printed separately, if requested (see parameter ‹sublim_output› in Sec. IV.3.*j*.α.i, p. 30).

If the value of ‹extinction_SFC› (resp. ‹extinction_DISM›) or ‹dust_emission_SFC› (resp. ‹dust_emission_DISM›) is ‹.false.›, assigning the ‹*_sublim_temp_SFC› (resp. ‹*_sublim_temp_DISM›) parameters has no effect.

The values of ‹carb_sublim_temp_SFC› and ‹carb_sublim_temp_DISM› may be given at once through the explicit assignment of parameter ‹carb_sublim_temp› ※43. The same holds, *mutatis mutandis*, for ‹sil_sublim_temp_SFC›, ‹sil_sublim_temp_DISM› and ‹sil_sublim_temp›.

θ. **‹self_abs_power›, ‹self_abs_power_SFC›, ‹self_abs_power_DISM›**

⟶ Sec. II.7.*c*, p. 15.

The value of ‹self_abs_power_SFC› (resp. ‹self_abs_DISM›) is the value in clouds (resp. the diffuse medium) of the parameter γ used in Eq. (38), p. 16. The default value corresponds to no self-absorption.

If the value of ‹extinction_SFC› (resp. ‹extinction_DISM›) or ‹dust_emission_SFC› (resp. ‹dust_emission_DISM›) is ‹.false.›, assigning ‹self_abs_power_SFC› (resp. ‹self_abs_power_DISM›) has no effect.

The values of ‹self_abs_power_SFC› and ‹self_abs_power_DISM› may be given at once through the explicit assignment of parameter ‹self_abs_power› ※43.

i. Parameters for star-forming clouds and nebular emission

⟶ P3AA, sec. 6.

⟶ App. E.9, p. 63.

α. **‹cloud_init_frac›, ‹cloud_duration›, ‹cloud_power›, ‹cluster_stel_mass›**

‹cloud_init_frac›, ‹cloud_duration› and ‹cloud_power› are the values of the parameters φ_0, θ and β in Eq. (40), p. 16. The parameter ‹cluster_stel_mass› is the typical initial stellar mass of a star cluster (quantity $M_{\rm sc}$ in eq. (29) of P3AA).

β. ‹nebular_emission›, ‹nebular_emission_SFC›, ‹nebular_emission_DISM›

The emission of the nebular gas in clouds (resp. the diffuse medium) is computed if and only if the value of parameter ‹nebular_emission_SFC› (resp. ‹nebular_emission_DISM›) is ‹.true.›.

The values of ‹nebular_emission_SFC› and ‹nebular_emission_DISM› may be given at once through the explicit assignment of parameter ‹nebular_emission› ※43.

i. ‹neb_emis_type› or ‹neb_emis_type_SFC› or ‹neb_emis_type_DISM› = ‹"automatic"›

If ‹neb_emis_type_SFC› (resp. ‹neb_emis_type_DISM›) is set to ‹"automatic"›, the fraction of Lyman continuum photons emitted by young stars in star-forming clouds (resp. the diffuse medium) which are absorbed (in the same region) by gas rather than by dust is computed as explained in Sec. II.8, p. 16, and app. A of P3AA.

The values of ‹neb_emis_type_SFC› and ‹neb_emis_type_DISM› may be given at once through the explicit assignment of parameter ‹neb_emis_type› ※43.

ii. ‹neb_emis_type› or ‹neb_emis_type_SFC› or ‹neb_emis_type_DISM› = ‹"constant"›: ‹neb_emis_const_frac›, ‹neb_emis_const_frac_SFC›, ‹neb_emis_const_frac_DISM›

If ‹neb_emis_type_SFC› (resp. ‹neb_emis_type_DISM›) is set to ‹"constant"›, the fraction of Lyman continuum photons emitted by young stars in star-forming clouds (resp. the diffuse medium) which are absorbed by the surrounding gas is assumed to be constant and is given by ‹neb_emis_const_frac_SFC› (resp. ‹neb_emis_const_frac_DISM›).

The values of ‹neb_emis_const_frac_SFC› and ‹neb_emis_const_frac_DISM› may be given at once through the explicit assignment of parameter ‹neb_emis_const_frac› ※43.

iii. ‹l10_mean_U_DISM›

⟶ P3AA, sec. 6.2.

Decimal logarithm of the mean value of the unitless ionization parameter in the diffuse interstellar medium.

j. Output files

⟶ App. E.10, p. 64.

α. Main output file of «spectra»

i. ‹spectra_output›, ‹RF_output›, ‹sublim_output›

Parameter ‹spectra_output› controls writing in the main output file of «spectra».

If ‹spectra_output› = ‹"basic"› (default), only the global SED is printed in the file of spectra (however, emission lines are distinguished from the continuum).

If ‹spectra_output› = ‹"detailed"›, the SEDs produced by the various galactic components (stars, ionized gas, grain species; see Table 9, p. 38) in the various regions (star-forming clouds and diffuse ISM) are printed separately, in addition to the global SED.

Whether the value of ‹spectra_output› is ‹"basic"› or ‹"detailed"›, a series of other quantities —masses of components, chemical composition, star formation and supernova rates… — are also written (see Table 8, p. 36). If ‹spectra_output› = ‹"none"›, no file of spectra is produced.

If ‹RF_output› is true, the mean radiation fields in star-forming clouds and the diffuse medium are written in the file of spectra; if ‹sublim_output› is true, the SEDs of grains at temperatures above the sublimation temperature are written separately. (See Table 9, p. 38.)

The file of spectra may be read with the procedure ‹read_spectra_output› defined in «mod_read_spectra_output.f90», as exemplified by codes «colors» and «plot_spectra.f90».

ii. ‹spectra_file›, ‹prefix›, ‹stamp_time›, ‹overwrite›

The value of ‹spectra_file› is the name of the main output file of «spectra» for the current scenario. Contrary to other parameters, this value is erased at the beginning of each scenario. If no value is provided, a default name is built from the date and time at the beginning of the execution of «spectra» (a “time-stamp”) and from the ordinal number of the scenario in the file of scenarios.

If the file of spectra already exists, a modified name is derived from the original name and from the time-stamp, unless ‹overwrite› is ‹.true.› (in which case the old file of spectra is replaced by the new one).

If ‹prefix› ≠ ‹""›, the name of the main output file is built by prepending the value of ‹prefix› to either the value provided for ‹spectra_file›, the default name or the modified name.

In all cases, the time-stamp is inserted between ‹prefix› and ‹spectra_file› if ‹stamp_time› = ‹.true.›. Underscores are used to separate the prefix and the time-stamp from what follows.

iii. ‹ages_file›

The value of ‹ages_file› is the name of the file containing the ages (variable ‹output_age› in code «spectra») ※[44] at which quantities are printed in the file of spectra (one age in Myr per line). This file should be in «ages_dir/».

β. Output files related to grain properties

i. ‹grain_temp_output›, ‹grain_temp_file›, ‹grain_SED_output›, ‹grain_SED_file›

By default, ‹grain_temp_output› = ‹.false.›. If set to ‹.true.›, the temperature probability distributions of individual dust grains are written in a file of grain temperatures.

The name of this file is given by parameter ‹grain_temp_file› and is processed in the same way as ‹spectra_file›. This name is erased at the beginning of each scenario; if not provided, a name is created from the processed value of ‹spectra_file›.

The output file may be read with the procedure ‹read_grain_temp› defined in «mod_read_grain_temp.f90», as exemplified by code «plot_grain_temp.f90».

■

If ‹grain_SED_output› = ‹.true.› (default: ‹.false.›), the spectral energy distributions of individual dust grains are written in a file of grain SEDs. The name of this file is given by parameter ‹grain_SED_file› (same remarks as for ‹grain_temp_file›). The output file may be read with the procedure ‹read_grain_SED› defined in «mod_read_grain_SED.f90», as exemplified by code «plot_grain_SED.f90».

■

The parameters ‹prefix›, ‹stamp_time› and ‹overwrite› also apply to the files of grain temperatures and grain SEDs.The output files produced if ‹grain_temp_output› or ‹grain_SED_output› are ‹.true.› may be huge. The values of the parameters ‹output_grain_*› defined hereafter may be changed to reduce their size.

ii. ‹grain_output_SFC›, ‹grain_output_DISM›

The temperature probability and spectral energy distributions of dust grains in star-forming clouds are written only if ‹grain_output_SFC› = ‹.true.› (default).

Idem for dust grains in the diffuse medium if ‹grain_output_DISM› = ‹.true.›.

iii. ‹grain_output_min_age›, ‹grain_output_max_age›

Grain temperature and spectral energy distributions are written for all ages in the interval [‹grain_output_min_age›, ‹grain_output_max_age›]. (By default, at all the ages defined by ‹ages_file›.)

iv. ‹grain_output_min_size›, ‹grain_output_max_size›

Grain temperature and spectral energy distributions are written for all grain radii in the interval [‹grain_output_min_size›, ‹grain_output_max_size›]. (By default, for all available radii.)

4. Random numbers: ‹seed›, ‹initialize_seed›

Random numbers are used to simulate stochastic star formation (see Sec. IV.3.*e*.*κ*, p. 24); they are generated from integer seeds s_1, …, s_n (beware, the number n of seeds depends on the compiler!). To set the seeds to specific values, assign the parameters ‹seed(i)› for $i \in [\![1, n]\!]$.

[44] Actually, «spectra» outputs the spectrum and other evolving quantities at the nearest age (‹convol_time›) for which the state of the galaxy was computed. These values are identical if ‹output_age› is a multiple of the convolution time-step (variable ‹time_step›) used in «spectra» to convolve the star formation history with the properties of SSPs.

The seeds generating the sequence of random numbers used for a given scenario are written in the main output file of «spectra»; search for the line starting with

“seed(1:*n*) = [”.

To generate the same sequence of random numbers for some other scenario, enter this line (and, if necessary, following lines up to the closing “]” included) in the section of the file of scenarios describing this other scenario. (You may need to break the line; see App. B.1, p. 47, item “Line length and continuation character”.)

To set the seeds to some random values determined from the computer's clock, use instead the command ‹initialize_seed›.

5. End of file and other statements

a. End of file

When the end of the file of scenarios (the end of the most-outer file in the case of included files; see Sec. IV.5.*f*, p. 32) is reached, control is returned to «spectra» and the last scenario is run (normally).

b. ‹end›

The ‹end› command has the same effect as the end of the file of scenarios. Nothing will be read thereafter. Practically, you may write “end” after the last scenario you want to run if this is not the last one in the file of scenarios.

c. ‹return›

The ‹return› command returns control to «spectra»; the most recently read scenario is then run. Once this one has been computed, «spectra» reads the next scenario in the file of scenarios. The string “return” must be written after each scenario (except, possibly, the last one since an end of file or an ‹end› statement will have the same effect).

d. ‹stop›

The ‹stop› command interrupts the execution of «spectra».

e. ‹echo›

If the ‹echo› command is encountered while reading the file of scenarios or a file included in it with ‹include›, the name of this file, the line number where the statement ‹echo› appears and the values of all the relevant parameters are printed on the screen and in the log file (see Sec. I.3.*b*, p. 6). The text produced by all the ‹add_text› commands is also printed (see Sec. IV.5.*i*, p. 33).

f. ‹include›

The contents of an external file (e.g. «*included_file*») may be inserted in the file of scenarios by typing “include "⌜*path/*⌟*included_file*"” (or “include '⌜*path/*⌟*included_file*'”), where the optional path ‹*path/*› is required only if the included file is not in «scenarios_dir/». An included file may itself contain ‹include› statements.

g. ‹verbosity›

If the value of the parameter ‹verbosity› is ⩾ 0, all warnings are written on the screen. If ‹verbosity› ⩾ 1, major steps in the execution of «spectra» are also shown.

h. ‹check_only›

By default, the value of the parameter ‹check_only› is ‹.false.›. If set to ‹.true.›, the scenarios are not run when the statements ‹return› and ‹end› or the end of the most outer file are encountered (see Sec. IV.5, p. 32), and this until “check_only = .false.” is met.

Note that setting ‹check_only› to ‹.true.› does not interrupt the reading of input parameters, so this may be used to check the syntax of the file of scenarios.

i. ‹add_text›, ‹erase_text›

Lines of text containing additional informations (the purpose of the scenario, for instance) may be written at the beginning of the output files of «spectra» with these commands. The line "add_text "*string*"" (or "add_text '*string*'"), where ‹*string*› denotes a string of characters, *appends* the line ‹*string*› to this text. To erase *all* these lines, use the command ‹erase_text›.

j. ‹reset_*› commands

α. ‹reset_cosmo›

This command resets all the cosmological parameters (listed in Sec. IV.3.*a*, p. 21) to their default values if they exist, or to "undefined" otherwise.

β. ‹reset_reserv_infall›

This command resets all the parameters related to reservoirs and infall episodes (listed in Sec. IV.3.*d*, p. 22) to their default values if they exist, or to "undefined" otherwise.

γ. ‹reset_SF›

This command resets all the parameters related to star formation episodes (listed in Sec. IV.3.*e*, p. 23) to their default values if they exist, or to "undefined" otherwise.

δ. ‹reset_outflow›

This command resets all the parameters related to outflow episodes (listed in Sec. IV.3.*f*, p. 25) to their default values if they exist, or to "undefined" otherwise.

ε. ‹reset_dust_evol›

This command resets all the parameters related to dust evolution (listed in Sec. IV.3.*g*, p. 27) to their default values if they exist, or to "undefined" otherwise.

ζ. ‹reset_dust_transfer›

This command resets all the parameters related to dust extinction and emission (listed in Sec. IV.3.*h*, p. 28) to their default values if they exist, or to "undefined" otherwise.

η. ‹reset_cloud_neb›

This command resets all the parameters related to clouds and nebular emission (listed in Sec. IV.3.*i*, p. 29) to their default values if they exist, or to "undefined" otherwise.

θ. ‹reset_output›

This command resets all the parameters related to output files (listed in Sec. IV.3.*j*, p. 30) to their default values if they exist, or to "undefined" otherwise. It also executes the command ‹erase_text› (see Sec. IV.5.*i*, p. 33).

ι. ‹reset_others›

This command resets all the parameters listed in Sec. IV.3.*b*, p. 21, in Sec. IV.3.*c*, p. 21, and in Sec. IV.5.*g*, p. 32, to their default values if they exist, or to "undefined" otherwise.

κ. ‹reset_all›

The ‹reset_all› statement executes all the ‹reset_*› statements listed above.

V. Outputs

1. Outputs of «SSPs»

The output files of «SSPs» are not written to be read by the user but to be processed by code «spectra». See Footnote 31, p. 16.

2. Outputs of «spectra»

a. Main output file

α. Reading procedure

The data written by «spectra» in a file of spectra may be read using the Fortran subroutine ‹read_spectra_output›※[45]. To read a file «*file_name*» from within a Fortran 95 program, do the following:

1. Type the line

   “use mod_read_spectra_output”

   right after the ‹program› statement to load the module ‹mod_read_spectra_output› defined in «mod_read_spectra_output.f90». This module provides the subroutine ‹read_spectra_output› and a data structure, ‹struct_spectra_output›. Tables 5 and 7 to 10 list all the fields in this structure;
2. Declare a variable, say ‹*data*›, which will contain all the data in the file of spectra. To do this, insert the line

   “type(struct_spectra_output) :: *data*”

   among the declarations of variables;
3. Read the data in the executable part of the program with the line

   “call read_spectra_output(⌈file_name = ⌋"*file_name*", ⌈file = ⌋*data*)”.

   This statement also allocates all the array fields of ‹*data*› to the right size;
4. The value of any component ‹*field*› (see Tables 5 and 7 to 10 for a complete list) of ‹*data*› is given by

   ‹*data* % *field*›.

To compile this program, link it (with the appropriate paths) to «util_dir/mod_types.f90», «util_dir/mod_dir_access.f90», «util_dir/mod_file_access.f90», «util_dir/mod_strings.f90», «source_dir/mod_directories.f90» and «source_dir/mod_read_spectra_output.f90».

Here is a basic Fortran example showing how to read, with the program «source_dir/example_read_spectra.f90» provided with the code, the files «*example_spectra1.txt*» and «*example_spectra2.txt*» computed in Sec. I.3.*b*, p. 6, and how to write to the screen the corresponding continuous spectrum at all ages and wavelengths:

```
!# With "example_read_spectra.f90" in "source_dir/", type for instance
!# ''gfortran -o example_read_spectra ../util_dir/mod_types.f90 \
        ../util_dir/mod_dir_access.f90 ../util_dir/mod_file_access.f90 \
        ../util_dir/mod_strings.f90 ../source_dir/mod_directories.f90 \
        ../source_dir/mod_read_spectra_output.f90 \
        ../source_dir/example_read_spectra.f90''
!# (without quotation marks!) to compile it from within "bin_dir/"
!# with `gfortran`.

program example_read_spectra

 use mod_read_spectra_output                                          → Step 1
 implicit none                                                          above.
 type(struct_spectra_output), dimension(2) :: data                    → Step 2.
 integer :: i, j, k

 call read_spectra_output("example_spectra1.txt", data(1))            → Step 3.
 call read_spectra_output("example_spectra2.txt", data(2))              Idem.
 do j = 1, data(1) % dim_output_age                                   → Step 4.
```

[45] This subroutine is also called in «plot_spectra.f90» and «colors.f90».

```
   write(*,*) "Wavelength, monochromatic luminosities in &
        &''example_spectra1.txt'' and ''example_spectra2.txt'' &
        &at age = ", data(1) % output_age(j), ":"                          Idem.
   do k = 1, data(1) % dim_cont                                            Idem.
      write(*,*) data(1) % lambda_cont(k), (data(i) % lum_cont(j, k), i = 1, 2)   Idem.
   enddo
 enddo

end program example_read_spectra
```

β. Age-independent quantities

Table 5. Fields defined in structure ‹struct_spectra_output›: array sizes.

Field	Meaning
‹*data* % dim_output_age›	Number of galactic ages for which quantities in Table 8, p. 36, are printed
‹*data* % dim_elem›	Number of elements followed during chemical evolution
‹*data* % dim_species_SFC›	Number of dust species (i.e. graphites, silicates, various kinds of PAHs...) in star-forming clouds
‹*data* % dim_species_DISM›	Number of dust species in the diffuse ISM
‹*data* % dim_cont›	Number of continuum wavelengths
‹*data* % dim_line›	Number of emission lines

Table 6. Meaning of indices used in Tables 7 to 10.

Index	Quantity referred to	Index range
i	Element ‹*data* % elem_id(i)›	⟦1, ‹*data* % dim_elem›⟧
j	Age ‹*data* % output_age(j)›	⟦1, ‹*data* % dim_output_age›⟧
k	Continuum wavelength ‹*data* % lambda_cont(k)›	⟦1, ‹*data* % dim_cont›⟧
ℓ	Emission line ‹*data* % line_id(ℓ)›	⟦1, ‹*data* % dim_line›⟧
m	Grain species ‹*data* % species_id_SFC(m)› or ‹*data* % species_id_DISM(m)›	⟦1, ‹*data* % dim_species_SFC›⟧ or ⟦1, ‹*data* % dim_species_DISM›⟧

Table 7. Fields defined in structure ‹struct_spectra_output›: constant quantities.

Field	Meaning	Unit
‹*data* % version_id›	Identifier of the code version	∅
‹*data* % version_date›	Date of the code version	∅
‹*data* % spectra_output›	Value of the scenario parameter ‹spectra_output› (see Sec. IV.3.*j*.*α*.i, p. 30)	∅
‹*data* % RF_output›	Value of the scenario parameter ‹RF_output› (see Sec. IV.3.*j*.*α*.i, p. 30)	∅
‹*data* % sublim_output›	Value of the scenario parameter ‹sublim_output› (see Sec. IV.3.*j*.*α*.i, p. 30)	∅
‹*data* % time_step›	Time-step used in the evolution procedure of «spectra»	Myr

↓

↓

Field	Meaning	Unit
‹data % elem_id(i)›	Identifier of the i-th chemical element[46]	∅
‹data % species_id_SFC(m)›	Identifier of the m-th dust species[47] in star-forming clouds	∅
‹data % species_id_DISM(m)›	Identifier of the m-th dust species[47] in the diffuse ISM	∅
‹data % lambda_cont(k)›	Value of the k-th continuum wavelength	Å
‹data % line_id(ℓ)›	Identifier[48] of the ℓ-th emission line	∅
‹data % lambda_line(ℓ)›	Wavelength of the ℓ-th emission line	Å

The grid of wavelengths for the continuum spectra of galaxies merges the wavelengths of the library of stellar spectra (see Sec. III.4, p. 18) and those of optical properties of grains (see Sec. II.5.*a*, p. 11).

γ. Age-dependent quantities

Table 8. Fields defined in structure ‹struct_spectra_output›: main variable quantities.

Field	Meaning	Unit
‹data % output_age(j)›	Galactic age for which output of the spectrum is requested	Myr
‹data % convol_time(j)›	Age for which the spectrum is given (differs from ‹data % output_age(j)› by at most ‹data % time_step›)	Myr
‹data % cosmic_time(j)›	Cosmic time	Myr
‹data % redshift(j)›	Redshift	∅
‹data % galaxy_mass(j)›	Normalized mass of the galaxy	∅
‹data % live_stars_mass(j)›	Normalized mass of stars still alive	∅
‹data % WD_mass(j)›	Normalized mass of white dwarfs	∅
‹data % BHNS_mass(j)›	Normalized mass of black holes and neutron stars	∅
‹data % inert_mass(j)›	Normalized mass of inert objects (brown dwarfs, etc.)	∅
‹data % ISM_mass(j)›	Normalized mass of the ISM	∅
‹data % ISM_Z(j)›	Metallicity of the ISM	∅
‹data % stel_Z_mass_avrg(j)›	Mean birth-metallicity of stars, averaged over the masses of stars still alive	∅
‹data % stel_Z_bol_avrg(j)›	Mean birth-metallicity of stars, averaged over the bolometric luminosities of stars still alive	∅
‹data % carb_abund(j)›	Mass fraction of carbonaceous dust relative to the ISM mass	∅
‹data % sil_abund(j)›	Mass fraction of silicate dust relative to the ISM mass	∅
‹data % ISM_abund(j, i)›	ISM abundance of the i-th element, in mass fraction, relative to all the elements in the ISM[49]	∅

↓

[46] In the current implementation: 1 → O; 2 → C; 3 → Fe; 4 → He; 5 → N; 6 → Ne; 7 → Mg; 8 → Si; 9 → S; 10 → Ca.

[47] In the current implementation: 1 → graphites; 2 → neutral PAHs; 3 → ionized PAHs; 4 → silicates.

[48] Warning: each value of ‹line_id› must refer to a single emission line. All emission lines are listed with their identifier and their wavelength in «Cloudy_dir/list_neb_lines.txt».

[49] In «spectra.f90», ‹data % ISM_abund(j, 0)› also exists and is the same as ‹data % ISM_Z(j)›.

↓

Field	Meaning	Unit
‹*data* % L_bol(*j*)›	Normalized bolometric luminosity (radiant power) of the galaxy	$\mathcal{L}_\odot/M_\odot$
‹*data* % tau_V(*j*)›	Optical depth in the V-band of dust in the diffuse ISM. Computed from side to side through the center of the galaxy (along the rotation axis for axisymmetric galaxies)	∅
‹*data* % dust_bol_ratio(*j*)›	Ratio of the dust luminosity to the bolometric one	∅
‹*data* % SF_rate(*j*)›	Normalized star formation (mass) rate	Myr^{-1}
‹*data* % Lyman_cont_rate(*j*)›	Normalized number rate of Lyman continuum photons emitted by stars	s^{-1} $M_\odot^{-1}$
‹*data* % CCSN_rate(*j*)›	Normalized number rate of core collapse supernovae	Myr^{-1} $M_\odot^{-1}$
‹*data* % SNIa_rate(*j*)›	Idem for type Ia supernovae	Myr^{-1} $M_\odot^{-1}$
‹*data* % stel_age_mass_avrg(*j*)›	Mean stellar age, averaged over the masses of stars still alive	Myr
‹*data* % stel_age_bol_avrg(*j*)›	Mean stellar age, averaged over the bolometric luminosities of stars still alive	Myr
‹*data* % Lyman_cont_gas_abs(*j*)›	Fraction (in number) of Lyman continuum photons emitted by stars which are absorbed by gas	∅
‹*data* % Lyman_cont_dust_abs(*j*)›	Fraction (in number) of Lyman continuum photons emitted by stars which are absorbed by dust	∅
‹*data* % ejec_rate_tot(*j*)›	Normalized mass ejection rate by stars of matter into the ISM of the galaxy	Myr^{-1}
‹*data* % infall_rate(*j*)›	Normalized mass infall rate from all reservoirs onto the galaxy	Myr^{-1}
‹*data* % outflow_rate(*j*)›	Normalized mass outflow rate from the ISM of the galaxy into the IGM	Myr^{-1}
‹*data* % ejec_cumul_mass(*j*)›	Cumulative normalized mass of stellar ejecta since the beginning	∅
‹*data* % SF_live_cumul_mass(*j*)›	Cumulative normalized mass of live stars formed since the beginning ※[50]	∅
‹*data* % infall_cumul_mass(*j*)›	Cumulative normalized mass of matter fallen from all reservoirs onto the galaxy since the beginning	∅
‹*data* % outflow_cumul_mass(*j*)›	Cumulative normalized mass of matter expelled by the galaxy into the IGM since the beginning	∅
‹*data* % L_dust_SFC(*j*)›	Normalized bolometric luminosity of dust grains in star-forming clouds	erg s^{-1} $M_\odot^{-1}$
‹*data* % L_dust_DISM(*j*)›	Idem for dust grains in the diffuse ISM	erg s^{-1} $M_\odot^{-1}$
‹*data* % lum_cont(*j*, *k*)›	Normalized attenuated (see Sec. IV.3.h.δ, p. 28) monochromatic continuum luminosity (spectral power) of the galaxy	erg s^{-1} Å^{-1} $M_\odot^{-1}$
‹*data* % L_line(*j*, *ℓ*)›	Normalized attenuated in-line luminosity of the galaxy	erg s^{-1} $M_\odot^{-1}$

[50] Contrary to ‹*data* % live_stars_mass›, the quantity ‹*data* % SF_live_cumul_mass› accounts for all the mass in stars considered as alive at birth: beside the mass in stars still alive at age ‹*data* % output_age(*j*)›, it includes the mass of stellar ejecta and compact stellar remnants, but not that in objects inert from the outset.

Table 9. Fields defined in structure ‹`struct_spectra_output`› if and only if field ‹`data % spectra_output`› = ‹`"detailed"`›.

Field	Meaning	Unit
‹`data % lum_stel_SFC_unatt(j, k)`›	Normalized unattenuated monochromatic continuum luminosity of stars in star-forming clouds	erg s^{-1} Å^{-1} $M_\odot^{-1}$
‹`data % lum_stel_DISM_unatt(j, k)`›	Idem for stars in the diffuse medium	erg s^{-1} Å^{-1} $M_\odot^{-1}$
‹`data % lum_neb_cont_SFC_unatt(j, k)`›	Normalized unattenuated monochromatic luminosity of the nebular continuum produced by ionized gas in star-forming clouds	erg s^{-1} Å^{-1} $M_\odot^{-1}$
‹`data % lum_neb_cont_DISM_unatt(j, k)`›	Idem for ionized gas in the diffuse medium	erg s^{-1} Å^{-1} $M_\odot^{-1}$
‹`data % lum_species_SFC(m, j, k)`›	Normalized monochromatic continuum luminosity emitted in star-forming clouds by the m-th dust species	erg s^{-1} Å^{-1} $M_\odot^{-1}$
‹`data % lum_species_DISM(m, j, k)`›	Idem for the m-th dust species in the diffuse medium	erg s^{-1} Å^{-1} $M_\odot^{-1}$
‹`data % L_line_SFC_unatt(j, ℓ)`›	Normalized unattenuated luminosity of emission lines produced by ionized gas in star-forming clouds	erg s^{-1} $M_\odot^{-1}$
‹`data % L_line_DISM_unatt(j, ℓ)`›	Idem for ionized gas in the diffuse medium	erg s^{-1} $M_\odot^{-1}$
Quantities printed only if ‹`data % RF_output`› = ‹`.true.`›:		
‹`data % RF_cont_SFC(j, k)`›	Mean monochromatic radiation field[51] in star-forming clouds	erg cm^{-3} μm^{-1}
‹`data % RF_cont_DISM(j, k)`›	Idem in the diffuse medium	erg cm^{-3} μm^{-1}
‹`data % RF_line_SFC(j, ℓ)`›	Mean in-line radiation field[52] in star-forming clouds	erg cm^{-3}
‹`data % RF_line_DISM(j, ℓ)`›	Idem in the diffuse medium	erg cm^{-3}
Quantities printed only if ‹`data % sublim_output`› = ‹`.true.`›:		
‹`data % sublim_lum_species_SFC(m, j, k)`›	Normalized monochromatic continuum luminosity emitted in star-forming clouds by dust grains of the m-th dust species at temperatures above the sublimation temperature	erg s^{-1} Å^{-1} $M_\odot^{-1}$
‹`data % sublim_lum_species_DISM(m, j, k)`›	Idem for dust grains in the diffuse medium	erg s^{-1} Å^{-1} $M_\odot^{-1}$

[51] This is the electromagnetic energy density per unit wavelength in the continuum, averaged in the medium, i.e. the quantity $\langle u_\lambda \rangle$ given by Eq. (35), p. 15.

[52] This is the total electromagnetic energy density in the emission line, averaged in the medium, i.e. $\int_{\text{line }\#\ell} \langle u_\lambda \rangle \, \mathrm{d}\lambda$, where $\langle u_\lambda \rangle$ is given by Eq. (35), p. 15.

δ. Warnings

Table 10. Fields defined in structure ‹struct_spectra_output›: warnings.

Field	Meaning	Unit
‹*data* % reserv_warn_present›	Flag set to ‹.true.› if the sum of the initial masses of the reservoirs exceeds the total mass of the system	∅
‹*data* % SF_warn_present›	Flag set to ‹.true.› if the total star formation rate exceeds the maximal possible value at some age	∅
‹*data* % SF_warn_age›	Age at which the total star formation rate exceeds the maximal possible value	Myr
‹*data* % dim_infall_warn›	Number of reservoirs for which the infall rate exceeds the maximal possible value at some age	∅
The index r below is in ⟦1, ‹*data* % dim_infall_warn›⟧.		
‹*data* % infall_warn_present(r)›	Index of the reservoir for which the infall rate exceeds the maximal possible value at some age	∅
‹*data* % infall_warn_age(r)›	Age at which the infall rate exceeds the maximal possible value in this reservoir	Myr
‹*data* % outflow_warn_present›	Flag set to ‹.true.› if the total outflow rate exceeds the maximal possible value at some age	∅
‹*data* % outflow_warn_age›	Age at which the total outflow rate possibly exceeds the maximal possible value	Myr
‹*data* % opt_depth_warn_present(j)›	Flag set to ‹.true.› if the optical depth is larger than the maximal one used in radiative transfer precomputations at age ‹*data* % output_age(j)›	∅
‹*data* % opt_depth_warn_min_lambda(j)›	Lower wavelength for which this occurs	Å
‹*data* % opt_depth_warn_max_lambda(j)›	Upper wavelength for which this occurs	Å

b. Outputs for grains

α. Reading a file of grain temperatures

The data written in a file of grain temperatures may be read with the subroutine ‹read_grain_temp› of module «source_dir/mod_read_grain_temp.f90». To do this, create a Fortran 95 code loading this module and declare a structure of type ‹struct_grain_temp›; its fields are described in Table 11, p. 39. Detailed explanations on the creation and the compilation of this code are provided in the header of «mod_read_grain_temp.f90» (the procedure is similar to the one used for the main output file of «spectra»; see Sec. V.2.*a*.*α*, p. 34). See «source_dir/plot_grain_temp.f90» for an example of use.

Table 11. Fields defined for a structure ‹*data*› of type ‹struct_grain_temp›.

Field	Meaning	Unit
‹*data* % version_id›	Identifier of the code version	∅
‹*data* % version_date›	Date of the code version	∅
‹*data* % dim_age_grains›	Number of ages	∅
The index j below is in ⟦1, ‹*data* % dim_age_grains›⟧.		

↓

↓

Field	Meaning	Unit
‹*data* % dim_region›	Number of regions (star-forming clouds, diffuse ISM)	∅
The index n below is in ⟦1, ‹*data* % dim_region›⟧.		
‹*data* % age_grains(j)›	Age	Myr
These ages are a subset of the ‹output_age(1:dim_output_age)› of Table 8, p. 36.		
‹*data* % region(n)›	Data for the n-th region	∅
Subfields of ‹*data* % region(n)› (denoted by “*region*” below):		
‹*region* % id›	Identifier of the region: ‹"SFC"› for star-forming clouds, ‹"DISM"› for the diffuse ISM	∅
‹*region* % dim_species›	Number of grain species (graphite, silicate, etc.) for this region	∅
The index m below is in ⟦1, ‹*region* % dim_species›⟧.		
‹*region* % stoch_heating›	Boolean set to ‹.true.› if the probability distribution of grain temperatures is computed (so, grains are stochastically heated), and to ‹.false.› if the equilibrium temperature is used	∅
‹*region* % species(m)›	Data for the m-th grain species in the n-th region	∅
Subfields of ‹*data* % region(n) % species(m)› (denoted by “*species*” below):		
‹*species* % id›	Identifier of the grain species	∅
‹*species* % dim_radius›	Number of grain radii	∅
The index p below is in ⟦1, ‹*species* % dim_radius›⟧.		
‹*species* % radius(p)›	Radius of the grain with the p-th radius	μm
‹*species* % state(j, p)›	Data for this grain at the j-th time	∅
Subfields of ‹*data* % region(n) % species(m) % state(j, p)› (denoted by “*state*” below):		
‹*state* % temp_eq›	Equilibrium temperature of the grain	K
‹*state* % dim_temp›	Number of grain temperatures	∅
The index q below is in ⟦1, ‹*state* % dim_temp›⟧.		
‹*state* % temp(q)›	Temperature T of the grain	K
‹*state* % prob(q)›	Value of $\mathrm{d}P/\mathrm{d}(\log_{10} T)$ for this temperature, where $P(x)$ is the probability that $T < x$	∅

β. Reading a file of grain SEDs

The data written in a file of grain SEDs may be read with the subroutine ‹read_grain_SED› of module «source_dir/mod_read_grain_SED.f90». To do this, create a Fortran 95 code loading this module and declare a structure of type ‹struct_grain_SED›; its fields are described in Table 12, p. 40. Detailed explanations on the creation and the compilation of this code are provided in the header of «mod_read_grain_SED.f90» (the procedure is similar to the one used for the main output file of «spectra»; see Sec. V.2.*a*.*α*, p. 34). See «source_dir/plot_grain_SED.f90» for an example of use.

Table 12. Fields defined for a structure ‹*data*› of type ‹struct_grain_SED›.

Field	Meaning	Unit
‹*data* % version_id›	Identifier of the code version	∅
‹*data* % version_date›	Date of the code version	∅

↓

↓

Field	Meaning	Unit
‹*data* % dim_age_grains›	Number of ages	∅
The index j below is in ⟦1, ‹*data* % dim_age_grains›⟧.		
‹*data* % dim_region›	Number of regions (star-forming cloud, diffuse ISM)	∅
The index n below is in ⟦1, ‹*data* % dim_region›⟧.		
‹*data* % age_grains(j)›	Age	Myr
These ages are a subset of the ‹output_age(1:dim_output_age)› of Table 8, p. 36.		
‹*data* % region(n)›	Data for the n-th region	∅
Subfields of ‹*data* % region(n)› (denoted by “*region*” below):		
‹*region* % id›	Identifier of the region: ‹"SFC"› for star-forming clouds, ‹"DISM"› for the diffuse ISM	∅
‹*region* % dim_species›	Number of grain species (graphites, silicates, etc.) for this region	∅
The index m below is in ⟦1, ‹*region* % dim_species›⟧.		
‹*region* % stoch_heating›	Boolean set to ‹.true.› if SEDs are computed both for stochastically heated grains (so, with a temperature distribution) and at the equilibrium temperature, and to ‹.false.› if SEDs are computed only at the equilibrium temperature	∅
‹*region* % species(m)›	Data for the m-th grain species in the n-th region	∅
Subfields of ‹*data* % region(n) % species(m)› (denoted by “*species*” below):		
‹*species* % id›	Identifier of the grain species	∅
‹*species* % dim_lambda›	Number of wavelengths	∅
The index k below is in ⟦1, ‹*species* % dim_lambda›⟧.		
‹*species* % lambda(k)›	Wavelength	µm
These wavelengths are a subset of the ‹lambda_cont(1:dim_cont)› of Table 7, p. 35. Beware the unit!		
‹*species* % dim_radius›	Number of grain radii	∅
The index p below is in ⟦1, ‹*species* % dim_radius›⟧.		
‹*species* % radius(p)›	Radius of the p-th grain	µm
‹*species* % lum_stoch(j, p, k)›	Monochromatic luminosity for a stochastically heated grain (void if ‹*region* % stoch_heating› is ‹.false.›)	erg s^{-1} µm^{-1}
‹*species* % lum_eq(j, p, k)›	Monochromatic luminosity for a grain at equilibrium temperature	erg s^{-1} µm^{-1}

3. Outputs of «calib» and «colors»

Code «colors» takes as input a file of spectra produced by «spectra». Beside other quantities, it computes the in-band luminosities and magnitudes of the evolving galaxy SED through a series of filter passbands.

«colors» also needs to read the file «calib.txt», which is created by «calib» and contains various filter-related properties, in particular the in-band fluxes of some reference stars.

Before we turn to the outputs of «colors», we first have to look at how, depending on the filter, in-band quantities are defined.

a. Preliminary: filters, in-band quantities and magnitudes

α. Filters

For any filter f used by «calib» or «colors», a ***file of filter*** containing the following informations must exist:

- **First (non-comment) line**: ***filter's identifier*** (string of characters) used in «calib.txt» and «colors»;
- **Second line**: ***type of transmission*** (string): either ‹"energy"› or ‹"nb_phot"›.
  - **‹"energy"›**: the transmission is the value of the system's (filter + detector) *energy* response function, denoted by "$T_{f,\lambda}$" hereafter (quantity "$S'(\lambda)$" in Bessell & Murphy (2012), app. A2);
  - **‹"nb_phot"›**: the transmission is the value of the *photon* response function (quantity "$S(\lambda)$" in Bessell & Murphy (2012), with $S(\lambda) \propto T_{f,\lambda}/\lambda$);
- **Third line**: ***type of calibration*** (string): either ‹"Vega"›, ‹"AB"›, ‹"TG"›, ‹"HST"›, ‹"FOCA"›, ‹"IRAS"› or ‹"D4000"›.

  For type ‹"Vega"›, the zero-point offset (real number) *may* be given on the same line (set to 0 if absent). For type ‹"IRAS"›, the nominal wavelength in Å (real number) *must* be given on the same line;
- **Following lines**: ***passband of the filter***, with, on each line,
  - the wavelength λ in Å,
  - the transmission at this wavelength.

The names of the files of filters are listed in «calib_dir/list_filters.txt» (one per line). To add another filter, create a file of filter as explained in this section, and insert its name in «list_filters.txt». Code «calib» needs to be run before the filter may be used by «colors».

β. In-band fluxes and luminosities

The ***in-band flux*** $\overline{F}_f$ through filter f of the light emitted by some object (denoted by "$\star$" hereafter) is defined as follows, depending on the type of calibration:

- **‹"Vega"›, ‹"AB"›, ‹"TG"›, ‹"HST"› or ‹"FOCA"›**:

$$\overline{F}_f(\star) := \frac{1}{N_f}\int F_\lambda(\star)\,T_{f,\lambda}\,\mathrm{d}\lambda, \tag{70}$$

  where

$$N_f := \int T_{f,\lambda}\,\mathrm{d}\lambda \tag{71}$$

  is the ***normalisation*** of the filter, and $F_\lambda(\star)$ is the value, at wavelength λ and per unit wavelength, of the monochromatic flux (spectral irradiance) of the object;
- **‹"D4000"›**: the in-band flux and the normalisation are defined as

$$\overline{F}_f(\star) := \frac{1}{N_f}\int_{\lambda_1}^{\lambda_2} F_\nu(\star)\,\mathrm{d}\lambda \quad\text{and}\quad N_f := \int_{\lambda_1}^{\lambda_2}\mathrm{d}\lambda, \tag{72}$$

  where $F_\nu(\star)$ is the value, at frequency ν and per unit frequency, of the monochromatic flux of the object, and λ_1 and λ_2 are the bounds of the passband.

  Currently used only to compute the Balmer break indices $D4000$ (Bruzual 1983) and $D_\mathrm{n}4000$ (Balogh et al. 1999) from the passbands respectively contained in the files {«D4000-.txt», «D4000+.txt»} and {«Dn4000-.txt», «Dn4000+.txt»} of «calib_dir/»;
- **‹"IRAS"›**: $\overline{F}_f$ is defined by Eq. (70), but the normalisation is given by

$$N_f := \int T_{f,\lambda}\,\frac{\lambda_f^\mathrm{nom}}{\lambda}\,\mathrm{d}\lambda, \tag{73}$$

  where λ_f^nom is the nominal wavelength of the filter (12 μm, 25 μm, 60 μm, 100 μm). The normalisation is defined so that, if $\lambda\,F_\lambda(\star)$ ($= \nu\,F_\nu(\star)$) is constant, then

$$\frac{1}{N_f}\int F_\lambda(\star)\,T_{f,\lambda}\,\mathrm{d}\lambda = F_\lambda(\star, \lambda = \lambda_f^\mathrm{nom}). \tag{74}$$

  Currently used for IRAS filters only.

The dimension of $\overline{F}_f$ is that of F_ν if the type of calibration is ‹`"D4000"`›, and that of F_λ otherwise.

The ***in-band luminosity*** through filter f, $\overline{L}_f$, is defined in the same way as $\overline{F}_f$, but from the monochromatic luminosity (spectral power) L_λ or L_ν instead of the monochromatic flux F_λ or F_ν.

γ. Magnitudes

The magnitude m_f in filter f of an object is defined in one of the following ways, depending on the type of calibration:

- **‹`"Vega"`›:** standard photometric system based on Vega.

$$m_f(\star) := m_f^{\mathrm{std}}(\star) = -2.5\log_{10}\frac{\overline{F}_f(\star)}{\overline{F}_f(\mathrm{Vega})} + m_f^{\mathrm{std}}(\mathrm{Vega}), \tag{75}$$

  where $m_f^{\mathrm{std}}(\mathrm{Vega})$ is the apparent magnitude adopted for Vega in the standard system. One has

$$m_f^{\mathrm{std}}(\mathrm{Vega}) = \langle\texttt{std\_Vega\_def}\rangle + \textsc{zpo}_f, \tag{76}$$

  where ‹`std_Vega_def`› is the default apparent magnitude of Vega (set to 0.03 in «`source_dir/mod_filters_constants.f90`») and $\textsc{zpo}_f$ is a possible zero-point offset for filter f (it is read from the file containing the passband and currently set to 0 for all filters);

- **‹`"AB"`›:** AB system.

$$\begin{aligned} m_f(\star) := m_f^{\mathrm{AB}}(\star) &= -2.5\log_{10}\frac{\int F_\nu(\star)\,T_{f,\nu}\,\mathrm{d}\nu}{\mathrm{erg\,s^{-1}\,cm^{-2}\,Hz^{-1}}\int T_{f,\nu}\,\mathrm{d}\nu} - 48.60 \\ &= -2.5\log_{10}\frac{\overline{F}_f(\star)}{\mathrm{erg\,s^{-1}\,cm^{-2}\,\text{Å}^{-1}}} + 2.5\log_{10}\frac{\mathrm{Hz}\int T_{f,\lambda}\,\mathrm{d}\lambda}{\text{Å}\int T_{f,\nu}\,\mathrm{d}\nu} - 48.60 \end{aligned} \tag{77}$$

  because $T_{f,\nu} = T_{f,\lambda}$ and $F_\nu\,\mathrm{d}\nu = F_\lambda\,\mathrm{d}\lambda$;

- **‹`"TG"`›:** Thuan & Gunn system based on star BD +17° 4708.

$$m_f(\star) := m_f^{\mathrm{TG}}(\star) = -2.5\log_{10}\frac{\overline{F}_f(\star)}{\overline{F}_f(\mathrm{BD}\,{+17^\circ}\,4708)} + 9.50\,; \tag{78}$$

- **‹`"HST"`›:** HST system.

$$m_f(\star) := m_f^{\mathrm{HST}}(\star) = -2.5\log_{10}\frac{\overline{F}_f(\star)}{\mathrm{erg\,s^{-1}\,cm^{-2}\,\text{Å}^{-1}}} - 21.10. \tag{79}$$

  For a monochromatic filter, $m_f^{\mathrm{HST}} = m_f^{\mathrm{AB}}$;

- **‹`"FOCA"`›:** FOCA system.

$$m_f(\star) := m_f^{\mathrm{FOCA}}(\star) = -2.5\log_{10}\frac{\overline{F}_f(\star)}{\mathrm{erg\,s^{-1}\,cm^{-2}\,\text{Å}^{-1}}} - 21.175\,; \tag{80}$$

- **‹`"IRAS"`› and ‹`"D4000"`›:** magnitude not defined.

The following relations hold between these definitions:

$$m_f^{\mathrm{AB/TG}}(\star) = m_f^{\mathrm{std}}(\star) + (m_f^{\mathrm{AB/TG}}[\mathrm{Vega}] - m_f^{\mathrm{std}}[\mathrm{Vega}]). \tag{81}$$

b. Outputs of «`calib`»

After a few lines of general information, «`calib`» write in «`calib_dir/calib.txt`» a table containing, for each filter listed in «`calib_dir/list_filters.txt`» (one per line), the following quantities[53]:

- **‹`filter_id`›:** the filter's identifier (see Sec. V.3.*a.α*, p. 42) defined in the corresponding file of filter and used in «`colors`»;
- **‹`i_filter`›:** ordinal number of the filter in «`calib_dir/list_filters.txt`»;

[53] When the wavelength range covered by the SED of an object does not fully contain that of the filter's passband, or when no magnitude is defined given the type of calibration of the filter (e.g. for Balmer break and IRAS filters), the magnitude of the object is set to ‹`undefined_mag`› (99.999 by default; this parameter is defined in «`source_dir/mod_filters_constants.f90`»).

- ‹**flux_band_Vega**› : in-band flux $\overline{F}_f$(Vega), through filter f, of the light emitted by Vega and received on the Earth. The monochromatic flux of Vega on the Earth is read from file «calib_dir/Vega_BaSeL_¬SED.txt».
  In erg s^{-1} Hz^{-1} cm^{-2} if the type of calibration is ‹"D4000"›; otherwise, in erg s^{-1} Å^{-1} cm^{-2};
- ‹**filter_norm**› : normalisation N_f of the passband;
- ‹**lambda_mean**› : mean wavelength in Å of the filter (= λ'_0 in Bessell & Murphy (2012)), defined as

$$\text{‹lambda\_mean›} = \frac{\int \lambda\, T_{f,\lambda}\, \mathrm{d}\lambda}{\int T_{f,\lambda}\, \mathrm{d}\lambda}\,; \tag{82}$$

- ‹**lambda_eff_Vega**› : effective wavelength in Å of Vega.
  If the type of calibration is ‹"D4000"›, the effective wavelength is defined as

$$\text{‹lambda\_eff(Vega)›} = \frac{\int_{\lambda_1}^{\lambda_2} \lambda\, F_\nu(\text{Vega})\, \mathrm{d}\lambda}{\int_{\lambda_1}^{\lambda_2} F_\nu(\text{Vega})\, \mathrm{d}\lambda}\,. \tag{83}$$

Otherwise,

$$\text{‹lambda\_eff(Vega)›} = \frac{\int \lambda\, F_\lambda(\text{Vega})\, T_{f,\lambda}\, \mathrm{d}\lambda}{\int F_\lambda(\text{Vega})\, T_{f,\lambda}\, \mathrm{d}\lambda}\,; \tag{84}$$

- ‹**AB_Vega**› : AB apparent magnitude of Vega;
- ‹**TG_Vega**› : Thuan & Gunn apparent magnitude of Vega.
  The monochromatic flux of BD +17° 4708 on the Earth is read from file «calib_dir/BD+17d4708_¬SED.txt»;
- ‹**lum_band_Sun**› : in-band luminosity $\overline{L}_f^{\odot}$, through filter f, of $L_\lambda^{\odot}$, the monochromatic luminosity of the Sun per unit wavelength (read from file «calib_dir/Sun_BaSeL_SED.txt»).
  In erg s^{-1} Hz^{-1} if the type of calibration is ‹"D4000"›; otherwise, in erg s^{-1} Å^{-1};
- ‹**mag_abs_Sun**› : absolute magnitude of the Sun in the photometric system defined by the type of calibration. Computed from the monochromatic flux received at 10 pc from the Sun,

$$F_\lambda^{\odot} = \frac{L_\lambda^{\odot}}{4\,\pi\,(10\,\mathrm{pc})^2}\,; \tag{85}$$

- ‹**lambda_pivot**› : pivot wavelength of the filter, defined as

$$\text{‹lambda\_pivot›} = \sqrt{\frac{\int T_{f,\lambda}\, \mathrm{d}\lambda}{\int \lambda^{-2}\, T_{f,\lambda}\, \mathrm{d}\lambda}}\,. \tag{86}$$

  The mean values of F_λ and F_ν through the passband are related by

$$\frac{\int F_\nu\, T_{f,\nu}\, \mathrm{d}\nu}{\int T_{f,\nu}\, \mathrm{d}\nu} = \frac{\text{‹lambda\_pivot›}^2}{c}\, \frac{\int F_\lambda\, T_{f,\lambda}\, \mathrm{d}\lambda}{\int T_{f,\lambda}\, \mathrm{d}\lambda}\,; \tag{87}$$

- ‹**filter_width**› : width of the filter, defined as

$$\text{‹width›} = \sqrt{\frac{\int (\lambda - \text{‹lambda\_mean›})^2\, T_{f,\lambda}\, \mathrm{d}\lambda}{\int T_{f,\lambda}\, \mathrm{d}\lambda}}\,; \tag{88}$$

- ‹**calib_type**› : type of calibration.

c. Outputs of «**colors**»

α. Overview

Code «colors» processes a file of spectra (see Sec. I.3.*d*, p. 7) and writes in a file of colors, after a few lines of general information, a series of tables providing various quantities (one per column) as a function of the age of the galaxy. Other quantities may be added to a given table or new tables may be produced using the procedure and functions described hereafter. To change the output of «colors», edit file «source_dir/¬incl_colors.f90» and compile the code as explained in Sec. I.2, p. 4.

The first column in each table is the *j*-th age in Myr (variable ‹data % output_age(*j*)›). Other columns are added to a table through the Fortran statement “`call add_column(`*heading*`, `*quantity*⌈`, `*format*⌋`)`”,

where

- ‹*heading*› is a string of characters choosen by the user as heading for the column;
- ‹*quantity*› is an array of reals containing the value at all ages of the quantity written in the column;
- the optional ‹*format*› is a string defining the Fortran-style format, used to output the quantity at a single age. A default format is used if the value of ‹*format*› is not provided.

When a table is complete, the Fortran statement

“`call write_table`”

prints it in the file of colors, and the next table, if any, may then be built.

β. Quantities available in the file of spectra

By default, «`colors`» outputs only a subset of the quantities directly available in the file of spectra (quantities ‹`data % `*field*› in «`incl_colors.f90`»; see Table 8, p. 36 and, if the output of «`spectra`» is detailed, Table 9, p. 38). Two functions, ‹`ISM_abund`› and ‹`L_line`›, have also been created to ease the access to the following quantities (through a string of characters instead of an integer index[※54]):

- **Interstellar medium abundances**. The abundance of the i-th element in the interstellar medium at age t = ‹`data % output_age(`j`)`› is given by

$$‹\texttt{ISM\_abund(}j\texttt{, }\textit{elem\_id}\texttt{)}› \coloneqq ‹\texttt{data \% ISM\_abund(}j\texttt{, }i\texttt{)}›, \tag{89}$$

  where the identifier ‹*elem_id*› is a string holding the value of ‹`data % elem_id(`i`)`›;
- **Emission line luminosities**. The normalized luminosity of the ℓ-th emission line at age t is given by

$$‹\texttt{L\_line(}j\texttt{, }\textit{line\_id}\texttt{)}› \coloneqq ‹\texttt{data \% L\_line(}j\texttt{, }\ell\texttt{)}›, \tag{90}$$

  where the identifier ‹*line_id*› is a string holding the value of ‹`data % line_id(`ℓ`)`›. All lines are listed with their identifier and their wavelength in «`Cloudy_dir/list_neb_lines.txt`».

For convenience, the functions ‹`ISM_abund`› and ‹`L_line`› are overloaded: to access the arrays containing the values of ‹`ISM_abund(`j`, `*elem_id*`)`› and ‹`L_line(`j`, `*line_id*`)`› at all ages, use ‹`ISM_abund(`*elem_id*`)`› and ‹`L_line(`*line_id*`)`› instead (as in the “`call add_column`” statements).

γ. Other quantities

Besides the quantities already present in the file of spectra, «`colors`» also outputs the bolometric magnitude of the galaxy, its in-band luminosities and absolute magnitudes through various filters, several colors, Balmer break indices and the equivalent width of some emission lines. These quantities are defined as follows in «`colors.f90`»:

- **Bolometric magnitude**. The normalized bolometric magnitude of the galaxy at age t is given by

$$‹\texttt{M\_bol(}j\texttt{)}› = 4.75 - 2.5\log_{10}\frac{\widehat{\mathcal{L}}_{\text{bol}}(t)\,M_\odot}{\mathcal{L}_\odot}\,. \tag{91}$$

  For a galaxy with a system's mass M_{sys}, the (unnormalized) bolometric magnitude would be

$$\mathcal{M}_{\text{bol}}(t) = ‹\texttt{M\_bol(}j\texttt{)}› - 2.5\log_{10}\frac{M_{\text{sys}}}{M_\odot}\,; \tag{92}$$

- **In-band luminosities**. The normalized in-band luminosity of the galaxy in filter f at age t (in erg $\text{s}^{-1}\,\text{Hz}^{-1}\,\text{M}_\odot^{-1}$ if the type of calibration is ‹`"D4000"`› and in erg $\text{s}^{-1}\,\text{Å}^{-1}\,\text{M}_\odot^{-1}$ otherwise) is

$$‹\texttt{lum\_band(}j\texttt{, }\textit{filter\_id}\texttt{)}› = \widehat{\overline{L}}_f(t), \tag{93}$$

  where ‹*filter_id*› is the identifier of filter f in the corresponding file of filter and in «`calib.txt`», and $\overline{L}_f$ is defined in Sec. V.3.$a.\beta$, p. 42.

  The normalized in-band luminosity of the galaxy relative to that of the Sun, in $M_\odot^{-1}$, is given by

$$‹\texttt{lum\_band\_rel\_Sun(}j\texttt{, }\textit{filter\_id}\texttt{)}› = \frac{\widehat{\overline{L}}_f(t)}{\overline{L}_f^\odot}\,; \tag{94}$$

[54] The value of the integer corresponding to a given string is computed by a hash function.

- **Absolute magnitudes**. The normalized absolute magnitude[※55] of the galaxy in filter f, ‹`magnit(`*j*`, `*`filter_id`*`)`›, is given by one of the equations in Sec. V.3.$a.\gamma$, p. 43, depending on the type of calibration, with

$$\overline{F}_f(t) = \frac{\widehat{\overline{L}}_f(t)\, M_\odot}{4\,\pi\,(10\ \mathrm{pc})^2}\,. \tag{95}$$

  For a galaxy with a system's mass M_{sys}, the (unnormalized) absolute magnitude in filter f would be

$$\mathcal{M}_f(t) = \langle\texttt{magnit(}j\texttt{, }\textit{\texttt{filter\_id}}\texttt{)}\rangle - 2.5\log_{10}\frac{M_{\mathrm{sys}}}{M_\odot}\,; \tag{96}$$

- **Colors**. Let m_1 and m_2 denote either the apparent or the absolute magnitudes of the galaxy in filters f_1 and f_2. The color[※55] of the galaxy at age t is given by

$$\langle\texttt{color(}j\texttt{, }\textit{\texttt{filter1\_id}}\texttt{, }\textit{\texttt{filter2\_id}}\texttt{)}\rangle = m_1(t) - m_2(t), \tag{97}$$

  where the strings ‹*`filter1_id`*› and ‹*`filter2_id`*› are the identifiers of the filters f_1 and f_2 in the corresponding files of filter and in «`calib.txt`»;

- **Balmer break indices**. The indices $D4000$ and $D_{\mathrm{n}}4000$ (‹`D4000`› and ‹`Dn4000`› in «`colors.f90`») are given by

$$\langle\texttt{D}\ulcorner\texttt{n}\urcorner\texttt{4000(}j\texttt{)}\rangle = \frac{\int_{\lambda_1^+}^{\lambda_2^+}\widehat{L}_\nu(t)\,\mathrm{d}\lambda}{\int_{\lambda_1^-}^{\lambda_2^-}\widehat{L}_\nu(t)\,\mathrm{d}\lambda}\,, \tag{98}$$

  with

$$(\lambda_1^-, \lambda_2^-, \lambda_1^+, \lambda_2^+) = \begin{cases}(3750, 3950, 4050, 4250)\ \text{Å} & \text{for } D4000 \text{ (Bruzual 1983)},\\ (3850, 3950, 4000, 4100)\ \text{Å} & \text{for } D_{\mathrm{n}}4000 \text{ (Balogh et al. 1999)};\end{cases} \tag{99}$$

- **Equivalent width of emission lines**. The equivalent width in Å of the ℓ-th emission line at age t is given by

$$\langle\texttt{eq\_width(}j\texttt{, }\textit{\texttt{line\_id}}\texttt{)}\rangle = \frac{\langle\texttt{L\_line(}j\texttt{, }\textit{\texttt{line\_id}}\texttt{)}\rangle}{\widehat{L}_{\mathrm{cont}}(t)}\,, \tag{100}$$

  where ‹*`line_id`*› is a string holding the value of ‹`data % line_id(`ℓ`)`›, and $\widehat{L}_{\mathrm{cont}}(t)$ is the normalized monochromatic luminosity of the galaxy at age t and wavelength ‹`lambda_line(`ℓ`)`› in the continuum; $\widehat{L}_{\mathrm{cont}}(t)$ is estimated by linear interpolation from the values of ‹`lum_cont(`*j*`, `*k*`)`› at adjacent wavelengths.

By definition, colors, Balmer break indices and equivalent widths do not depend on the normalization to the mass of the system.

The quantities ‹`lum_band(`*j*`, `*`filter_id`*`)`›, ‹`magnit(`*j*`, `*`filter_id`*`)`›, ‹`color(`*j*`, `*`filter1_id`*`, `*`filter2_id`*`)`› and ‹`eq_width(`*j*`, `*`line_id`*`)`› are provided by overloaded functions: to access the arrays containing their values at all ages, use ‹`lum_band(`*`filter_id`*`)`›, ‹`magnit(`*`filter_id`*`)`›, ‹`color(`*`filter1_id`*`, `*`filter2_id`*`)`› and ‹`eq_width(`*`line_id`*`)`› instead.

[55] Magnitudes and colors are set to ‹`undefined_mag`› (see Footnote 53, p. 43) if they may not be computed.

Appendices

A. Typographical conventions

The following typographical conventions are used in this documentation:

‹xyz›	Code text or shell command; also used for e-mail addresses.[※56]
«xyz»	File or directory name; also used for Web addresses.[※51]
⌞xyz⌝	Optional text.
¬	Symbol used to show that a string of characters has been hyphenated.
For dialogs with the codes:	
◄ xyz⌞↵⌝	Text input from the keyboard (once the Return/Enter key is pressed). The symbol ↵ is always written if the text is empty;
► xyz	Text printed on the terminal.
[═══]	Skipped lines.

The non-ascii symbols ⌞, ⌝, ¬, ↵, ◄, ►, ‹, ›, «, » and [═══] should not be typed by the user!

Placeholders (i.e. names of dummy files, statements, keywords or values which the user must replace by actual ones) are written in slanted characters.

B. Syntax used in files of scenarios

1. Syntax used for all statements

Statements are written in a Fortran 90-like manner:

- **Line length and continuation character**. A line cannot be longer than 132 characters.
  A statement can be continued on the following line by appending an ampersand (‹&›) to the current line. It the line is cut in the middle of a string of characters, a keyword or a numerical or boolean value, an ampersand must also appear at the beginning of the continuation line;
- **Multiple statements**. Multiple statements, separated by semi-colons, can be put on the same line;
- **End-of-line comments**. On a line, anything following an exclamation mark is considered to be a comment and is skipped, unless the exclamation mark is within a string of characters.
  Blank lines are considered as comments;
- **Block comments**. It is also possible to comment sections of the file in a C-language manner: anything between ‹/*› and ‹*/› is skipped, unless ‹/*› is within a string of characters. Block comments are not nested;
- **Strings of characters**. Strings of characters are enclosed either in pairs of quotes (‹"›) or of apostrophes (‹'›).
  In a string enclosed in quotes, a quote character may be obtained by typing it twice (‹""›). In a string enclosed in apostrophes, an apostrophe character may be obtained by typing it twice (‹''›);
- **Case of letters**. The case of letters is not significant, except in character strings;
- **Spaces and tabulations**. Space characters are not significant, except in character strings.
  Tabulations are treated as spaces.

2. Syntax used for assignments

A file of scenarios contains a sequence of statements, most of them of the form "*key*⌞(*indices*)⌝ = *val*", where ‹*key*› is the name of a parameter, ‹*val*› is either a scalar value or an array of scalar values assigned to this parameter, and the optional ‹(*indices*)› provides the indices for which this assignment is done.

Depending on the parameter considered, a scalar value is either

- an integer;

[51] ‹, ›, « and » are omitted in strings of characters mentioned (denoted by " ") rather than used.

- a real;
- a boolean, ‹`.true.`› or ‹`.false.`› (the case of the characters used here does not matter);
- a string of characters delimited by quotes or apostrophes.

a. Parameters: scalars and arrays

Depending on the parameter considered, ‹*key*⌜`(`*indices*`)`⌟› may either be a scalar or an array:

- If it is necessarily a scalar, ‹`(`*indices*`)`› *must not* be present;
- Else, ‹*key*⌜`(`*indices*`)`⌟› *may* be an array.
  If "`(`*indices*`)`" is present, ‹*indices*› defines the indices for which ‹*key*› is defined by ‹*val*›. Unless ‹*indices*› refers to a single index (see below), ‹*key*`(`*indices*`)`› is an array.
  If the "`(`*indices*`)`" part is absent, then ‹*key*› is a scalar and writing "*key* `=`" is the same as writing "*key*`(1) =`". Note that this is not the usual convention in Fortran.

b. Indices

Possible forms of ‹*indices*› are the following:

- A *single* integer *index*, ‹*i*›;
- An *index loop*:
  - ‹f`:`l›: all the integers $i \in [\![f,\ l]\!]$ (the set of indices is void if $f > l$).
    Note that ‹*key*`(`i`:`i`)`› is considered as an array of parameters while ‹*key*`(`i`)`› is a scalar one, although both refer to the same object;
  - ‹f`:`l`:`s›: all the integers $i = f + j\,s$, where $j \in [\![0,\ n]\!]$ and $n = \lfloor (l - f)/s \rfloor$. ("$\lfloor x \rfloor$" denotes the integer part (floor function) of x).
    f, l and s are integers. The stride s must be different from 0. The signs of $l - f$ and s must be the same; otherwise, the set of indices is void.

  Other notations, such as ‹*key*`(:)`› or ‹*key*`(:`f`:)`›, although standard in Fortran, are not allowed in this code;
- A *list of single indices* or *index loops*,

  ‹`[`*indices*$_1$`,` *indices*$_2$`,` $\cdots$`]`›,

  where ‹*indices*$_1$›, ‹*indices*$_2$›, etc., are either single indices or index loops. An index can appear only once in a list.

All the indices must be larger than or equal to 1. The maximal upper bound depends on the parameter considered.

c. Values

Possible forms of ‹*val*› are the following:

- a *single scalar value*, ‹*v*›.
  If ‹`(`*indices*`)`› is present, then

  $$\forall\ i \in \text{‹}\mathtt{indices}\text{›}, \quad \text{‹key›}(i) = v.$$

  If not, then

  $$\text{‹key›}(1) = v;$$

- an *array of scalar values* of the same type, ‹`[`v_1`,` $\cdots$ `,` v_n`]`›, n being the number of indices in ‹*indices*›.
  In this case, the ‹`(`*indices*`)`› part must be present and must not consist in a single index. If ‹*indices*› contains indices $j_1,\ \cdots\ ,\ j_n$ in that order, then

  $$\forall\ i \in [\![1,\ n]\!], \quad \text{‹key›}(j_i) = v_i.$$

d. An example

```
SF_type = "ISM_mass"
SF_type(3:6) = "ISM_mass"
```

```
3 SF_type(2:4:2) = "exponential"
4 SF_expo_timescale(4:1:-2) = [2000, 500]
5 SF_ISM_timescale([1:5]) = 1000
6 SF_type(2) = "exponential"
7 SF_type([3, 5:6]) = "none"
8 SF_begin_time(2:4) = 1000
9 SF_end_time([1, 4, 3]) = [500, 2000, 500]
```

Let E_i denote the i-th star formation episode.

- **Line 1** : ‹`SF_type`› is set to ‹`"ISM_mass"`› for E_1 (SFR proportional to the ISM mass);
- **Line 2** : The same for E_3 to E_6;
- **Line 3** : ‹`SF_type`› is set to ‹`"exponential"`› for E_2 (SFR exponentially decreasing or increasing with age); ‹`SF_type`› is also changed to ‹`"exponential"`› for E_4;
- **Line 4** : ‹`SF_expo_timescale(2)`› is set to 500 Myr and ‹`SF_expo_timescale(4)`› to 2000 Myr;
- **Line 5** : Parameter ‹`SF_ISM_timescale`› is set to 1000 Myr for episodes E_1 to E_5, which sets ‹`SF_type`› to ‹`"ISM_mass"`› for E_2 and E_4 (‹`SF_ISM_timescale`› is still undefined for E_6);
- **Line 6** : ‹`SF_type`› is set back to ‹`"exponential"`› for E_2 and ‹`SF_expo_timescale`› recovers its previous value (500 Myr);
- **Line 7** : Star formation is finally cancelled for E_3, E_5 and E_6;
- **Line 8** : Star formation begins at 1 Gyr for E_2, E_3 and E_4;
- **Line 9** : Star formation ends at 500 Myr for E_1 and E_3 and at 2 Gyr for E_4.

The final state is the following:

- E_1 begins at 0 Myr (default value) and ends at 500 Myr. ‹`SF_type(1)`› = ‹`"ISM_mass"`› and ‹`SF_ISM_timescale(1)`› = 1000 Myr;
- E_2 and E_4 are consecutive to E_1: both begin at 1000 Myr and overlap until 2000 Myr, at which age E_4 ends; E_2 ends at 20000 Myr (default). ‹`SF_type(2)`› = ‹`"exponential"`› and ‹`SF_expo_timescale(2)`› = 500 Myr. ‹`SF_type(4)`› = ‹`"ISM_mass"`› and ‹`SF_ISM_timescale(4)`› = 1000 Myr;
- E_3 does not form stars since ‹`SF_end_time(3)`› = 500 Myr < ‹`SF_begin_time(3)`› = 1000 Myr;
- E_5 and E_6 do not form stars either.

C. Playing with IMFs

⟶ Sec. III.1, p. 16.

1. Adding other IMFs

Let us first remind that, in the code, an IMF ϕ is defined by

$$\phi(m) := \mathrm{d}n/\mathrm{d}(\ln m), \tag{101}$$

where m is the initial mass of a star and dn is the number of stars, per unit of initial mass of the SSP, born with a mass in $[m, m + \mathrm{d}m[$.

You may use other IMFs besides the ones already provided with the code (see Sec. III.1, p. 16). Any additional IMF must be a continuous, piecewise power-law function. More precisely,

- ϕ is continuous for all m in some interval $[m_{\mathrm{inf}}, m_{\mathrm{sup}}]$ (with $0 < m_{\mathrm{inf}} < m_{\mathrm{sup}}$) and null outside of this interval. To be in agreement with the default stellar evolutionary tracks, it is recommended to take $m_{\mathrm{inf}} \geqslant 0.09\,M_\odot$ and $m_{\mathrm{sup}} \leqslant 120\,M_\odot$;
- $[m_{\mathrm{inf}}, m_{\mathrm{sup}}]$ is divided in p ($\geqslant 1$) subintervals $[m_i, m_{i+1}]$, with $m_1 = m_{\mathrm{inf}}$, $m_{p+1} = m_{\mathrm{sup}}$ and $m_i < m_{i+1}$ for all $i \in \llbracket 1,\ p \rrbracket$;
- $$\forall\ m \in [m_i, m_{i+1}], \quad \phi(m) \propto m^{s_i}\,, \tag{102}$$

  where the slope s_i is a constant.

To define a new IMF, create an IMF file containing the following informations[52]:

- zero or more comment lines (blank lines and lines where the first non blank character is “ ! ”) providing, for instance, the reference paper for this IMF;

[52] Have a look to «`IMFs_dir/IMF_Kroupa.txt`» for example.

- p lines containing, for all integers i from 1 to p, the value, in solar masses, of the bottom mass m_i of the i-th mass bin and of the slope s_i in this bin. These values should be separated by one or more space characters;
- a last line containing the value of m_{p+1} in solar masses.

Finally, add a line containing the name of this file to «IMFs_dir/list_IMFs.txt».

The values of m_{inf} and m_{sup} given in the IMF file may be superseded when running «SSPs».

Code «SSPs» will automatically ensure that, for all $i \in [\![2, p]\!]$, the IMF is continuous at m_i and that the IMF is normalized, i.e.

$$\int_{m=m_{\text{inf}}}^{m_{\text{sup}}} m\,\phi(m)\,\mathrm{d}(\ln m) = 1. \tag{103}$$

2. "Evolving" the IMF

The stellar initial mass function is constant in the code. It is however possible to mimic its evolution as a function of the ISM metallicity by mixing several sets of SSPs, as long as they use the same library of stellar spectra and the same output ages. As an example, let us consider two sets of SSPs, «*Salpeter*_SSPs.txt» and «*RB*_SSPs.txt», respectively computed by «SSPs» for the IMFs of Salpeter (1955) and Rana & Basu (1992) (from the files «IMF_Salpeter.txt» and «IMF_Rana_Basu.txt» in «IMF_dir/»). The first file will look like this:

```
[≡≡≡]
Salpeter_tracks_Z0.008+.txt
Salpeter_tracks_Z0.02+.txt
[≡≡≡]
```

The second file will look the same, with "*RB*" instead of "*Salpeter*".

To implement a *smooth* transition from the Salpeter's IMF to the Rana & Basu's one between $Z = 0.008$ and $Z = 0.02$, create a file containing all the lines in «*Salpeter*_SSPs.txt» up to, and *excluding*, "*Salpeter*_tracks_Z0.02+.txt", followed by all the lines in «*RB*_SSPs.txt» from, and *including*, "*RB*_tracks_Z0.02+.txt".

To implement a *sudden* transition from the Salpeter's IMF to the Rana & Basu's one at $Z = 0.02$, create a file containing all the lines in «*Salpeter*_SSPs.txt» up to, and *including*, "*Salpeter*_tracks_Z0.02+.txt", followed by all the lines in «*RB*_SSPs.txt» from, and *including*, "*RB*_tracks_Z0.02+.txt".

D. Additional notes on the modeling

1. Relating normalized quantities to observed ones

To clarify the meaning of normalized quantities and their relation to observed ones, let us take the following example. Consider a galaxy observed at wavelengths λ' in the range $[\lambda'_-, \lambda'_+]$ with an intensity (= spectral irradiance) $F^{\text{obs}}_{\lambda'}$. Let z be the redshift of the galaxy and D its luminosity distance to the observer. The monochromatic luminosity of the galaxy in the emitter's frame is then $L^{\text{emit}}_{\lambda} = 4\,\pi\,D^2\,(1 + z)\,F^{\text{obs}}_{\lambda'}$, where λ' is related to the emission wavelength λ by $\lambda' = (1 + z)\,\lambda$.

Denote by $\widehat{L}^{\text{mod}}_{\lambda}$ (quantity ‹lum_cont› in Table 8, p. 36) the normalized monochromatic luminosity of a model fitting best the *shape* of the emitted spectrum on $[\lambda_-, \lambda_+] := [\lambda'_-, \lambda'_+]/(1 + z)$. For instance, the best model may minimize

$$\chi^2 = \min_{k>0} \int_{\lambda=\lambda_-}^{\lambda_+} \left(L^{\text{emit}}_{\lambda} - k\,\widehat{L}^{\text{mod}}_{\lambda}\right)^2 \mathrm{d}\lambda. \tag{104}$$

One has then $L^{\text{emit}}_{\lambda} \approx k\,\widehat{L}^{\text{mod}}_{\lambda}$ on $[\lambda_-, \lambda_+]$ for the (model-dependent) scalar k minimizing above integral. Since $L^{\text{mod}}_{\lambda} = M_{\text{sys}}\,\widehat{L}^{\text{mod}}_{\lambda}$, one has $L^{\text{mod}}_{\lambda} \approx L^{\text{emit}}_{\lambda}$ on $[\lambda_-, \lambda_+]$ if one takes k as an estimate of M_{sys}.

All unnormalized quantities have to be multiplied by this factor to obtain the "real" value. For instance, an estimate of the mass of live stars in the galaxy is M_{sys} × ‹live_stars_mass› (see Table 8, p. 36).

2. Using stellar spectra available on too narrow a wavelength range

The approach described in Sec. II.2, p. 9, to compute $\ell^{\text{lib}}_{\lambda,i}$ ("ℓ_λ" below) is appropriate if the contribution to the bolometric luminosity of radiations emitted outside of the wavelength range covered by the spectrum is negligible. This is the case for the libraries of theoretical spectra we use.

To compute ℓ_λ from spectra covering a narrower wavelength range $[\lambda_-, \lambda_+]$, e.g. observed spectra, one needs the ***bolometric correction*** BC_f of the star in some filter f. The passband transmission of f, $T_{f,\lambda}$, must be null outside of $[\lambda_-, \lambda_+]$. Then,

$$\ell_\lambda = \frac{F_\lambda}{\overline{F}_f} \frac{\overline{F}_f}{\mathcal{F}_\text{bol}}, \tag{105}$$

where F_λ is the spectral irradiance of the star at wavelength λ, $\overline{F}_f := \int F_\lambda\, T_{f,\lambda}\, \text{d}\lambda / \int T_{f,\lambda}\, \text{d}\lambda$ is its average value in filter f, and $\mathcal{F}_\text{bol}$ is the (bolometric) irradiance of the star. The ratio $F_\lambda/\overline{F}_f$ is derived directly from the spectrum. To compute the other factor, $\overline{F}_f/\mathcal{F}_\text{bol}$, one uses the following relations:

- $m_f = -2.5 \log_{10}(\overline{F}_f/\overline{F}_f^\text{ref}) + m_f^\text{ref},$ (106)

  where m_f is the apparent magnitude of the star in filter f, m_f^ref is that of some reference source (real, e.g. Vega, or virtual), and $\overline{F}_f^\text{ref}$ is the value of $\overline{F}_f$ for the latter;

- $m_\text{bol} = m_f + \text{BC}_f\,,$ (107)

  where m_bol is the apparent bolometric magnitude of the star;

- $m_\text{bol} - \mathcal{M}_\text{bol} = 5 \log_{10}(D/[10\,\text{pc}]),$ (108)

  where $\mathcal{M}_\text{bol}$ is the absolute magnitude of the star, and D is its distance;

- $\mathcal{M}_\text{bol} = \mathcal{M}_\odot - 2.5 \log_{10}(\mathcal{L}_\text{bol}/\mathcal{L}_\odot),$ (109)

  where $\mathcal{M}_\odot = 4.74$ is the absolute bolometric magnitude of the Sun (Mamajek et al. 2015), $\mathcal{L}_\text{bol}$ is the bolometric luminosity of the star, and $\mathcal{L}_\odot$ is that of the Sun;

- $\mathcal{F}_\text{bol} = \mathcal{L}_\text{bol}/(4\,\pi\, D^2).$ (110)

Combining these equations, one obtains

$$\frac{\overline{F}_f}{\mathcal{F}_\text{bol}} = \frac{4\,\pi\,(10\,\text{pc})^2\, \overline{F}_f^\text{ref}}{\mathcal{L}_\odot} 10^{(\text{BC}_f - \mathcal{M}_\odot)/2.5}. \tag{111}$$

3. Self-absorption

One can also obtain Eq. (38), p. 16, as the limit of a sequence of iterations: the term of order 0 is the original light emitted by stars and the ionized gas, attenuated by dust; at first order, one takes into account the attenuation of energy emitted by grains in the first processing of the original photons; at second order, one also considers the attenuation of the energy emitted by grains in the second processing, and so on. So,

$$L_\lambda^{+\text{SA}} = \overline{\Theta}_\lambda\, L_\lambda^0 + \sum_{k=1}^{\infty} L_\lambda^{\text{d},k}, \tag{112}$$

where $L_\lambda^{\text{d},k}$ is the transmitted contribution from the k-th processing. With the two approximations made to model self-absorption, one has

$$L_\lambda^{\text{d},k} = \overline{\Theta}_\lambda^{\,\gamma}\, L_\lambda^{\text{d},k,0}, \tag{113}$$

where $L_\lambda^{\text{d},k,0}$ is the unattenuated contribution from the k-th processing and

$$L_\lambda^{\text{d},k,0} = \beta_k\, L_\lambda^{\text{d},-\text{SA}}. \tag{114}$$

The energy absorbed by grains must be re-emitted, so

$$\int L_\lambda^{\text{d},k+1,0}\, \text{d}\lambda = \int \left(1 - \overline{\Theta}_\lambda^{\,\gamma}\right) L_\lambda^{\text{d},k,0}\, \text{d}\lambda. \tag{115}$$

Therefore,

$$\beta_{k+1} \int L_\lambda^{\text{d},-\text{SA}}\, \text{d}\lambda = \beta_k \int \left(1 - \overline{\Theta}_\lambda^{\,\gamma}\right) L_\lambda^{\text{d},-\text{SA}}\, \text{d}\lambda \tag{116}$$

and, since $\beta_1 = 1$,

$$\beta_k = \beta_2^{k-1} \quad \text{with} \quad \beta_2 = \frac{\int \left(1 - \overline{\Theta}_\lambda^{\,\gamma}\right) L_\lambda^{\mathrm{d},-\textsc{sa}}\,\mathrm{d}\lambda}{\int L_\lambda^{\mathrm{d},-\textsc{sa}}\,\mathrm{d}\lambda}\,. \tag{117}$$

One finally obtains

$$\begin{aligned} L_\lambda^{+\textsc{sa}} &= \overline{\Theta}_\lambda\, L_\lambda^0 + \sum_{k=1}^{\infty} \overline{\Theta}_\lambda^{\,\gamma}\, \beta_2^{k-1}\, L_\lambda^{\mathrm{d},-\textsc{sa}} = \overline{\Theta}_\lambda\, L_\lambda^0 + \frac{\overline{\Theta}_\lambda^{\,\gamma}}{1-\beta_2}\, L_\lambda^{\mathrm{d},-\textsc{sa}} \\ &= \overline{\Theta}_\lambda\, L_\lambda^0 + \alpha\, \overline{\Theta}_\lambda^{\,\gamma}\, L_\lambda^{\mathrm{d},-\textsc{sa}}. \end{aligned} \tag{118}$$

E. Trees and tables of parameters

1. Preliminaries

This section contains the following additional informations on the parameters defining a scenario:

- **Trees** connecting related parameters (see Fig. 1 to 7, p. 53 to 64). In these trees, the names of parameters are framed, but not their values. The values of parameters are shown only if the extension of the set of values is finite. (This typically excludes numbers and arbitrary strings such as filenames.)

  The name of parameters having a default value is marked with the symbol ‡. When values are shown, the default value is also marked with ‡.

  A ***node*** consists in a parameter name and its associated value. Nodes are arranged in descending order from the left to the right. Every node in a tree may have zero, one or more children (the nodes connected to their right side in trees), but has at most one parent (the node connected to their left side in trees). Arrows point from the name a parameter to the value of its parent node.

  We recall here the two rules already stated in Sec. IV.1, p. 19:

  1. **Explicitely setting the value of a parameter implicitely assigns all its ancestors to consistent values**[53]**;**
  2. **Parameters which are not assigned in a scenario, whether explicitely or implicitely, retain the values they had in the previous scenario or remain undefined if they have no default value and have never been assigned.** (See Footnote 33, p. 19, for exceptions.)

- **Tables** providing the physical unit, type, range and default value (if any) of all parameters. See Tables 14 to 22, p. 53 to 65.

  Default values are usually either natural values, typical values, or simple values chosen to avoid forcing the user to make choices for other associated parameters... Values are usually adapted to the case where there is only one episode.

These informations complete those already provided in Sec. IV.3, p. 21, to Sec. IV.5, p. 32.

2. Cosmology

⟶ Sec. IV.3.*a*, p. 21.

Table 13. Parameters related to cosmology.

Parameter	Unit	Type	Possible values	Default value
‹Omega_m›	∅	real	$\in [0,1]$	0.308 [54]
‹H_0›	$\mathrm{km\,s^{-1}\,Mpc^{-1}}$	real	$\geqslant 0$	67.8 [54]
‹form_redshift›	∅	real	$\geqslant 0$	10
‹CBR›	∅	boolean	‹.true.›, ‹.false.›	‹.false.›

[53] So, these trees are more like drainage/river systems, with downstream on the left and upstream on the right.

[54] See Planck Collaboration et al. (2016).

3. Single stellar populations and chemical evolution parameters

⟶ Sec. IV.3.*b*, p. 21, and Sec. IV.3.*c*, p. 21.

Table 14. Single stellar populations and chemical evolution.

Parameter	Unit	Type	Possible values	Default value
‹SSPs_set›	∅	string		undefined
‹ISM_init_Z›	∅	real	∈ [0, 1]	0
‹close_bin_frac›	∅	real	∈ [0, 1]	0.05

4. Reservoirs and infall

⟶ Sec. II.1.*a*, p. 8, and Sec. IV.3.*d*, p. 22.

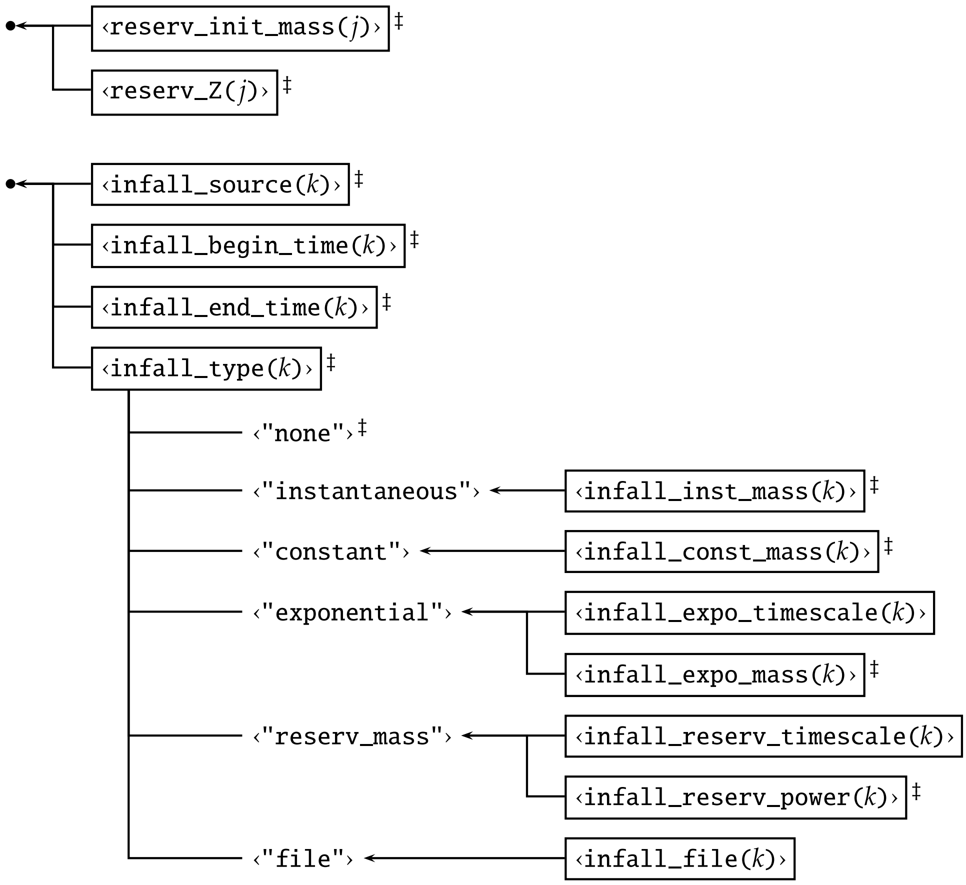


Figure 1. Trees of parameters related to the *j*-th reservoir and to the *k*-th episode of infall.

Table 15. Reservoir and infall parameters.

Parameter	Unit	Type	Possible values	Default value
‹reserv_init_mass›	∅	array of reals	$\in [0,1]$	0
‹reserv_Z›	∅	array of reals	$\in [0,1]$	0
‹infall_source›	Myr	array of reals	$\in [0,20000]$	1
‹infall_begin_time›	Myr	array of reals	$\in [0,20000]$	0
‹infall_end_time›	Myr	array of reals	$\in [0,20000]$	20000
‹infall_type›	∅	array of strings	‹"none"›, ‹"instantaneous"›, ‹"constant"›, ‹"exponential"›, ‹"reserv_mass"›, ‹"file"›	‹"none"›
‹infall_inst_mass›	∅	array of reals	$\geqslant 0$	1
‹infall_const_mass›	∅	array of reals	$\geqslant 0$	1
‹infall_expo_timescale›	Myr	array of reals	$\neq 0$	undefined
‹infall_expo_mass›	∅	array of reals	$\geqslant 0$	1
‹infall_reserv_timescale›	Myr	array of reals	> 0	undefined
‹infall_reserv_power›	∅	array of reals		1
‹infall_file›	∅	array of strings		undefined

5. Star formation

⟶ Sec. II.3, p. 10, and Sec. IV.3.*e*, p. 23.

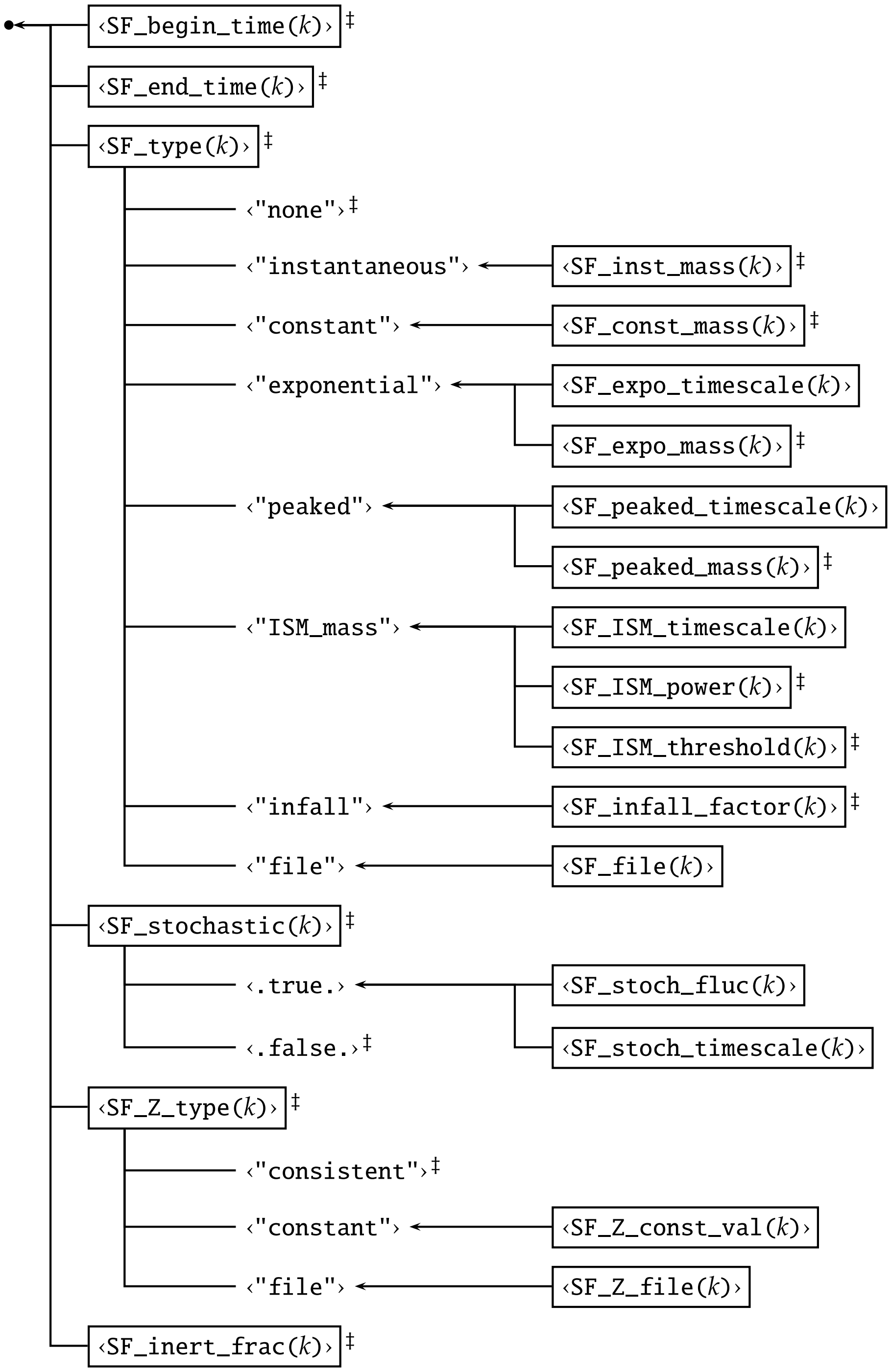


Figure 2. Tree of parameters related to the *k*-th episode of star formation.

Table 16. Star formation parameters.

Parameter	Unit	Type	Possible values	Default value
‹SF_begin_time›	Myr	array of reals	$\in [0, 20000]$	0
‹SF_end_time›	Myr	array of reals	$\in [0, 20000]$	20000
‹SF_type›	∅	array of strings	‹"none"›, ‹"instantaneous"›, ‹"constant"›, ‹"exponential"›, ‹"peaked"›, ‹"ISM_mass"›, ‹"infall"›, ‹"file"›	‹"none"›
‹SF_inst_mass›	∅	array of reals	$\geqslant 0$	1
‹SF_const_mass›	∅	array of reals	$\geqslant 0$	1
‹SF_expo_timescale›	Myr	array of reals	$\neq 0$	undefined
‹SF_expo_mass›	∅	array of reals	$\geqslant 0$	1
‹SF_peaked_timescale›	Myr	array of reals	> 0	undefined
‹SF_peaked_mass›	∅	array of reals	$\geqslant 0$	1
‹SF_ISM_timescale›	Myr	array of reals	> 0	undefined
‹SF_ISM_power›	∅	array of reals		1
‹SF_ISM_threshold›	∅	array of reals	$\in [0, 1]$	0
‹SF_infall_factor›	∅	array of reals	$\geqslant 0$	1
‹SF_file›	∅	array of strings		undefined
‹SF_stochastic›	∅	array of booleans	‹.true.›, ‹.false.›	‹.false.›
‹SF_stoch_fluc›	∅	array of reals		undefined
‹SF_stoch_timescale›	Myr	array of reals	> 0	undefined
‹SF_Z_type›	∅	array of strings	‹"consistent"›, ‹"constant"›, ‹"file"›	‹"consistent"›
‹SF_Z_const_val›	∅	array of reals	$\in [0, 1]$	undefined
‹SF_Z_file›	∅	array of strings		undefined
‹SF_inert_frac›	∅	array of reals	$\in [0, 1]$	0

6. Galactic outflows

⟶ Sec. II.1.*a*, p. 8, and Sec. IV.3.*f*, p. 25.

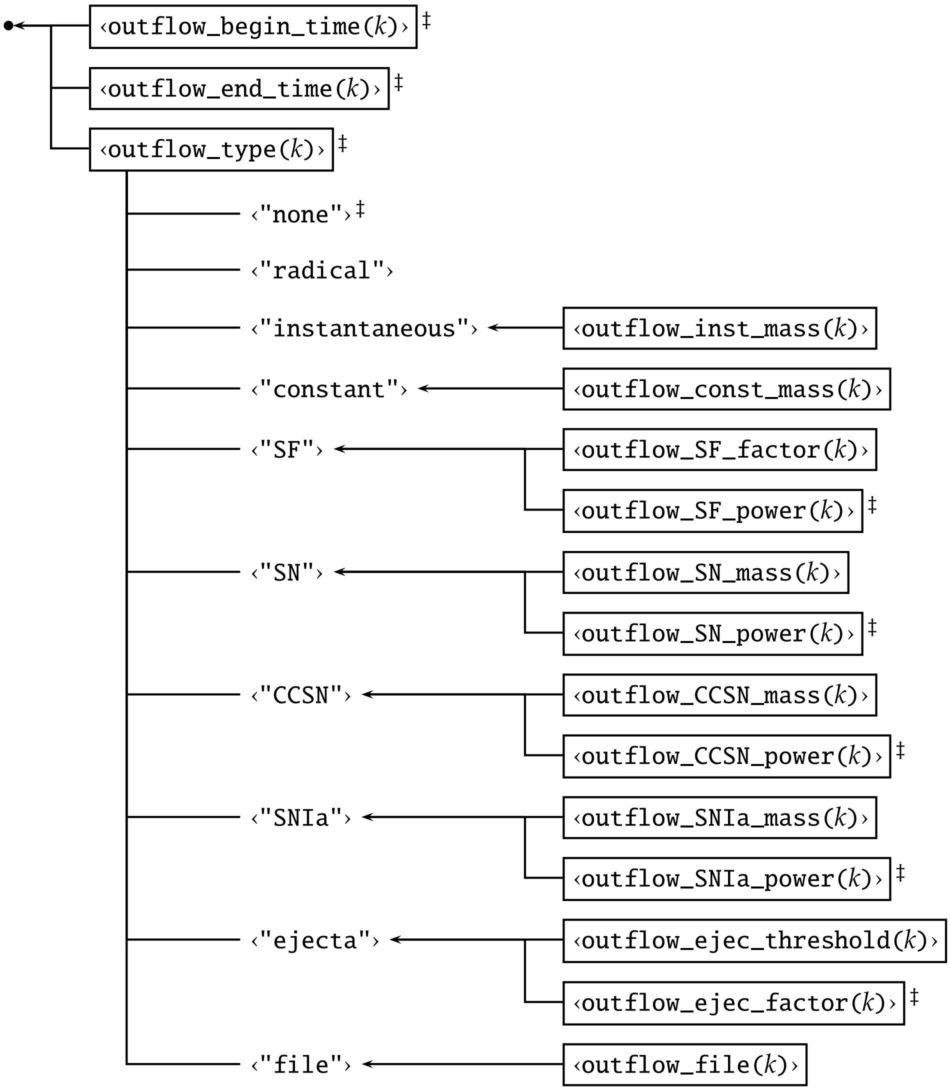


Figure 3. Tree of parameters related to the *k*-th episode of galactic outflow.

Table 17. Outflow parameters.

Parameter	Unit	Type	Possible values	Default value
‹outflow_begin_time›	Myr	array of reals	$\in [0, 20000]$	0
‹outflow_end_time›	Myr	array of reals	$\in [0, 20000]$	20000
‹outflow_type›	∅	array of strings	‹"none"›, ‹"radical"›, ‹"instantaneous"›, ‹"constant"›, ‹"SF"›, ‹"SN"›, ‹"CCSN"›, ‹"SNIa"›, ‹"ejecta"›, ‹"file"›	‹"none"›
‹outflow_inst_mass›	∅	array of reals	$\geqslant 0$	undefined
‹outflow_const_mass›	∅	array of reals	$\geqslant 0$	undefined
‹outflow_SF_factor›	∅	array of reals	$\geqslant 0$	undefined
‹outflow_SF_power›	∅	array of reals	$\geqslant 0$	0
‹outflow_SN_mass›	$M_\odot$	array of reals	$\geqslant 0$	undefined
‹outflow_SN_power›	∅	array of reals	$\geqslant 0$	0
‹outflow_CCSN_mass›	$M_\odot$	array of reals	$\geqslant 0$	undefined
‹outflow_CCSN_power›	∅	array of reals	$\geqslant 0$	0
‹outflow_SNIa_mass›	$M_\odot$	array of reals	$\geqslant 0$	undefined
‹outflow_SNIa_power›	∅	array of reals	$\geqslant 0$	0
‹outflow_ejec_threshold›	∅	array of reals		undefined
‹outflow_ejec_factor›	∅	array of reals	$\geqslant 0$	1
‹outflow_file›	∅	array of strings		undefined

7. Dust evolution

⟶ Sec. II.5.*b*.*α*, Sec. II.5.*b*.*β*, and Sec. IV.3.*g*, p. 27.

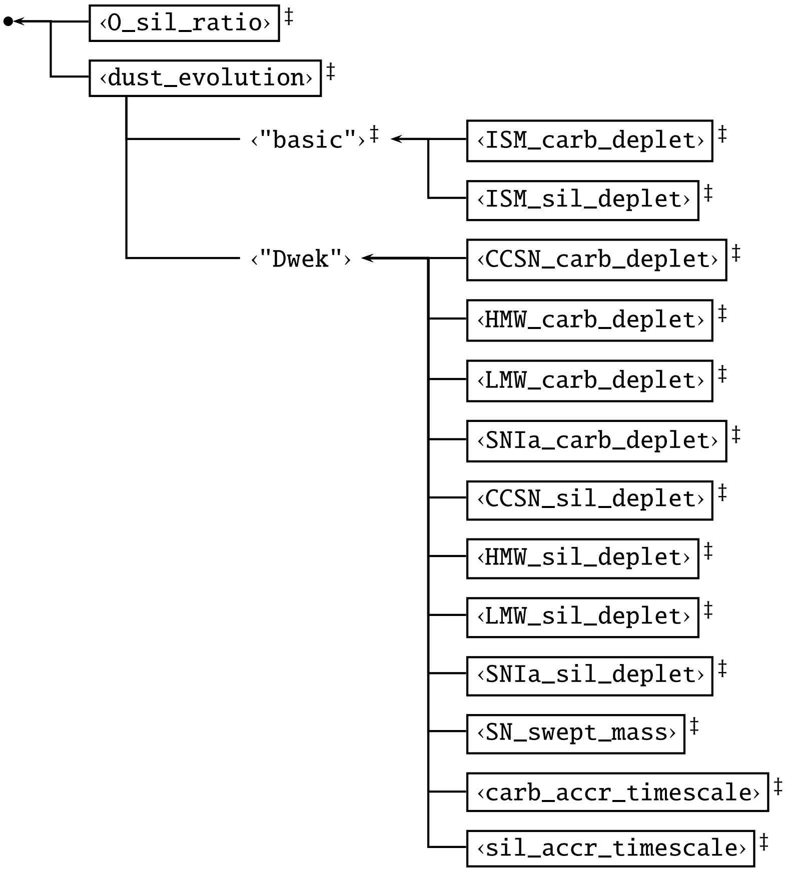


Figure 4. Tree of parameters related to dust evolution.

Table 18. Parameters related to dust evolution.

Parameter	Unit	Type	Possible values	Default value
‹O_sil_ratio›	∅	real	$\geqslant 0$	1[※55]
‹dust_evolution›	∅	boolean	‹"basic"›, ‹"Dwek"›	‹"basic"›
‹ISM_carb_deplet›	∅	real	$\in [0,1]$	1/3
‹ISM_sil_deplet›	∅	real	$\in [0,1]$	1
‹CCSN_carb_deplet›	∅	real	$\in [0,1]$	0.5[※56]
‹CCSN_sil_deplet›	∅	real	$\in [0,1]$	0.8[※56]
‹HMW_carb_deplet›	∅	real	$\in [0,1]$	1[※56]
‹HMW_sil_deplet›	∅	real	$\in [0,1]$	1[※56]
‹LMW_carb_deplet›	∅	real	$\in [0,1]$	1[※56]
‹LMW_sil_deplet›	∅	real	$\in [0,1]$	1[※56]
‹SNIa_carb_deplet›	∅	real	$\in [0,1]$	0.5[※56]
‹SNIa_sil_deplet›	∅	real	$\in [0,1]$	0.8[※56]
‹SN_swept_mass›	$M_\odot$	real	$\geqslant 0$	400[※57]
‹carb_accr_timescale›	Myr	real	$\geqslant 0$	250[※58]
‹sil_accr_timescale›	Myr	real	$\geqslant 0$	250[※58]

[55] Value from Dwek (1998), sec. 5.2. McKinnon et al. (2016) recommend a lower value.

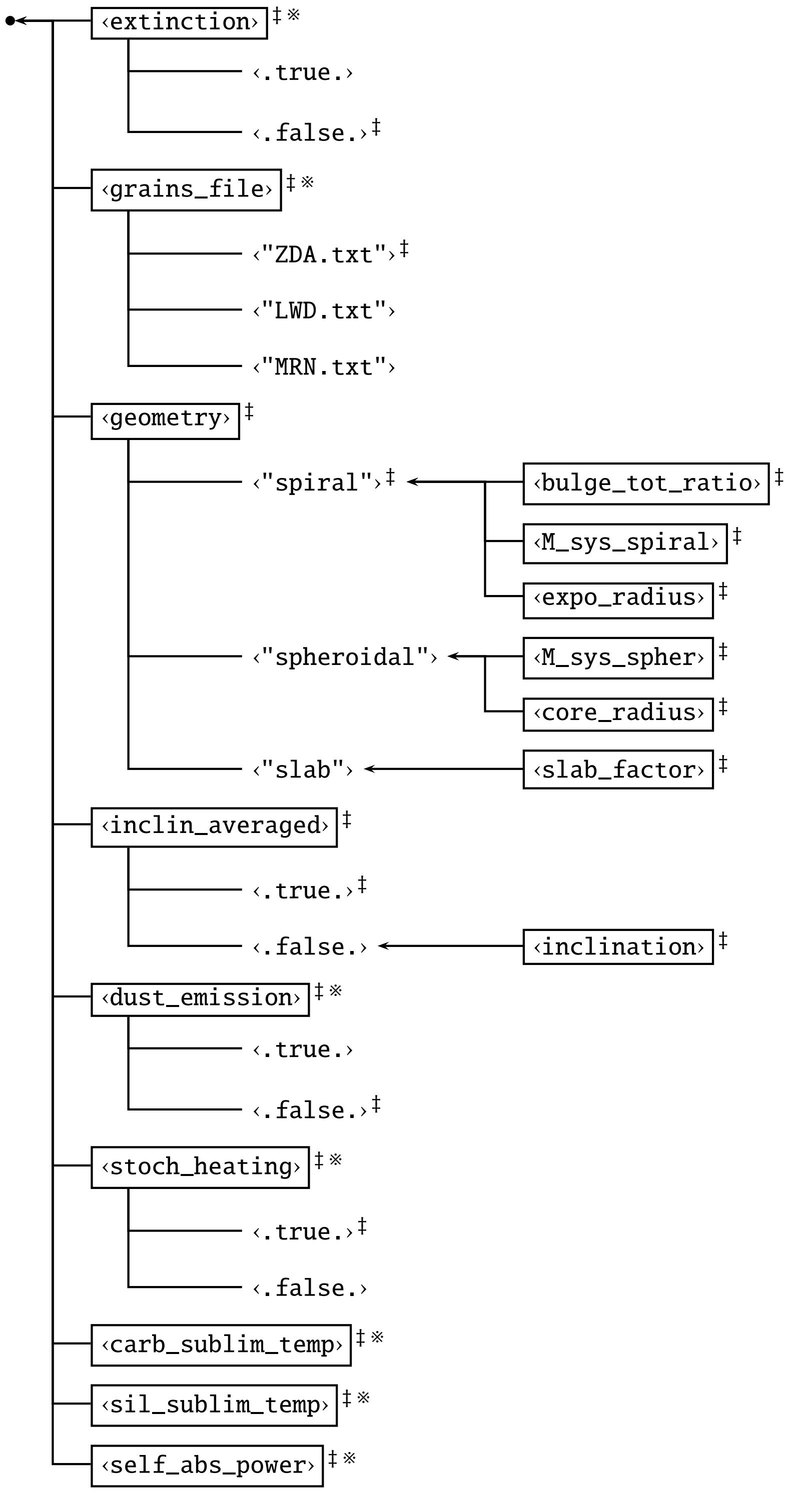
[56] Values from Dwek (1998), sec. 5.2.

[57] $\approx \langle m_\mathrm{d} \rangle_\mathrm{MW} / Z_\mathrm{d,\,MW}$: see Dwek & Cherchneff (2011), after their eq. (9).

[58] See McKinnon et al. (2016), sec. 3.2.

8. Dust attenuation and emission

⟶ Sec. II.6, p. 14, Sec. II.5.*a*, p. 11, and Sec. IV.3.*h*, p. 28.



※ Explicitely assigning to some value, say ‹*val*›, a parameter marked with the symbol ※, say ‹*param*›, automatically sets to ‹*val*› the parameters ‹*param*_SFC› and ‹*param*_DISM›, respectively related either to star-forming clouds or to the diffuse medium. For instance, the explicit assignment "dust_emission = .true." entails both ‹dust_emission_SFC› = ‹.true.› and ‹dust_emission_DISM› = ‹.true.›.

The parameters ‹*_SFC› and ‹*_DISM› may be assigned directly to define the scenario more finely. They are not shown in the tree for the sake of clarity. All of them are listed in Sec. IV.3.*h*, p. 28, and in Table 19 below.

Figure 5. Tree of parameters related to the attenuation by dust grains and to their emission.

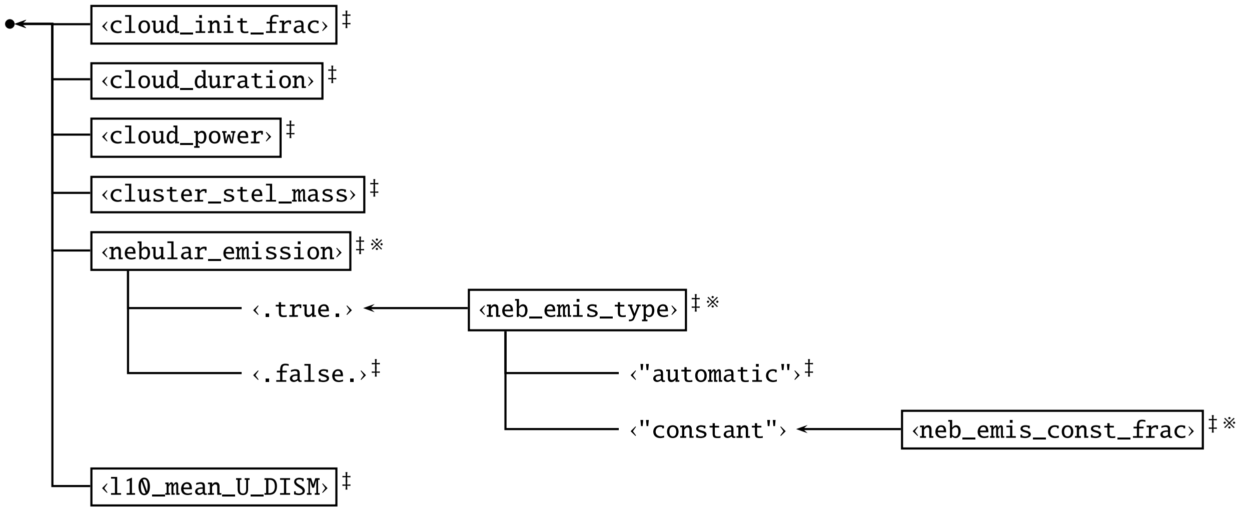
Table 19. Parameters related to the attenuation by dust grains and to their emission.

Parameter	Unit	Type	Possible values	Default value
‹extinction›	∅	boolean	‹.true.›, ‹.false.›	‹.false.›
‹extinction_SFC›	=	=	=	=
‹extinction_DISM›	=	=	=	=
‹grains_file›	∅	string	‹"ZDA.txt"›, ‹"LWD.txt"›, ‹"MRN.txt"›	‹"ZDA.txt"›
‹grains_file_SFC›	=	=	=	=
‹grains_file_DISM›	=	=	=	=
‹geometry›	∅	string	‹"spiral"›, ‹"spheroidal"›, ‹"slab"›	‹"spiral"›
‹inclin_averaged›	∅	boolean	‹.true.›, ‹.false.›	‹.true.›
‹inclination›	degrees	real	$\in [0, 90]$	0
‹bulge_tot_ratio›	∅	real	$\in [0, 1]$	1/7
‹M_sys_spiral›	$M_\odot$	real	$\geqslant 0$	7×10^{10}
‹expo_radius›	parsecs	real	> 0	3.5×10^{3}
‹M_sys_spher›	$M_\odot$	real	$\geqslant 0$	2.79×10^{11}
‹core_radius›	parsecs	real	> 0	192
‹slab_factor›	H atoms/cm^2	real	$\geqslant 0$	6.8×10^{21}
‹dust_emission›	∅	boolean	‹.true.›, ‹.false.›	‹.false.›
‹dust_emission_SFC›	=	=	=	=
‹dust_emission_DISM›	=	=	=	=
‹stoch_heating›	∅	boolean	‹.true.›, ‹.false.›	‹.true.›
‹stoch_heating_SFC›	=	=	=	=
‹stoch_heating_DISM›	=	=	=	=
‹carb_sublim_temp›	kelvins	real	> 0	1750[※59]
‹carb_sublim_temp_SFC›	=	=	=	=
‹carb_sublim_temp_DISM›	=	=	=	=
‹sil_sublim_temp›	kelvins	real	> 0	1400[※59]
‹sil_sublim_temp_SFC›	=	=	=	=
‹sil_sublim_temp_DISM›	=	=	=	=
‹self_abs_power›	∅	real	$\in [0, 1]$	0
‹self_abs_power_SFC›	=	=	=	=
‹self_abs_power_DISM›	=	=	=	=

[59] Laor & Draine (1993), sec. 2.3, 3rd paragraph.

9. Star-forming clouds and nebular emission

⟶ Sec. II.8, p. 16, and Sec. IV.3.*i*, p. 29.



※ See footnote to Fig. 5, p. 61. The parameters ‹*_SFC› and ‹*_DISM› are listed in Sec. IV.3.*i*, p. 29, and in Table 20 below.

Figure 6. Tree of parameters related to star-forming clouds and nebular emission.

Table 20. Parameters for star-forming clouds and nebular emission.

Parameter	Unit	Type	Possible values	Default value
‹cloud_init_frac›	∅	real	$\in [0,1]$	1
‹cloud_duration›	Myr	real	$\geqslant 0$	10
‹cloud_power›	∅	real	$\geqslant 0$	1
‹cluster_stel_mass›	$M_\odot$	real	$\geqslant 0$	10^4
‹nebular_emission›	∅	boolean	‹.true.›, ‹.false.›	‹.false.›
‹nebular_emission_SFC›	=	=	=	=
‹nebular_emission_DISM›	=	=	=	=
‹neb_emis_type›	∅	string	‹"automatic"›, ‹"constant"›	‹"automatic"›
‹neb_emis_type_SFC›	=	=	=	=
‹neb_emis_type_DISM›	=	=	=	=
‹neb_emis_const_frac›	∅	real	$\in [0,1]$	0.7
‹neb_emis_const_frac_SFC›	=	=	=	=
‹neb_emis_const_frac_DISM›	=	=	=	=
‹l10_mean_U_DISM›	∅	real		−3.5[※60]

[60] See Flores-Fajardo et al. (2011) and Dopita et al. (2006).

10. Parameters for output files

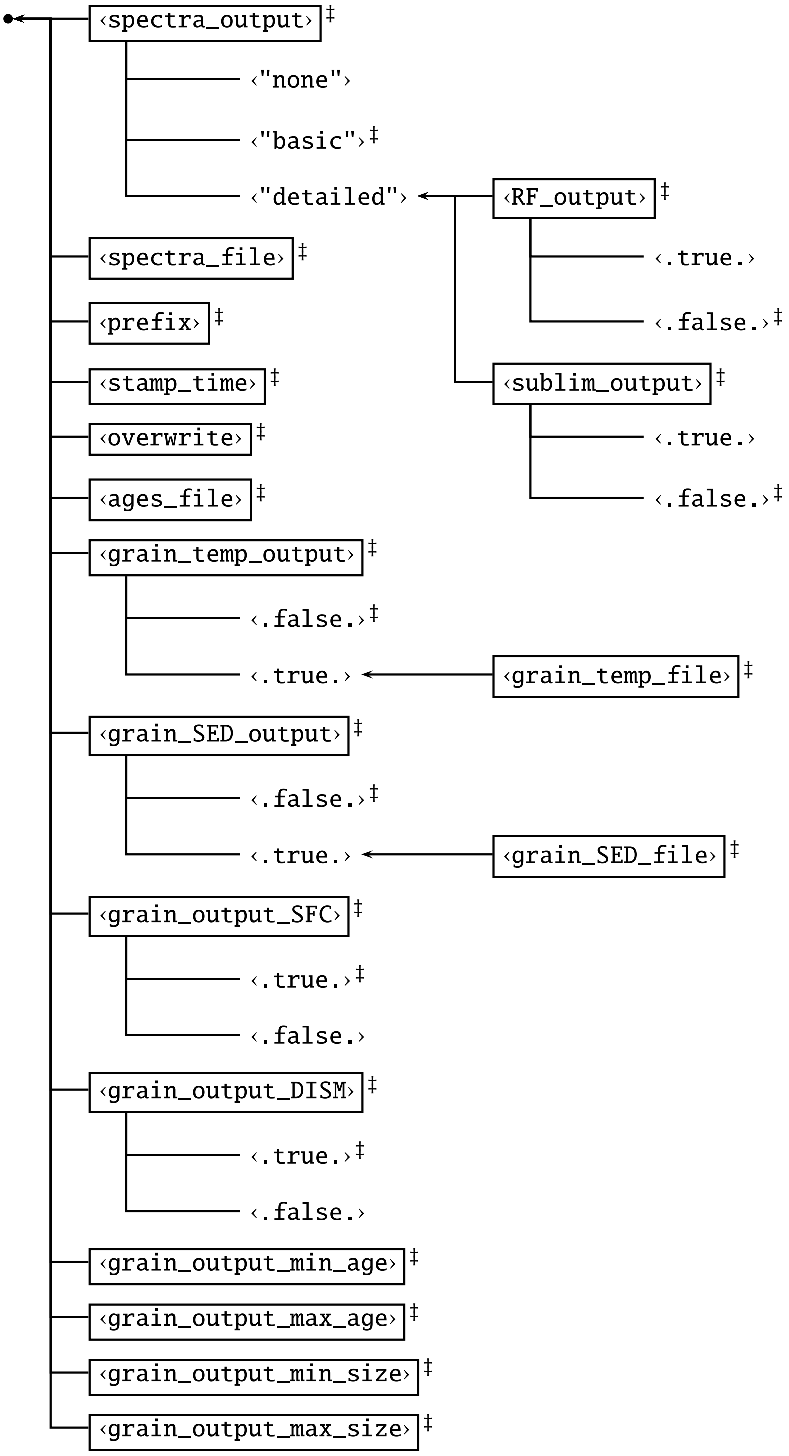


Figure 7. Tree of parameters related to output files.

Table 21. Parameters for output files.

Parameter	Unit	Type	Possible values	Default value
‹spectra_output›	∅	string	‹"none"›, ‹"basic"›, ‹"detailed"›	‹"basic"›
‹RF_output›	∅	boolean	‹.true.›, ‹.false.›	‹.false.›
‹sublim_output›	∅	boolean	‹.true.›, ‹.false.›	‹.false.›
‹spectra_file›	∅	string		See Sec. IV.3.*j*.*α*.ii, p. 30.
‹prefix›	∅	string		‹""›
‹stamp_time›	∅	boolean	‹.true.›, ‹.false.›	‹.false.›
‹overwrite›	∅	boolean	‹.true.›, ‹.false.›	‹.false.›
‹ages_file›	∅	string		‹"spectra_ages.txt"›
‹grain_temp_output›	∅	boolean	‹.true.›, ‹.false.›	‹.false.›
‹grain_temp_file›	∅	string		See Sec. IV.3.*j*.*β*.i, p. 31.
‹grain_SED_output›	∅	boolean	‹.true.›, ‹.false.›	‹.false.›
‹grain_SED_file›	∅	string		See Sec. IV.3.*j*.*β*.i, p. 31.
‹grain_output_SFC›	∅	boolean	‹.true.›, ‹.false.›	‹.true.›
‹grain_output_DISM›	∅	boolean	‹.true.›, ‹.false.›	‹.true.›
‹grain_output_min_age›	Myr	string	⩾ 0	0
‹grain_output_max_age›	Myr	string	⩾ 0	+∞ (!)
‹grain_output_min_size›	µm	string	⩾ 0	0
‹grain_output_max_size›	µm	string	⩾ 0	+∞ (!)

11. Other parameters

Table 22. Other parameters.

Parameter	Unit	Type	Possible values	Default value
‹verbosity›	∅	integer	See Sec. IV.5.*g*, p. 32.	0
‹check_only›	∅	boolean	‹.true.›, ‹.false.›	‹.false.›
‹seed›	∅	array of integers	See Sec. IV.4, p. 31.	

References

Anders, E. & Grevesse, N. 1989, Geochim. Cosmochim. Acta, 53, 197

Balogh, M. L., Morris, S. L., Yee, H. K. C., Carlberg, R. G., & Ellingson, E. 1999, ApJ, 527, 54

Bessell, M. & Murphy, S. 2012, PASP, 124, 140

Binney, J. & Tremaine, S. 2008, Galactic Dynamics: Second Edition (Princeton University Press)

Bloecker, T. 1995, A&A, 299, 755

Bruzual, G. 1983, ApJ, 273, 105

Chabrier, G. 2003, PASP, 115, 763

Chabrier, G. 2005, in ASSL Vol. 327: The Initial Mass Function 50 Years Later, 41

Conroy, C., van Dokkum, P., & Kravtsov, A. 2015, ApJ, 803, 77

Dalla Vecchia, C. & Schaye, J. 2008, MNRAS, 387, 1431

Dopita, M. A., Fischera, J., Sutherland, R. S., et al. 2006, ApJ, 647, 244

Draine, B. T. & Lee, H. M. 1984, ApJ, 285, 89

Dwek, E. 1998, ApJ, 501, 643

Dwek, E. & Cherchneff, I. 2011, ApJ, 727, 63

Fabian, A. C., Nulsen, P. E. J., & Canizares, C. R. 1984, Nature, 310, 733

Ferland, G. J., Chatzikos, M., Guzmán, F., et al. 2017, Rev. Mexicana Astron. Astrofis., 53, 385

Ferrini, F., Franco, J., & Matteucci, F., eds. 1990, Chemical and dynamical evolution of galaxies

Fioc, M. 1997, Ph.D. thesis, Université Paris XI, in French. Available at «ftp://ftp.iap.fr/pub/from_users/¬ fioc/these.pdf»

Fioc, M. & Rocca-Volmerange, B. 1997, A&A, 326, 950

Fioc, M. & Rocca-Volmerange, B. 1999, arXiv:astro-ph/9912179

Fioc, M. & Rocca-Volmerange, B. 2019, A&A, in press. Available at «arXiv:1902.07929»

Flores-Fajardo, N., Morisset, C., Stasińska, G., & Binette, L. 2011, MNRAS, 415, 2182

Groenewegen, M. A. T. & de Jong, T. 1993, A&A, 267, 410

Kennicutt, Jr., R. C. 1983, ApJ, 272, 54

Koester, D. & Schoenberner, D. 1986, A&A, 154, 125

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

Kurucz, R. L. 1979, ApJS, 40, 1

Laor, A. & Draine, B. T. 1993, ApJ, 402, 441

Larson, R. B. 1974, MNRAS, 169, 229

Lejeune, T., Cuisinier, F., & Buser, R. 1998, A&AS, 130, 65

Li, A. & Draine, B. T. 2001, ApJ, 554, 778

Mamajek, E. E., Torres, G., Prsa, A., et al. 2015, arXiv e-prints

Marigo, P. 2001, A&A, 370, 194

Mathis, J. S., Rumpl, W., & Nordsieck, K. H. 1977, ApJ, 217, 425

Matteucci, F. & Greggio, L. 1986, A&A, 154, 279

McKinnon, R., Torrey, P., & Vogelsberger, M. 2016, MNRAS, 457, 3775

Miller, G. E. & Scalo, J. M. 1979, ApJS, 41, 513

Paczyński, B. 1971, Acta Astron., 21, 417

Pearson, T. J. 2002, PGPlot Graphics Subroutine Library, «`www.astro.caltech.edu/~tjp/pgplot/`»

Peebles, P. J. E. 1993, Principles of Physical Cosmology

Planck Collaboration, Ade, P. A. R., Aghanim, N., et al. 2016, A&A, 594, A13

Portinari, L., Chiosi, C., & Bressan, A. 1998, A&A, 334, 505

Rana, N. C. & Basu, S. 1992, A&A, 265, 499

Rauch, T. 2003, A&A, 403, 709

Reimers, D. 1975, Mémoires de la Société Royale des Sciences de Liège, 8, 369

Renzini, A. 1981, in Astrophysics and Space Science Library, Vol. 88, Physical Processes in Red Giants, ed. I. Iben, Jr. & A. Renzini, 431–446

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

Sandage, A. 1986, A&A, 161, 89

Scalo, J. 1998, in Astronomical Society of the Pacific Conference Series, Vol. 142, The Stellar Initial Mass Function (38th Herstmonceux Conference), ed. G. Gilmore & D. Howell, 201

Scalo, J. M. 1986, Fund. Cosmic Phys., 11, 1

Schoenberner, D. 1983, ApJ, 272, 708

Sharma, M. & Nath, B. B. 2012, ApJ, 750, 55

Thielemann, F.-K., Nomoto, K., & Yokoi, K. 1986, A&A, 158, 17

Tsai, J. C. & Mathews, W. G. 1995, ApJ, 448, 84

Weingartner, J. C. & Draine, B. T. 2001, ApJ, 548, 296

Westera, P., Lejeune, T., Buser, R., Cuisinier, F., & Bruzual, G. 2002, A&A, 381, 524

Woosley, S. E. & Weaver, T. A. 1995, ApJS, 101, 181

Zubko, V., Dwek, E., & Arendt, R. G. 2004, ApJS, 152, 211

Index

List of figures

List of tables

Table of contents

Appendices

Figure 8. Charlemagne's advisers show him the Milky Way (cathedral of Chartres, 13th century).

✶ ✶
✶ ✶

Ce travail est dédié à mes filles, les plus belles œuvres auxquelles j'ai contribué.

M. F.