SINOPSIS: a (not so) quick guide

Jacopo Fritz

April 6, 2021

Version 1.6.8

Contents

LICENSE

1 Introduction

This document is meant to be a quick guid to install and use SINOPSIS (SImulatiNg OPtical Spectra wIth Stellar populations models).

This code is intended as a tool to reproduce the main spectral features of observed spectra of galaxies in a (rest–frame) range from the UV to the near-infrared. In principle, the limitation in wavelengths range strongly depends on the adopted theoretical models.

The code is based on the spectrophotometric model described in Fritz et al. (2007) and Fritz et al. (2011). A more detailed description of the algorithm which measures the Equivalent Widths of spectral lines can be found in Fritz et al. (2014).

While SINOPSIS can currently only deal with a wavelength–limited dataset, it will include eventually the far–infrared (FIR, hereafter) and sub-millimetre dust emission as well.

To my knowledge, SINOPSIS is the only spectrophotometric fitting code available in the literature which combines the following features:

- allows an age–dependent treatment of interstellar dust extinction (selective extinction);
- uses simple stellar population models (SSP) which have nebular emission included;
- does not use a precompiled library of models built with specific star formation history patterns, but it searches instead for the best combination(s) of star formation rate and extinction values, as a function of the SSP age, which better reproduced given observed features in an observed spectrum.
- can deal with spectra in the most commonly used formats (ascii, fits, cube,..);
- can successfully fit an observed spectrum in less than 1 second[1].

As typical of such codes, SINOPSIS yields, as an output, several physical parameters, such as the stellar mass, the star formation rate in given age bins, the dust extinction, and the average age of the galaxy, with associated errorbars.

[1] Measured on a Apple iMac, with Intel core i7 @ 3.5 GHz, on spectra with a wavelength extension of about 5000 Å.

2 How does SINOPSIS work?

SINOPSIS measures the mean features of an optical spectrum, namely the average flux in significant portions of continuum–dominated spectral regions, and the equivalent width (EW) of emission and absorption lines, and tries to find a model that minimizes the differences with these observable features in the data.

The model spectrum is constructed by considering a number of Simple Stellar Population (SSP) spectra of different ages and a common, pre–determined, value of the metallicity. To account for effects given by different metallicities, we search for the best fit using different sets of SSP at a different value of the metallicity. Currently, 3 values are implemented: sub–solar ($Z = 0.004$), solar ($Z = 0.02$), and super–solar ($Z = 0.05$), but there is in principle no limitation on the number of metallicity values which can be used.

The age of the oldest stellar population is automatically chosen to be as close as possible to the age of the Universe (T_U) at the galaxy's redshift. To compose the final model, each SSP spectrum is weighted by a given value of the stellar mass and, before being added together, extinction is applied to each one of them, leaving its value free to change as a function of the SSP age, according to the selective extinction hypothesis (e.g. Calzetti et al., 1994). Like this, we allow the youngest stars to have higher extinction values compared to the old ones. In this way, the parameters of the problem are the Star Formation Rate (SFR, which can directly be converted into stellar mass) and the extinction of each SSP of each age.

The model spectrum is hence constructed as follows:

$$F_{MOD}(\lambda) = \sum_{i=1}^{N_{SSP}} SFR(t_i) \cdot \Delta t_i \cdot F_i(\lambda) \cdot 10^{-0.4 R_V E(B-V)_i A_\lambda / A_V} \quad (1)$$

On the right–side of Eq. 1 we have the total number of SSP models, N_{SSP} (each one of different age but common metallicity value), the spectrum of the $i-th$ SSP, $F_i(\lambda)$, the star formation rate value of that SSP, $SFR(t_i)$, the time scale of each SSP, Δt_i, and the color excess (i.e. the extinction) for each SSP, $E(B-V)_i$. The ration A_λ/A_V is the extinction curve, normalized to the V–band.

The number of free parameters mainly depends on the chosen pattern for the star formation history but, if you use the standard settings for the SSP models (see Fritz et al., 2007), it can be as high as 24. Finding the combination which minimizes the differences between the observed and the model spectrum, is hence a non trivial task. This is achieved by using an Adaptive Simulated Annealing (ASA) algorithm, which is a technique particularly suited to deal with non–linear problems in a multi parameters space where there are several local minima.

We have already given an overview of the method in Fritz et al. (2007), and I will not go into those details here. It helps to recall that the best fit search is performed by randomly exploring the parameters space, and that the choice of the parameters' value

at each iteration depends on the previous trial point. E.g., if a different starting point is chosen, a different set of trial points will be explored, leading in general to a different best–fit solution.

As you might have already noticed, the problem is underdetermined, i.e. the number of observable constraints is –in general– less than the number of degree of freedom. Furthermore, the observed constraints (continuum flux and equivalent widths of lines), are not independent. Such problems have an infinite number of solutions, and they are all contained within a polyhedron in the parameters space. We exploit these characteristics of the ASA algorithm to give an estimate of the uncertainties on the physical quantities which are computed (stellar mass, star formation rate, mean ages,...). For each set of SSP at a given value of the metallicity, we run several (the maximum number currently allowed is 11) spectral fitting each of them starting from a different point in the parameter space. Each of this fitting runs will result in a best fit model which is, in general, as good as the others (at least in terms of χ^2 values), but the final parameters will be different. We will use this to give uncertainties. To do so, a "reference", best–fit model, is chosen among those obtained from the SSP sets at a metallicity which gives the lowest χ^2 (which we label as χ^2_B). The model with the median stellar mass is used for this. Hence, uncertainties are calculated as the minimum and maximum values of each parameter from all the other models having a $\chi^2 \leq \tau_m \times \chi^2_B$, where τ_m is a threshold coefficient, which is set equal to 3 as default, but that can be easily changed in the configuration file.

3 Installation

The software, written in Fortran 90, comes in a `.tar` file containing all the data and models you need, together with the source files. Compiling the software should be, in principle, quite platform–independent. If you encounter any problem I would very much appreciate if you could report it to me (j.fritz@irya.unam.mx). What is instead platform–dependent, is the compilation and linking of the `cfitsio` libraries.

3.1 Installing cfitsio libraries

You will need to download and compile the `cfitsio` libraries, which allow SINOPSIS to deal with `fits` files. To do so, follow the following instructions:

- installation on a Mac OS X computer.
  1. go to the NASA's HEASARC page dedicated to the installation of the `cfitsio` libraries on Mac OS X;
  2. following the instructions found on the website should lead to a clean installation[2].
- installation on a linux computer.
  1. download the tarfile containing the libraries from NASA's HEASARC page dedicated to the installation of the `cfitsio` libraries on a UNIX machine (the version at the moment this part of the manual was written is 3390).
  2. you will find all the instructions for configuring and compiling the libraries in the README file in the cfitisio directory;
  3. it seems that the best place to install the libraries is in the the `/usr/` directory, for which you might need root permission;
  4. once you have configured and installed the libraries, open the `Makefile` in the `source/` directory of SINOPSIS. You might need to specify the correct path of these libraries in the linking flags (i.e. the variable `LDFLAGS`);
  5. in the `/usr/` directory, locate the path of the files `drvrsmem.h`, `fitsio.h`, `fitsio2.h`. and `longnam.h`. This is the path you should specify after the "`-C -I`" keywords;
  6. hence, locate the path of the file `libcfitsio.a`. In the same directory you should also find the subdirectory `pkgconfig/`. This is the path you should specify after the "`-L`" keyword.

[2] You might need to run the installation script as superuser.

3.2 Installing SINOPSIS

Create, on a location of your choice, a directory called `SINOPSIS/` and copy the tarfile there. Expanding the tarfile four subdirectories will be created: `data/`, `source/`, `examples/`, and `macros/`. Files in the first two directories should by no means be modified, unless *a)* you exactly know what you are doing or, *b)* it is explicitly required by the set–up and installation process or, finally, *c)* you need a different setup, so for example you need to change the cosmology or to implement a different set of SSPs or theoretical models (in which case you should anyway be aware of what needs to be done).

The other directories, `examples/` and `macros/` contain respectively, some examples – which you can use as templates to create the configuration file and the catalog files you will need to run the code–, and some python macros to help you visualize the results.

VERY IMPORTANT:

To proceed with the installation, you have to update the code's path, so that you can run it from whatever directory you want. In the directory `SINOPSIS/source/`, open the file called `user.f90` and, on the fourth line, which reads as:

```
character(len=64),parameter ::  maindir='/Users/jfritz/'
```

change the variable `maindir`, with the full–path directory where you have uncompressed the code (i.e. the directory where the `SINOPSIS/` directory is located). Do not forget to put a slash (/) at the end of the path, as it is in the original file. For example, if your SINOPSIS path reads as: `/home/username/software/sedfitting/SINOPSIS/`, you will change the above line to:
`character(len=64),parameter ::  maindir='/home/username/software/sedfitting/'`.

3.3 Compiling the code

Before being able to successfully compile the code, you first need to modify the `Makefile`, that you can find in the `source/` directory. This needs to be done in case the path of the installation of the `cfitsio` libraries is different from default. Hence you will have to locate where those libraries were installed, and change the path accordingly. This is done on the third line on the `Makefile` itself. Typical changes of this can be similar to something like: `LDFLAGS=-C -I /home/user/Utilities/cfitsio/include -L /home/user/Utilities/cfitsio/lib -lcfitsio`.

You might have to change the name of the fortran compiler installed on your computer. The default choice is the `gfortran` compiler. If you have another one, just change the first line and substitute `gfortran` with the name of the preferred compiler (e.g. `ifort`, `f90`, etc.).

Note that SINOPSIS was written using `gfortran` version 5, and might not work if your

version is older than this one. Furthermore, to properly compile on OSX, you must have the developer's tools installed, and the command line tools.

You can now compile the code by typing in a terminal window, from within the `source/` directory, the command `make`. If everything works as it should, this will create an executable called `sinopsis`, which you can call, by typing its full path and name, from the directory in your computer containing the files with the observed spectra, the catalog file, and the configuration file.

I will soon make a config file available so that this whole procedure will be not needed anymore and the make setup will be automatic.

3.4 Other set–up options

There is a number of options you might want to check in order to better customise the fitting runs. Most of these option are not modifiable from the `config.sin` file, and are instead "hardcoded" in SINOPSIS. Some require a deep knowledge of the code and/or the models it uses, others are more straightforward and easy to modify.

3.4.1 Cosmology

SINOPSIS offers you the possibility to adjust the cosmological parameters to adapt them to your preferred cosmology. The cosmological parameters, that is H_0, Ω_Λ and Ω_M, are declared in a file called `cosmology.dat`, which you can find in the directory `data/`. Changing these values does not require that you re-compile the code. The default values are 70, 0.70 and 0.30 for the three parameters, respectively.

Furthermore, the name of the file containing these parameters can be specified in the `config.sin` file, so that you can run simultaneously several fitting runs with different cosmologies, by simply creating other files with different parameters and storing them with different names: the file should be placed in the directory where SINOPSIS is runnning. If no file is specified, the `cosmology.dat` will be used as default.

3.4.2 Continuum bands definition

It is possible to define the wavelength limits and the number of the continuum bands which are used as constrains for the model. In the directory `data/` you can find the default file where these are specified: `default_cont_bands.dat`. The three columns contain the lower and the upper wavelength limit, respectively for the first two, and the weight each band has assigned in the χ^2 computation. All the wavelengths are defined in the restframe, and they will be shifted according to the redshift of the observed galaxy.

These bands were carefully chosen so that they sample the continuum emission in a uniform and complete way. But you might want to have your own set. I suggest that, instead of changing the default file, you create your own one, and place it in the directory

in which you are running SINOPSIS. This can be specified in the `config.sin` file. See Sect. 4.2 for more details on this.

3.4.3 The main stellar age bins

In Fritz et al. (2007) we have performed several test to asses the reliability of the star formation history, that is the star formation rate as a function of cosmic age. This was performed on the WINGS dataset and, for the typical S/N of those spectra, it turned out that 4 age bins (young, young-intermediate, intermediate-old, old) is the best we could do with this method.

Nevertheless, you can change both the number of bins and the ages to fit your own needs. This can be done...

3.4.4 Extending the photometric bands set

SINOPSIS can use, together with the spectrum, broad band photometry data as a further constraint. A pre–defined set of photometric bands is already included within the `data/filters/` directory. You can extend or change it by simply copying the transmission curves of the filters you're interested in, in this directory. Furthermore, update the `filter_list.dat` file, where all the transmission filters are listed.

To be properly used, the transmission curve filters should have the same format as those already given as default: a 2–columns file, with the first being the wavelength array, in Å, and the second being the transmissivity.

4 Running the code: setup and input files

Now, you need to have your data –the catalog containing the needed details of your objects, the spectra and, in case they are available, photometric datapoints as well– in the right format.

In this section we will go into the details of the format for the input files, and we will see what are the possible formats that SINOPSIS accepts for the data.

Before you run the code you should have ready, in a common directory:

1. your observed spectra, which can be given in different formats:

   - a single `ascii` file for each spectrum containing two columns: the wavelength grid in the first one, and the flux on the second. A header, with an arbitrary number of lines, is allowed, provided each line begins with a "#" character. Alternatively, it is possible to tell the code to skip, **in addition to those lines beginning with a #**, an arbitrary number of lines (see the related keyword on the configuration file, Sect. 4.2).
   - a single fits file for each spectrum (NOTE: SINOPSIS rely on the header to extracts mandatory information to properly interpret the `.fits` file. Hence the header should contain standard information);
   - a 2–dimensional `fits` file, containing ny spectra having nx wavelength datapoints (where nx and ny are the number of pixels in the x direction and y coordinate, respectively);
   - a 3–dimensional (a cube) `fits` file, containing $nx \times ny$ spectra, where the z coordinate is used for the wavelength array (i.e. the number of wavelength is nz);

   Note: the wavelength array must be in Å, and the flux in F_λ units (but the absolute normalization, as it will be discussed later, can be arbitrary); both the wavelengths and the flux array can be optionally given in logarithmic units.

2. a "catalog" file containing the details of the spectra. Depending on the format of your spectra, the characteristics of this file might change. An example for each one of the various formats can be found in the directory `examples/`, and they will be discussed in detail in Sect. 4.1;

3. a configuration file, named `config.sin`, of which you can find a template in the `example/` directory, and that is described in detail in Sect. 4.2. This file is used to define the parameters and some of the characteristics of the fits.

4. if your data are given in a cube, you will also need a redshift mask, that is a fits file with the same `NAXIS1` and `NAXIS2` keywords as the cube, for which each pixel

contains the redshift value for the corresponding spectrum. In this case SINOPSIS accepts 2 redshifts mask: one from derived from the stellar (absorption) component, and another one derived from the gas (emission). The latter will be used to measure emission lines equivalent widths only.

4.1 The input catalog format

A "catalog" file is used to provide the code with the spectra filenames, the redshifts, and other quantities, such as observed magnitudes, that SINOPSIS needs to run properly. The format of this input catalog may vary, mainly depending on the format of the spectra.

There are some different possible combinations for this file, even though I have tried to limit this number. These are hereafter defined as "CASEs". While the general content does not differ much from case to case, some details need to be adjusted. In the `examples/` directory you can find examples of input catalogs for all the different cases hereafter outlined.

4.1.1 Observed spectra in separate files (CASE 1 to 3)

One of the possibilities is to feed the observed spectra to SINOPSIS as single `ascii` or `fits` files. In this case, the input catalog file will contain the name of the files of the observed spectra in the first column, and their corresponding redshifts in the second. Table 1 shows an example of the input catalog for this case, in which it is assumed that the spectra have a reliable absolute flux calibration. Here, the first line contains the number of photometric

0	
# specname	Redshift
spec1.ascii	0.04400611
spec2.ascii	0.04385601
spec3.ascii	0.14571311
spec4.ascii	0.01481592
spec5.ascii	0.04167116
spec6.ascii	0.10818050

Table 1: Example of the input "catalog file", containing the properties of the spectra, given as `ascii` files in this example (the `fits` file extension will be used if the spectra are in fits format). CASE 1.

bands to be used as a further constraint, if available (hence, none in this case). If this number is not 0, further information needs to be provided, in the format described in Table 2. The second line, for both Table 1 and Table 2, is a mandatory comment line (you might leave the columns names unspecified, but the code will skip this second line anyway).

2 Jon J K					
# specname	Redshift	Jmag	J_err	Kmag	K_err
spec1.ascii	0.04400611	17.4581	10	15.4072	15
spec2.ascii	0.04385601	17.6022	10	15.9483	15
spec3.ascii	0.14571311	19.3187	10	17.6595	15
spec4.ascii	0.01481592	20.5865	10	19.1483	15
spec5.ascii	0.04167116	17.5115	10	14.8413	15
spec6.ascii	0.10818050	19.7917	10	18.9322	15

Table 2: Example of the " input catalog file", containing the general properties of the spectra, for CASE 2.

In Table 2, I show the same example as before, but in this case 2 photometric data points are used as constraints (the J and K band magnitudes). Here, the first element on the first line contains, as in CASE 1, the number of photometric bands to be used as constraints for the model. In this case, we will provide: the magnitude type (currently only Johnson–Vega and AB are supported), and the name of the bands, two in this case. After the comment line we have, column by column: the spectrum file name, its redshift, the value of the magnitudes and the errorbar to be associated, given in flux percentage, band by band in the same order as specified in the first line.

A further option is allowed (CASE 3): that is, the absolute flux calibration of the spectra is based on an observed magnitude defined in the same range as the observed spectrum. This option is especially intended to be used when the absolute flux calibration is poor, or in case fiber spectra are analyzed. In the latter case, it is possible to normalize (calibrate) the observed spectra to the value of a magnitude measured on an aperture matching that of the fiber, and defined in the same position on the observed galaxy.

This option will allow the possibility of normalizing the physical quantities that SINOPSIS derives to the total magnitude as well obtaining, in this way, both "aperture" and "total" –related physical properties. For the use of this option see its application to the WINGS sample (Fritz et al., 2011).

The input catalog for CASE 3 will look like that described in Table 3. The structure of the input catalog is similar to that of the previous casea. In detail: if no photometric datapoints are used, we still need to specify the magnitude type (Johnson or AB) as the second flag on the first line. After the comment line, there are, again, the spectra filenames in column 1, the redshifts in column 2 followed by the values of the magnitude in an aperture matching that of the spectral fiber, the value of the total aperture magnitude, and the name of the band in which these magnitudes are measured (note this band can differ for each spectrum).

A combination of CASE 2 and CASE 3, that is the use of magnitude to absolute flux–calibrate the spectra, combined with photometric datapoints to be used as a further constraint, can be used as well. To allow this, the first line will take the same form as in

0 Jon				
# specname	Redshift	Aperture	Total	Band
spec1.ascii	0.04400611	17.4581	15.4072	V
spec2.ascii	0.04385601	17.6022	15.9483	V
spec3.ascii	0.14571311	19.3187	17.6595	V
spec4.ascii	0.01481592	20.5865	19.1483	V
spec5.ascii	0.04167116	17.5115	14.8413	V
spec6.ascii	0.10818050	19.7917	18.9322	V

Table 3: Example of the " input catalog file", containing the general properties of the spectra, for CASE 3.

CASE 2, while the columns will be similar to that of CASE 3. The values of the broad–band data points will have to be located after the last column of a CASE 3–like catalog, following the same format as for CASE 2.

The type of normalization (i.e. either based on the flux measured on the observed spectra, or on the flux derived by photometry in a given band) can be specified in the configuration file (see Sect. 4.2 for further details).

We refer the reader to Sect. 7 for further details on how the broad–band photometry is used as a constraint.

Note that, when the '`eqw`' keyword is chosen, and the code will hence only measure the equivalent widths of the lines, without performing any spectral fitting, only the first 2 columns are needed (spectra filename and redshift), and the other ones can be skipped (see the description of the `config.dat` setup file, below). Furthermore, the information contained in the first line can be skipped as not used in the measure. Hence, the input catalog file will in this case contain an arbitrary number of header lines, each identified by a "#" symbol, followed by the aforementioned information. An example of such an input catalog is given in Table 4.

#0 Jon	
# specname	Redshift
spec1.ascii	0.04400611
spec2.ascii	0.04385601
spec3.ascii	0.14571311
spec4.ascii	0.01481592
spec5.ascii	0.04167116
spec6.ascii	0.10818050

Table 4: Example of the input "catalog file" in case only the equivalent width values are measured and no spectral fit is performed.

4.1.2 Observed spectra in a 2-D fits file (CASE 4)

Observed spectra can be also read from a 2-D fits file, containing one spectrum for each pixel line. In this case, if nx and ny are the number of pixels of the 2-D array in the x and y direction respectively, nx represents the number of wavelengths datapoints, and ny the number of spectra. Note that not all the lines in the `fits` image might contain a usable spectra, but this information will be given in column 1 of the input file. In this way, only the lines containing real spectra will be analyzed.

0	
2D_specfile.fits	
# specnumber	Redshift
1	0.04400611
2	0.04385601
5	0.14571311
6	0.01481592
7	0.04167116
12	0.10818050

Table 5: Example of the input "catalog file", containing the general properties of the spectra, for CASE 4 (analogous to CASE 1).

In the case of the input catalog depicted in Table 5, the situation is similar to CASE 1: we assume that the absolute calibration of the spectra is reliable, so all of the physical quantities are calculated with the spectral flux normalization. Furthermore, no extra–information from photometric data is used for this example. Adapting this catalog to the analogous of CASE 2 and CASE 3 is straightforward.

4.1.3 Observed spectra in a 3-D cube (fits) file (CASE 5)

The increasing number and use of Integral Field Units (IFU) has led to the development of an IFU option on SINOPSIS as well. The supported formats are, at the moment, only those provided by observations with the MUSE instrument on the VLT telescope, but I have tried to keep this as general as possible.

The input catalog file, for this case, is extremely short and simple. You are requested to specify the name of the datacube file containing all the spectra, the name of a 2–dimensional mask, in fits format, containing the redshift of each pixel of the cube (hence, this file will have the same number of pixels in the x and y coordinate as for the datacube), and the redshift of the galaxy (which will be used to calculated its distance and, hence, the proper normalization for the stellar masses and star formation rates). Optionally, in the same row as the redshift, you can put a value for the distance, in case the Hubble law cannot be used to calculate a reliable distance value.

datacube_file.fits	2
zmask_absorption_file.fits	zmask_emission_file.fits
0.004283	16.7
band1.dat	band2.dat
phot_image1.fits	phot_image2.fits
error_image1.fits	error_iamge2.fits

Table 6: Example of the " input catalog file", for CASE 5, i.e. for when an IFU datacube is used. Note that the the `zmask_emission_file.fits` is optional, and the last 3 lines as well.

Table 6 shows the structure of the input catalog file for this case. Note that the only supported format for the redshift is currently the aforementioned 2D fits image. Pixels without a redshift should be assigned negative value, and will be skipped by the code.

Since version 1.6.3, it is possible to specify 2 redshifts masks, one derived from absorption lines (stellar redshift) and one derived from emission lines (gas redshift). Like this, it is possible to take into account possible differences in the wavelength of the center of absorption and emission lines (typically the Balmer lines). Hence, the gas redshift, will be used to measure emission lines.

The code assumes that, in the input catalog file, the first name is the one for absorption lines, and the second is for emission lines redshift.

An IFU cube can contain several thousands of spectra. It is sometimes useful to only run the code on a subset of data. SINOPSIS allows this by only considering pixels contained within a rectangular region, defined by the user, by means of the `x` and `y` coordinates of the lower–left and upper right corners of the rectangle. To use this option, instead of simply calling SINOPSIS at the terminal's prompt, add the coordinates of the corners defining the region, i.e.:

```
prompt> sinopsis 100 90 120 110
```

will run the code on set of pixels with coordinates in the range $100 \leq x \leq 120$ and $90 \leq y \leq 110$.

Similarly, a fits format mask can be fed to sinopsis to decide which spaxels will be fitted and which not. The mask needs to have the same `NAXIS1` and `NAXIS2` values as the IFU cube. Spaxels with values larger than 0 will be fitted, others will be skipped. Similarly to the case described above, to use this option simply provide the name of the file containing the mask after the call to SINOPSIS:

```
prompt> sinopsis fittingmask.fits
```

This will do.

Since version 1.6.7 it is also possible to use, combined to the spectra, photometric datapoints outside the observed spectral range. To do so, you need to specify how many bands you will use just after the spectral cube filename (see 6) and, after the redshift

4.1.4 The "advanced" catalog option

SINOPSIS is mainly designed to work with large spectral databases. Nevertheless, you might need to use it to fit only few spectra and, in this case, you might also want to make sure that all possible observed details are well reproduced. To this end, you can use the `advanced` catalog option.

When this option is chosen, you can tweak the fit for each of the spectra separately. A template of a catalog file can be found in the `example/` directory. A list of the tuneable options is given hereafter.

- equivalent width values. You can choose, for each spectrum, which spectral line is measured, how, the measured uncertainty, and its weight on the χ^2. You can also measure the equivalent width value yourself, and use this as a constraint;
- continuum bands: it is possible to define, for each spectrum, the set of continuum bands, and their weight on the χ^2.

4.1.5 Using EW from lines that are outside the observed spectral range

This option, which was introduced starting from version 1.6.7, allows SINOPSIS to use as a model constraint the EW value from a spectral line that is not included in the observed range. This might happen if there are multiple observations of the same objects with spectrographs sampling different spectral range (of course it is left to the user to check that the data are matched in aperture and position).

In order to use this option, set the relative keyword to `yes` (see 4.2), and add the EW value on the catalog file in columns as the example shown below displays.

```
0  3
# specname   redshift  lines         ew                  err
spec.dat     0.0193    Mg  NaD  Ha   2.00  99.  -59.29   0.5  99.  6.0
```

The example shown above uses a "CASE 2" catalog, i.e. a list of spectra that do not need any normalization and for which no further photometric constraint is given/used. In this case the number "3" represents the number of lines that are measured (from other data) but not used because out of the spectral range: the magnesium line (Mg at ~ 5177 Å), the sodium lines (NaD at ~ 5894 Å) and Hα. The labels are firstly specified, and they need to be the same as used through out the code (see note below), hence the EW values and the estimated errors, following the same order of the labels. They need to be measured at the same redshift at which the fit is performed. If a line is not detected then a 99. flag should be used.

In the example shown above, the `NaD` and `Mg` lines are used, while they are not taken into account as constraints in the modeling. If you wish that they are instead fitted by the model, just change the corresponding keywords in the `spec_lines_param.dat` or

`fix_lines_param.dat` (depending on which kind of line measurement you are using) files (see Sect. 6.1 and 6.2 for details).

If you are using a different kind of input catalog (e.g. where photometry is used as well), just put these column after all the others.

IMPORTANT: only lines that are in the pre-defined lists used by SINOPSIS (i.e. that are listed in the files `spec_lines_param.dat` and `fix_lines_param.dat`) are allowed.

4.2 The configuration file

The third file you need to prepare is a configuration file, which is recognized by SINOPSIS by its mandatory name, `config.sin`. Here you have to set up some parameters needed by SINOPSIS. Its format is quite free, meaning that there is no specified order for the parameters, and any comment can be added, provided it begins with a "#". There is only one mandatory parameter, and this is the main catalog file name. All the other parameters can be omitted, as their values will be taken by a default set. An example of the configuration file can be found in the directory `example/`.

In order to detect the presence, and the value, of a given keyword, a specific string identifier needs to be read. This string starts at column 1 in the `config.sin` file, and must end with the symbols "`:=`", which will precede the value of that given keyword. For this reason, the string identifier must be exactly as described below (and as found in the `config.sin` file, given as an example).

For the sake of clarity, the configuration file is subdivided in various sections which, as stated above, need not to be in the same order as in the example file. When needed, a comment line, listing the allowed values for a given keyword, precedes the keyword string identifier, so that the user knows what values are allowed.

Below I show an example of how such a file should look like, and give some basic information on each keyword. We refer to the proper section, when needed, for more detailed explanations on the meaning of each keyword.

```
##############################################################################################
###                                                                                        ###
###                          Configuration file for SINOPSIS                               ###
###                                  version 1.6.8                                         ###
###                                                                                        ###
### If a keyword is not defined here, a default value will be assumed, when possible.      ###
###     When a keywords has a set of possible values, these are indicated between ''       ###
###                                                                                        ###
##############################################################################################
##############################################################################################
#######               #######
###     INPUT CATALOG
Name of the input catalog := inputcatalog.dat
# Allowed keywords: 'basic', 'advanced', 'eqw'.
Type of input catalog := basic
```

```
################################################################################################
#######                                                  #######
###     OBSERVED SPECTRA CHARACTERISTICS and OPTIONS
# Allowed keywords: 'ascii', 'fits', 'mfits', 'cube'
Format of the observed spectrum or spectra:= ascii
Spectral resolution of the data (FWHM in Angstroem) := 9.0
# Allowed keywords: 'linear', 'logarithmic'
Wavelength array in linear or logarithmic units := linear
# Allowed keywords:  'linear', 'logarithmic'
Flux array in linear or logarithmic units := linear
Number of lines to skip in the observed spectra (ascii format only) := 0
Cut the observed spectra in the blue part by this amount (in Angstroem) := 200
Cut the observed spectra in the red part by this amount (in Angstroem) := 300
# Allowed keywords: 'default', custom filename
File with the parameters for the variable equivalent widths measurements := default
Use fixed bandwidths to measure equivalent widths in observed spectra := no
# Allowed keywords: 'default', custom filename
File with the parameters for the fixed equivalent widths measurements := default
Account for a contribution of [NII] to Halpha flux := 0.0
Measure the equivalent width of Hbeta using fixed bandwidth := no
Use constraints from spectral lines outside the observed spectral range := no
Smooth the observed spectra resolution to match the SSP resolution := no
Smooth the observed spectra to a custom resolution := no
Resolution of the smoothed observed spectra (FWHM in Angstroem) := 6.0
Write the smoothed observed spectra := no
################################################################################################
#######                                                                                #######
###     NORMALIZATION OF THE MODEL SPECTRUM & OBSERVED CONSTRAINTS (CONTINUUM)
# Allowed keywords: 'phot', 'spec', 'none', 'norm'
Normalize the model spectrum to := spec
Normalisation factor := 1.0e-17
# Allowed keywords: 'default', custom filename
File with the definition of the continuum bands used as constraints := default
# Allowed keywords: 'default', custom value
Normalization of observed and model spectra := default
################################################################################################
#######            #######
###     EXTINCTION
##Allowed keywords: 'MW', 'SMC', 'CAL', '2.5', '4.0', '5.0'
Extinction curve to be adopted := MW
################################################################################################
#######                                  #######
###    CHARACTERISTICS OF THE MODELLING
# Allowed keywords: 'jm', 'cb20'
SSP set := jm
# Allowed keywords: 'ff', 'dexp', 'logn'
Star formation history pattern := ff
# Allowed keywords: 'default', custom filename
File with the star formation history parameters := default
Number of different metallicity values := 3
```

```
Metallicity values to be used := 0.004 0.02 0.05
Smooth SSP spectra to the observed spectra resolution := no
# Allowed keywords: 'zage', maxage, zform
# When maximum age is given, it must be in units of yr (i.e.: 5.5e9)
Set maximum age of the stellar populations := zage
##############################################################################################
#######                                  #######
###     UNCERTAINTIES DETERMINATION
Number of separate runs for each metallicity value := 11
Chi2 threshold value to calculate uncertainties := 2.0
##############################################################################################
#######         #######
###     VARIOUS
Create a model magnitudes catalog := yes
# Allowed keywords: 'Jon', 'AB'
Magnitudes type for the model catalog := Jon
Catalog of redshift-independent distances to be used := mydistances.dat
# Allowed keywords: 'default', custom filename
File with cosmological parameters := default
Redshift value below which a redshift-independent distance is used := 0.001
Output all the best fits for all runs and metallicities := no
Write output file for each reference model := yes
Write out the model spectra without emission lines := yes
Write out the flag mask of fitted pixels in a cube dataset := yes
Write out the contribution to the continuum flux from the SSPs := yes
Write out the model spectra for each used age := no
Minimize memory usage := n
##############################################################################################
#######                                   #######
###     DUSTY SSP AND IR CONSTRAINTS
Include dusty SSPs := n
##############################################################################################
```

In the following we list the meaning of each single keyword. The default assumed value, whenever possible, is indicated with a **(d)**.

1. `Name of the input catalog`. This is the only mandatory keyword. It is the name of the input catalog file, where the information about the spectra is given;

2. `Type of input catalog`. Only three values are allowed here:

   - `basic` **(d)**
   - `advanced`
   - `eqw`

   They specify what kind of main input file you want to use. In case `eqw` is used, no spectral fitting will be performed, but only the measure of the equivalent widths, with the trend curve analysis method. When this option is used, only the filenames of the

spectra, and their redshifts, need to be specified in the input "catalog file". Using the keyword `basic`, you will perform a fully automatic measurement of all the observed quantities. The catalog format can be any of those described in Sect. 4.1, depending on the format of your data. The `advanced` option allows you to use a much more detailed catalog file which contains more information for each one of the spectra you are fitting. This format, in particular, allows you to provide the observed values of the equivalent width of the lines of interest, and to use a custom definition of the bands used to get a measure of the continuum emission, for each single spectrum. This might be especially useful if you are dealing with few spectra, and if you want a better tweak for the fitting. Its format will be discussed later on, and it is coming back fully supported very soon;

3. `Format of the observed spectra`. This keyword specifies the format of the observed spectra. At the moment 4 formats are supported, and they are identified by the following keywords:

   (a) `ascii` **(d)**: all the spectra are given in separate text files, whose format has been described in Sect. 4;

   (b) `fits`: all the spectra are given in separate fits files;

   (c) `mfits`: the spectra are given on a single 2-D fits file. The different spectra are given at different y–coordinates, while the x–coordinate gives the wavelength array;

   (d) `cube`: the spectra are given as a IFU cube: the x and y dimensions represent the spatial coordinate of the spectra, and the z axis contains the wavelength dependence;

4. `Spectral resolution of the data (FWHM in Angstroem)`. Must be given as floating point value, and it is only needed if you want to smooth either the observed spectra, or the SSPs models (see also below);

5. `Wavelength array in linear or logarithmic units`. The two possible values, `linear` **(d)** and `logarithmic`, allow to properly deal with spectra having the wavelength array in logarithmic units;

6. `Flux array in linear or logarithmic units`. The two possible values, `linear` **(d)** and `logarithmic`, allow to properly deal with spectra having the flux array in logarithmic units;

7. `Number of lines to skip in the observed spectra (ascii format only)`. Sometimes spectra in ascii format are converted from `iraf` using, e.g., the `wspectext` packages which, among the other, gives the option of leaving the header of the corresponding fits file. Setting this value to the number of the lines occupied by the

header (provided they are NOT preceded by means a # character), will make the code skip these lines;

8. `Cut the observed spectra in the blue part by this amount (in Angstroem)`. Setting this keyword to a floating higher than 0, will cause SINOPSIS to neglect the blue part of the observed spectra by that given amount, in Å. Like this, it is possible to avoid low S/N or bad calibrated parts of the spectrum;

9. `Cut the observed spectra in the red part by this amount (in Angstroem)`. Setting this keyword to a floating higher than 0, will cause SINOPSIS to neglect the red part of the observed spectra by that given amount. Like this, it is possible to avoid low S/N or bad calibrated parts of the spectrum;

10. `Use fixed bandwidths to measure equivalent widths in observed spectra`. This option allows to chose between two different methods of measuring the equivalent widths of spectral lines: the variable FWZI method, chosen when answering `no` **(d)**, and the fixed FWZI, chosen when answering `yes`;

11. `File with the parameters for the equivalent widths measurements`. This is only used when choosing the fixed FWZI method to measure lines' EW (see point above). If left to `default` **(d)**, then the `fix_lines_param.dat`, located in the `data/` directory is used. Otherwise, the user can define their own parameters to be written in a custom file, in the same format as then default one, to be placed in the directory where SINOPSIS is run.

12. `Account for a contribution of [NII] to Halpha flux`. This keyword allows to add a contribution to the flux (EW) of the Hα line due to the adjacent [NII] lines (might be useful if the line in the observed spectra are blended and they are not in the model spectrum). This number is multiplied to the measured flux (EW) of Hα if in emission, to account for this spurious contribution. A value of 0 **(d)** of this parameter means no contribution, a value of, say, 0.5, would add a 50% contribution;

13. `Measure the equivalent width of Hbeta using fixed bandwidth`. If set to `yes`, uses fixed FWZI to measure the Hβ line only. Particularly useful when no lines blueward of Hβ are not present. The default value is `no` **(d)**;

14. `Use constraints from spectral lines outside the observed spectral range`. If set to `yes`, allows the use of EW values of lines outside the observed spectral range as constraints. The EWs values will be specified in the catalog file, and must be measured at the same redshift at which the fit is performed (other wise a correction for redshift has to be performed);

15. `Smooth the observed spectra resolution to match the SSP resolution`. If the spectral resolution of the data is higher than that of the SSP models, it is possible

to reduce the resolution by smoothing the observed spectra. To do so, the user has to provide the value of the observed spectral resolution;

16. `Smooth the observed spectra to a custom resolution`. Answer `yes` if you wish to lower the resolution of the observed spectra to a user–defined value (in Å);

17. `Resolution of the smoothed observed spectra (FWHM in Angström)`. If the value of the previous keyword is `yes`, provide a value of the full width half maximum (FWHM) of the gaussian that will be used to smooth the observed spectra;

18. `Write the smoothed observed spectra`. Setting this keyword to `yes` will produce, for each spectrum, an ascii file (with the extension `.smooth`) containing the smoothed observed spectrum;

19. `Normalize the model spectrum to`. There are four possible options:

    - `phot`: when this option is used, the input catalog must contain two magnitude values which will be used to normalize the spectrum to a physical flux scale. This option, created to deal with fiber spectroscopy, needs a magnitude value taken in the same position and with the same aperture of the spectroscopic fiber, and a total magnitude value. In case only one of the two is available, then the same value should be repeated;
    - `spec` **(d)**: in this case the spectrum is assumed to have a proper absolute calibration, and all physical quantities (stellar masses, star formation rates, etc.) will be computed accordingly;
    - `none`: this option allows the user to deal with, for example, stacked spectra for which the absolute calibration is not important. In this case absolute quantities (e.g. the total stellar mass) will have no meaning;
    - `norm`: this option can be used when the observed spectra come in normalized units.

20. `Normalization factor`. This is the normalization factor by which the observed spectra are multiplied inside the code, in case the `norm` option is used (see point above). This number must be in units of erg/s/cm^{-2}/Å;

21. `File with the definition of the continuum bands used as constraints`. The possible values are `custom file` (where here you should use a filename with your own bands definition) and `default` **(d)**. SINOPSIS uses, by default, this pre–defined spectral bands to calculate the continuum flux to constrain the model. These bands are chosen in such a way that they do not contain (strong) spectral features such as emission or absorption lines, and are defined in the file `default_cont_bands.dat`, located in the `data/` directory. You can use the aforementioned file as a template to

build your own set of continuum bands. In this case, place this file in the directory in which you are running SINOPSIS, and specify the name of the file. When using the `advanced` option of the input catalog, it is given the user the possibility to define a custom set of such bands for each spectrum (the exact band is chosen from a list internally defined).

22. `Normalization of observed and model spectra`. If set to `default` **(d)** SINOPSIS normalizes the observed (and model) spectrum to the reddest bands possible. If a number is given (`custom value`), then this will be taken as the rest–frame central wavelength, in angström, of a band (of 200 Å in width) where the normalization flux is calculated;

23. `Extinction curve to be adopted`. There are several possibilities for the extinction curve:

    - `MW`, corresponds the the Milky Way extinction curve (Cardelli et al., 1989) **(d)**;
    - `SMC`, is the one of the Small Magellanic Cloud;
    - `CAL`, is the attenuation law by Calzetti et al. (1994);
    - `2.5`, corresponds to an extinction curve with $R_V = 2.5$;
    - `4.0`, corresponds to an extinction curve with $R_V = 4.0$;
    - `5.0`, corresponds to an extinction curve with $R_V = 5.0$.

24. `SSP set`. Defines the set of SSP spectra to be used. The public version can currently deal only with one set, the Jacoby-Miles as described in Fritz et al. (2007). The new Charlot & Bruzual models should be available as soon as these models becomes publicly available. A set with the Bruzual & Charlot (2003) models will be soon available as well;

25. `Star formation history pattern`. Three possibilities for this option:

    - `ff`**(d)**: this is what we call the *free-free* approach, in which SSP of at most 12 different ages are used. Their star formation rates and extinction values are –if not otherwise specified– let free to vary independently with respect to each other;
    - `dexp`: a double, decaying exponential is used to mimic the star formation rate as a function of the SSP age (i.e. the star formation history. See Sect. 5.1 for more details);
    - `logn`: a log–normal law is s used to mimic the star formation history (see Sect. 5.1 for more details);

26. `File with the star formation history parameters`. The possible values are `custom file` and `default` **(d)**. This file contains the setup of the stellar populations and extinction parameters, including the initial values for the construction of the first model and the maximum and minimum values of the parameters that are being minimized. To customize it, it is strongly suggested that you copy the default file (it is one of the `ssp_setup_???.dat` files that you can find in the `data/` directory) in the directory where SINOPSIS is run, and change the parameters of interest. You can use any file name, including the original one: the most important thing is that it needs to be specified here;

27. `Number of different metallicity values`. It is the number of the different metallicities of the SSP which will be used in the best–fit search. Note that 3 is currently the maximum number available;

28. `Metallicity values to be used`. Specifies the values of the SSPs metallicity. Only three values are currently supported (that is with the use of the Jacobi–Miles option): Z=0.004, Z=0.02 (solar), and Z=0.05. The number and the values of the available metallicities depends, in any case, on the SSP set which is used.

29. `Smooth SSP spectra to the observed spectra resolution`. Allows the SSP models to be smoothed in order to match the resolution of the observed spectra;

30. `Set maximum age of the stellar populations`. Used to fix the age of the oldest stellar population. Three options are available:

    - `zage` **(d)**; the default option sets the oldest stellar population such that it is equal or or lower than the age of the Universe at the galaxy's redshift. If there is no SSP with the exact age value as the age of the Universe, then the immediately lower age is chosen;
    - maximum age; this value has to be given in units of yr, and will be the same for all the galaxies in the catalog;
    - redshift of formation: the user will provide here a value for the redshift of galaxy formation. The oldest age will be calculated as the difference between the age of the Universe at the galaxy's redshift, and the age of the Universe at the redshift of formation.

    Note that the code automatically distinguishes when a maximum age or a redshift of formation is given.

31. `Number of separate runs for each metallicity value`. This value, which has to be an odd number ≤ 11, is related to the calculation of the uncertainties (see Sect.9.12).

32. `Chi2 threshold value to calculate uncertainties`. Is the threshold coefficient value, τ_m applied to the minimum χ^2, relative to the minimum, for a model to be used to calculate uncertainties.

33. `Create a model magnitudes catalog`. It is possible to have, as an output, a catalog of model magnitudes which extend on a wider spectral range with respect to the coverage of the observed spectra.

34. `Magnitudes type for the model catalog`. In case the value of the previous question is `yes`, you will need to specify the type of magnitude for the aforementioned catalog. Two values are currently supported: `Jon (d)` (for Johnson magnitudes, calibrated with the SED of Vega), and `AB`;

35. `Redshift value below which a redshift-independent distance is used`. This is the threshold value below which SINOPSIS looks in the appropriate file for a distance value;

36. `Catalog of redshift-independent distances to be used`. The calculation of the distance of a galaxy is performed exploiting the Hubble law. In case an object is too close, and the Hubble-Lamaitre law does not apply, SINOPSIS will look for a value of the distance provided within this file. See Sect. 4.3 for more information.

37. `Contribution of [NII] to Halpha flux`. This parameter can assume values between 0 and 1 (extremes included), and is used in case the measurement of the Hα line is contaminated by the presence of the [NII] lines.

38. `Output all the best fits for all runs and metallicities`. If set to `yes`, SINOPSIS will write the details of the best–fit model calculated for every run at every metallicity metallicity values. The related files will be stored in separate directories;

39. `Write output file for each reference model`. Will write (or not) an output file (extension `.out`) for each spectrum, giving details of the features which are reproduced by the model, how well they are reproduced, and the values of the values of the model parameters.

40. `Write out the model spectra without emission lines`. If set to `yes`, SINOPSIS will write, in the `.spec` files which contain the spectrum of best fit model, a third column containing a model spectrum calculated neglecting SSP spectra younger than $2 \cdot 10^7$ years (i.e. those containing optical emission lines). If possible, this model spectrum will be using the pure stellar spectra.

41. `Write out the flag mask of fitted pixels in a cube dataset`. When spectra in the form of a datacube are used, it is possible to create a mask of the pixels which are fitted and those which are not, specifying also why.

42. `Write out the contribution to the continuum flux from the SSPs`. This option will write the contribution of stars in different age bins to the total flux of the spectrum. At the moment this option is only available when using datacubes.

43.

44. `Minimize memory usage`. When set to `yes`, most of the tables which are written out at the end of the run, are instead written as hidden files, so that the codes does not need to keep them loaded in the RAM. When all the spectra are fitted, they are read, converted into `.fits` file, and saved;

45. `Include dusty SSPs`. If set to `yes`, photometric bands at $\lambda > 3$ μm will be used as a further constraint, and dusty SSP templates will be included in the fit. Note that this option is currently not fully supported yet.

4.3 Using redshift-independent distances

In the case you are dealing with very nearby galaxies, the redshift alone might lead to an incorrect luminosity distance estimate, as proper motions give a significant, non-negligible contribution to the Hubble flow. To properly deal with these objects, starting from version 1.5.0, the possibility of using a redshift-independent estimate of the luminosity distance was introduced. The 35th option line of the configuration file (see Sect. 4.2) will be used to specify the filename of the list of distances and, on the same line, a threshold redshift will have to be provided (line 36 of the config file), below which SINOPSIS will not calculated the luminosity distance from the redshift anymore, but will instead use the value provided in the aforementioned file.

The structure of the file is simple: it will contain 2 columns. The first is the name of the spectrum, in the same identical format as in the input catalog file (see Sect. 4.1), and the second will contain the distance value you wish to use for that spectrum, expressed in Mpc. Note that you do NOT need to specify the distances for all the spectra, but only for those for which you do not wish to use the redshift as a distance estimator. The code will take care of the matching. An header with an unlimited number of lines, each one starting with a #, is allowed.

5 The Physics and the parameters

5.1 Star formation history

The code currently allows 2 main Star Formation History (SFH) patterns: Free–Form (*FF*), and Analytic (*An*). In the *FF* approach, SSP averaged spectra of up to 12 different ages are used, and both the SFR and the extinction, parametrized by means of the $E(B-V)$, of each one of them is let completely free to vary, independently of the age. In this way, the total number of parameters in play can be up to 24 (2×12, SFR and $E(B-V)$ values).

A second approach tries to limit the number of parameters, by assuming an analytical form for the SFR as a function of the cosmic time. A double, decaying exponential form is used to calculate both the SFR and the $E(B-V)$ at each age. The first peak represents the burst episode which built the bulk of the stellar mass, and the second represent a subsequent burst:

$$SFR(t) = \left(\frac{T_U - t}{T_U}\right)^{n_1} \cdot \exp\left(-\frac{T_U - t}{\tau_i T_U}\right) + M_B \cdot \left(\frac{T_B - t}{T_B}\right)^{n_2} \cdot \exp\left(-\frac{T_B - t}{\tau_B T_B}\right) \qquad (2)$$

where T_U is the age of the Universe at the galaxy's redshift (i.e. the age of the oldest SSP), τ_i is the initial star formation burst duration, M_B is the percentage of stellar mass formed in the second, more recent, burst, T_B is its age (i.e. the age of the stellar populations when it begins) and τ_B is its duration. t is the stellar populations age.

Extinction is calculated in a similar fashion: it is considered constant until the burst –if any– begins, and then it assumes an exponential trend, according to the following:

$$E(B-V)(t) = EBV_0 + EBV_1 \cdot \left(\frac{T_B - t}{T_B}\right)^{n_3} \cdot \exp\left(-\frac{T_B - t}{\tau_B T_B}\right) \qquad (3)$$

These equations, if we assume that a fixed age of formation (T_U), make a total number of free parameters to be fitted of 9.

A third possible formalism, yet analytical, to represent the change in the SFR across the time is now implemented, and it adopts a log-normal pattern, according to the following:

$$SFR(t) = \frac{1}{(T_U - t) \times \sqrt{2\pi\tau_i^2}} \cdot \exp\left(-\frac{[\ln(T_U - t) - T_0]^2}{2\tau_i^2}\right) \qquad (4)$$

where the meaning of the variables is similar to that given for Eqn. 2. In this case T_0 is the (natural) logarithm of the delay time.

As for the extinction pattern, we allow also in this case the attenuation to vary as a function of the SSP's age, and we have parametrized it with the following:

$$E(B-V)(t) = EBV_0 + \frac{EBV_M - EBV_0}{2} \cdot \exp\left(-\frac{t - t_E}{t_E} \cdot \ln 2\right) \qquad (5)$$

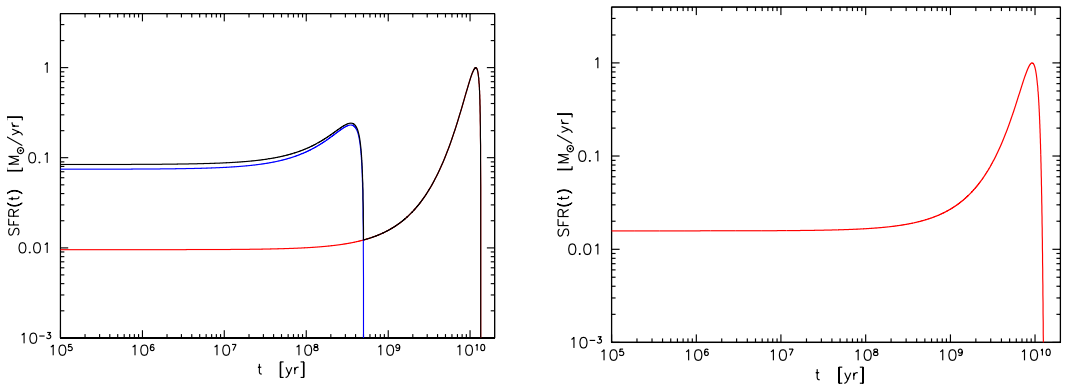

Figure 1: Left panel: Example of a double exponential star formation history. The red and blue lines are the star formation histories corresponding to the first and to the younger burst of star formation, respectively. Right panel: example of a log–normal pattern.

where EBV_M is the maximum extinction (for the youngest SSP), EBV_0 is the (fixed) extinction value for the oldest stars, and t_E is the age of the SSP for which the extinction is not constant any more, and it increases with an exponential trend towards the youngest stars.

5.2 Extinction

The effect of extinction, which is generally considered as a quantity varying as a function of the SSP's age, is treated by assuming that dust is placed in front of the stars, and distributed like a uniform screen. While it is more likely that having a "mixed" geometry, where stars and dust occupy the same regions, is the most physical representation of reality, accounting for geometrical effects in a proper way would require a full radiative transfer treatment of the issue. An attempt to describe the effect of having dust and stars mixed together on the resulting SED was made by Calzetti et al. (1994), where they try to take into account the effect of scattered starlights. This approach is currently not implemented in the model. Furthermore, Liu et al. (2013) have demonstrated that adopting a slab, foreground screen is a fairly good representation of the dust effect on large scales.

Instead, it is possible to use extinction curves with different values of R_V plus, the attenuation law derived by Calzetti et al. (1994) for local starburst is implemented. Using a different extinction curve can be done by using the appropriate keyword in the `config.dat` file. What follows is the list of the options available in version 1.0.0., together with the keywords which adopted since version 1.1.0.

Keyword	filename	description
MW	`al_av.dat`	Milky Way
SMC	`al_av_smc.dat`	Small Magellanic Cloud
CAL	`al_avcalz.dat`	Calzetti et al. (1994) attenuation
2.5	`al_av2_5.dat`	Extinction curve with $R_V = 2.5$
4.0	`al_av4.dat`	Extinction curve with $R_V = 4.0$
5.0	`al_av5.dat`	Extinction curve with $R_V = 5.0$

Table 7: List of the extinction curves currently available.

5.3 The SSP set

SINOPSIS uses three sets of SSP spectra with metallicity of Z=0.004, Z=0.02 (solar), and Z=0.05. Each of them contains 108 theoretical spectra of different ages ranging from 10^5 to $20 \cdot 10^9$ yr. By default, the oldest SSP spectrum which can be used has an age of 14.1 Gyr, which is fairly comparable with the age of the Universe. In Sect. 10 I will give more details on how to change this number.

The SSPs that are currently available are built from the Padova isochrones Bertelli et al. (1994), using the Jacoby et al. (1984) libraries, for ages younger than 1 Gyr, which are extended to the Infrared and to the UV by means of Kurucz models. Older SSPs use the MILES library instead. The Initial Mass Function (IMF) is a Salpeter (1955), with stellar mass in the limits 0.15 to 120 $M_\odot$. These SSPs have been processed with CLOUDY (Ferland, 1996) to calculate the intensity of emission lines, which characterize the theoretical spectra up to a stellar age of $\sim 2 \times 10^7$.

Optical spectra of SSPs younger than $\sim 2 \times 10^7$ display both permitted and forbidden emission lines (typically, hydrogen, [OII], [OIII], [NII] and [SII]). This nebular component was computed assuming *case B recombination* (see Osterbrock, 1989), an electron temperature of 10^4 K, and an electron density of 100 cm^{-3}. The radius of the ionizing star cluster was assumed to be 15 pc, and its mass 10^4 $M_\odot$. Finally, emission from the circumstellar envelopes of AGB stars was computed and added as described in Bressan et al. (1998). Note that the nitrogen line at 6548 Å is not calculated and hence is not included in the spectra.

If the *FF* option is chosen for the star formation history, then the 108 SSPs spectra, within one metallicity set, are binned in 12 average SSPs, at the very beginning of the code. These 12 age bins have been chosen in such a way that the spectra which form a given bin have similar characteristics i.e., colours, equivalent width of the most prominent lines, and the Balmer decrement. These bins are specified within the file `ssp_setup_ff.dat` located within the `data/` directory. When creating these 12 averaged SSPs, they are summed at a **constant stellar mass** (that is, not necessarily constant SFR). Then, they are renormalized in such a way that each averaged SSPs represent the emission of 1 $M_\odot$. Changes

to this file must be made taking into account the set of available ages. Again, before you change this, make sure you know what you are doing.

When using an analytic description of the SFR as a function of the cosmic time (such as a lognormal or an exponential function), each of the synthetic spectra is used and no age–binning is done.

6 Equivalent widths

A crucial characteristic of SINOPSIS is a proper measurement of the equivalent width (EW) of spectral lines. SINOPSIS measures, by default, 14 EW and uses the values of 7 of them (if present and/or measurable) to constrain the best fit model. Table 8 presents a list of the lines that are measured and those that are also used as constraints.

Line	λ_C [Å]	Constraint?
[OII]	3726.03+3728.81	✓
Hθ	3797.90	✗
Hη	3835.38	✓
Hζ	3889.05	✗
CaII (H)	3933.66	✓
CaII (K)+Hϵ	3968.47+3970.07	✓
Hδ	4101.73	✓
CO G-band	4305.61	✗
Hγ	4340.46	✗
Hβ	4861.32	✓
[OIII]	5006.84	✗
Mg	5176.70	✗
Na (D)	5889.95+5895.92	✗
Hα	6562.80	✓

Table 8: List of spectral lines that are measured by default. On the rightmost column a flag indicates which ones are used as constraints by default (✓) and which ones are simply measured (✗). This can be changed as explained in the text.

Properly measuring a EW value, requires a solid definition of the continuum underlying the spectral line. Normally, instead of measuring the continuum exactly below the line, it is calculated by averaging the continuum values measured in two bands located on a blue and red portion of the spectrum sufficiently close to the line to measure, avoiding the inclusion of other strong spectral lines (if you are familiar with optical spectra you surely will think about Hα and the two [NII] lines). This might be a problem if the spectrum is very noisy. With the aim of mitigating this issue, two different methods are implemented to measure EW: I will present them in the following sections.

6.1 The variable FWZI method

This method was first presented in Fritz et al. (2007, 2011) with the goal of optimizing the EW measurement in WINGS spectra (Cava et al., 2009) and making it as close as possible to a "manual" one. When using this method, the continuum emission is defined

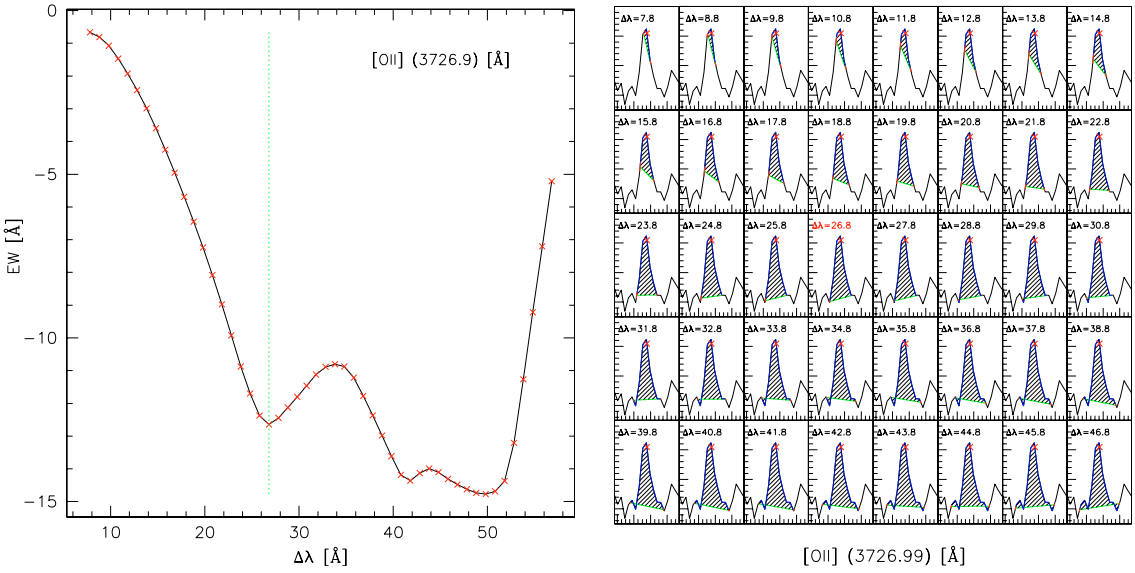

Figure 2: Example of how an emission line is detected, on the [OII] line. On the left-hand panel, the EW value as a function of the $\Delta\lambda$ (trend curve). On the right-hand panel a representation of how the line is measured at increasing values of the $\Delta\lambda$. The value of $\Delta\lambda$ at which the EW is measured corresponds to the first point for which the first derivative of the trend curve is 0, and it is indicated by a green, vertical line (left panel). Figure taken from Fritz et al. (2007)

in two narrow bands whose length is chosen depending on which line is measured. These bands are located respectively on the blue and red portion (with respect to the line) of the spectrum adjacent to the spectral region defined by the Full Width at Zero Intensity (FWZI) of the line itself. Finding the proper measure of a line's EW, hence turns into finding a proper definition of the FWZI (hereafter also defined as the $\Delta\lambda$). Adopting a too small value will turn in an underestimation of the EW value, while a too large one might be affected by the presence of other close lines or by noise features.

To chose the best value of the FWZI ($\Delta\lambda$), SINOPSISmeasures the EW value adopting several $\Delta\lambda$ values, from small ones, allowing a partial measurement of the line (i.e., the central part, see Fig. 2), up to large one, encompassing a spectral region much wider than that defined by the line. Then, the EW values thus obtained are analyzed by means of "trend curves", displaying the changes in EW as a function of the $\Delta\lambda$. Two such examples are shown in Fig. 2 (for the [OII] line) and Fig. 3 (for the Hβ line).

Several criteria have been implemented when analyzing the trend curve to detect whether a measurement really correspond to a real line detection or if it is noise–dominated.

The advantage of this method is that it provides quite reliable measurements in noisy

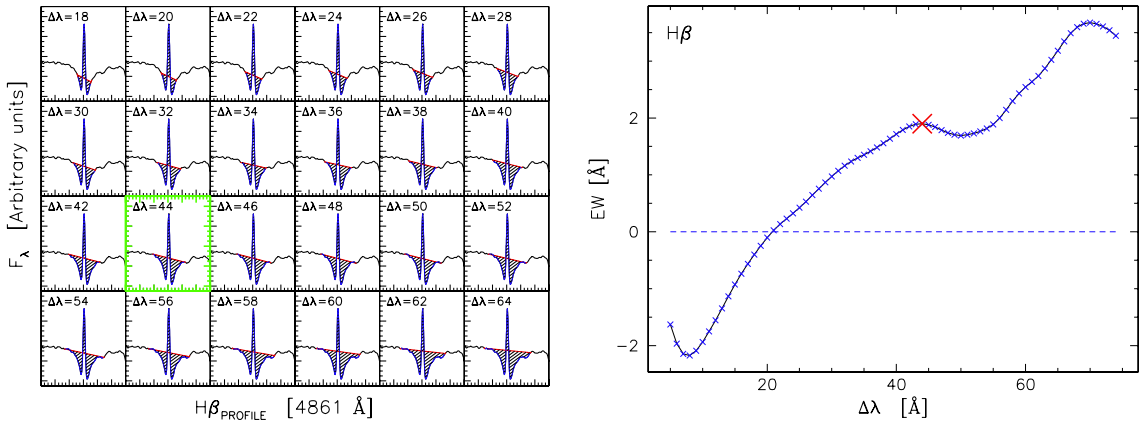

Figure 3: Example of how an "emission+absorption" profile is detected, on the Hβ line. On the left-hand panel a representation of how the line is measured at increasing values of the $\Delta\lambda$. On the right-hand panel, the trend curve is shown. For this particular case, the EW is measured at a $\Delta\lambda$ value that corresponds to the second point where the first derivative of the trend curve is 0. Figure taken from Fritz et al. (2014).

spectra and, furthermore, it is capable of detecting noisy features that might be mistaken as real spectral lines. The results obtained applying his method have been compared with visual/manual measurements, yielding a very good comparison.

Every spectral line from the ones listed in Table 8 has, in general, different parameters that have been optimized for its measurement. These parameters can be found in the `data/` directory in the `spec_lines_param.dat` file. Here we will briefly discuss the meaning of each of the parameters, so that the user can modified them at their willing. However, it should be kept in mind that they were already chosen in order to optimize the measurements. Hereby we present a representation of the file content (note: for formatting reasons I have limited the lines parameters displayed below to 9; nevertheless, the other 5 lines adopt a similar scheme):

```
14
  [OII]       Hth       Het        Hz       CaK       CaH        Hd       GCO        Hg
3727.00   3797.90   3835.00   3889.00   3933.70   3969.00   4101.70   4305.60   4340.50
   4.00      6.00      6.00      8.00      6.00      6.00      6.00     10.00      6.00
   0.10      0.10      0.10      0.80      1.20      1.00      0.20      0.40      0.40
   1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00
   0.10      0.10      0.10      0.10      0.10      0.10      0.10      0.10      0.10
      1         0         1         0         1         1         1         0         0
```

Here's the meaning of the content of each line:

1. the total number of lines that SINOPSIS will try to measure. Having a number $n < 14$ (14 is the default number of lines, see Tab. 8) will cause the code to only measure the first n lines only;

2. the label identifying each line. This is the way SINOPSIS recognize how to measure each lines, hence they should not be modified. Adding a line to measure must be hardcoded;

3. the central wavelength of the line, used for the measurement (it is assumed that the lines' profile are symmetric);

4. the initial $\Delta\lambda$, that is the value of the FWZI used to measure the first EW. It may vary depending on the line, e.g. broad lines such as the CO G band will have a higher value of this parameter;

5. $\delta\Delta\lambda$: this parameter defines how wide will the continuum bands be (note: they are NOT the bandwidths);

6. this is the weight assigned to the χ^2 for each line when used as a constraint. Hence, if you want to be sure that a particular line is reproduced, you might increase this value to 2-3;

7. this line contains the minimum uncertainty in each line measurement, expressed in Å (used for χ^2 calculation);

8. last line is used to specify which line is used (1) or not (0) as a model constraint.

Note: labels for the other default lines are:

```
Hb  [OIII]  Mg  NaD  Ha
```

This method is applied to all lines apart from Hα in emission, as it is intrinsically difficult to measure due to the proximity of the [NII] doublet (at $\sim$ 6548 and $\sim$ 6584 Å). Especially for low spectral resolution data or in objects for which the lines are intrinsically broad, using the trend curve analysis method might lead to a wrong measurement. To deal with this case, the code first checks whether the line is in emission or absorption by performing the EW measurement with a narrow $\Delta\lambda$ hence allowing the detection of a possible emission. If this is the case, then the EW is measured by adopting a $\Delta\lambda = 72$ Å, meaning that the resulting value will also include the two nitrogen lines. As these lines are also included in the models the SFR estimated by SINOPSIS should still be reliable (provided that the nitrogen lines intensities are typical of those observed in star forming regions).

The EWs of the model spectrum are measured using the same $\Delta\lambda$ used for the observed spectrum; furthermore, all the values are measured at the redshift of the observed spectrum hence, in order to get rest-frame values, they need to be divided by $(1 + z)$.

6.2 The fixed FWZI method

This is a much more classical method to measure an equivalent width. For each of the 14 lines a $\Delta\lambda$ is defined together with the two narrow bands that are used to calculate the average continuum flux. These are values that are customized for each line, and they can be found in the file `fix_lines_param.dat` located in the `data/` directory.

Changing these parameters can be easily done, and this is something that can be specified in the `config.sin` file. As for the previous case, I am hereby reporting the content of the `fix_lines_param.dat` file together with a short explanation, so that it should be easier for the user to customize the measurements if needed.

```
# line     lc         use   weig   errmi     l1      l2      l3      l4
[OII]      3727.00    1     1.0     0.1   3668.1  3718.6  3737.9  3774.4
Hth        3797.90    0     1.0     0.1   3776.0  3790.5  3806.9  3822.9
Het        3835.00    1     1.0     0.1   3807.3  3823.3  3846.2  3861.2
Hz         3889.00    0     1.0     0.1   3871.6  3876.1  3902.6  3912.6
CaK        3933.70    1     1.0     0.1   3908.9  3923.4  3945.6  3955.1
CaH        3969.00    1     1.0     0.1   3944.9  3956.1  3982.8  3995.7
Hd         4101.70    1     1.0     0.1   4076.4  4088.8  4117.2  4136.7
GCO        4305.60    0     1.0     0.1   4272.7  4293.2  4316.5  4328.0
Hg         4340.50    0     1.0     0.1   4314.3  4328.8  4357.2  4381.5
Hb         4861.30    1     1.0     0.1   4806.0  4826.0  4896.0  4918.0
[OIII]     5007.00    0     1.0     0.1   4985.5  4997.7  5018.2  5035.8
Mg         5176.70    0     1.0     0.1   5136.6  5158.3  5190.0  5213.4
NaD        5894.00    0     1.0     0.1   5862.0  5883.7  5902.6  5925.9
Ha         6562.80    1     1.0     0.1   6505.0  6535.0  6595.0  6625.0
```

For each line, the meaning of the columns is the following:

1. the label to identify the line;
2. the wavelength of the line;
3. a flag specifying whether to use (1) or not (0) the EW as a constraint for the model;
4. the minimum error on the measurement (in Å);
5. wavelength defining the lower limit of the blue band used to define the continuum;
6. wavelength defining the upper limit of the blue band used to define the continuum;
7. wavelength defining the lower limit of the red band used to define the continuum;
8. wavelength defining the upper limit of the red band used to define the continuum.

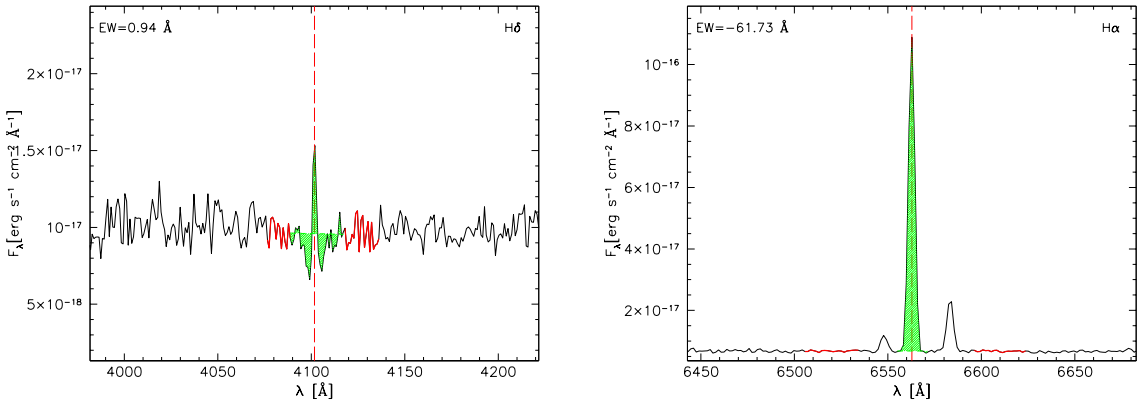

Figure 4: Examples of EW measurements with fixed FWZI. The red, dashed line marks the line nominal center. The red, continuos lines marks the band where the continuum emission is calculated. The green dashed region represent the area used to calculate the EW. On the left-hand panel, measurement of the Hδ line with an emission+absorption profile. On the right-hand panel, measurement of Hα. In this case the continuum emission is defined using two bands located farther away from the line.

In Fig. 4 two example of the measurements are shown. The values in columns 5, 6, 7, and 8 ($l1$, $l2$, $l3$, and $l4$) correspond to the wavelengths delimiting the portions of spectral continuum colored in red. In this case we have that $\Delta\lambda = l3 - l2$ for all lines except for Hα (see the right panel of Fig. 4). This choice is dictated, once again, by the possible presence of the nitrogen doublet lines. When using this option, Hα is measured with a $\Delta\lambda = 16$ Å (this is hardcoded).

6.3 The file with the equivalent width measurements

One of the outputs of SINOPSIS is a file with the EW measurements of the 14 aforementioned lines. This generally comes in the format of an ascii table or, in case data are in the form of a cube (such as IFU spectroscopy) they will be given in a cube as well.

The format of this file is quite standard: the first column will contain the name of the spectrum for which the lines have been measured. Columns from 2 to 15 will contain the EW values of the lines from the shortest wavelength to the longest wavelength. The two columns after the EW values contain the $D4000$ and the D_n4000, respectively. After this, the uncertainty on the measurements of each line is reported (following the same order as the EW measurement). Last column contains the average S/N of the spectrum.

Whenever one of the aforementioned quantities is not measured, it will be assigned a value of 99.00. Similarly, if the spectral range of the date does not allow a measurement, a value of 999.00 will be given to that variable.

Final note: adopting the standard convention, EW of emission lines have negative values, while those in absorption have positive ones.

7 Broad–band photometry as a further constraint

Photometric data points can be included in the observed constrains that the model is required to reproduce. For this section, we will limit ourselves to photometry at wavelengths shorter than 3.6 μm, at rest frame (e.g. the shortest bands of the *WISE* or *Spitzer* observatory). This is because, in general, dust emission can become important at larger wavelengths, and this is not yet included in the model, and it will be anyway treated in a different way, by exploiting different models.

In order for observed broad–band photometric data points to be properly included in the fit procedure, they should first of all be extracted on the same aperture as the spectrum, or as the observed magnitude which is used to normalize the spectrum. Then, in your input "catalog file", in the first row, you should add the number of photometric bands for which you have data. There is no limit in this number, as long as the bands are within the SSPs model range. The number indicating how many bands you want to use is the third one (see, e.g., Table 3), and when set to 0 only the spectrum will be considered.

This flag must be followed by the name of the used photometric bands, which must be strings of maximum 5 characters of length. Their number must of course coincide with that of the photometry flag. If for example you want to include 2MASS data, your header will look as follows:

```
3   Jon    J    H    Ks
```

indicating that you have data for three bands: J, H and K-short, and that the Units are Johnson magnitude (the `Jon` keyword; see below for other allowed units). The filter response curves for the majority of the photometric systems are already included in the database of SINOPSIS. The data must be placed, for each object, after the column specifying the normalization magnitude, even when this is not used, i.e. the 5th column, and they must follow the same order as the bands' name given in the header. Each observed value has to be followed by its uncertainty which, currently, must be a relative error (e.g. 8%, 15%, etc.), calculated on the flux (i.e. NOT on the magnitude, when this unit is used). Table 2 contains an example of the main input catalog file, where photometry is used. If data are not available for one or more (even all) of the bands which have been specified in the header, for one or more objects, then they must be replaced with a 99.99 flag. Like this, the code will skip the use of those point and rest on the ones which are available.

To calculate the model flux of a given pass-band, it is assumed that the adopted response filter curve are defined as the *contribution to the detector signal per photon entering the atmosphere of Earth* (see: `http://www.astro.ljmu.ac.uk/~ikb/research/mags-fluxes/`). In this way, the model flux F_λ^B in a photometric band B is calculated as:

$$F_\lambda^B = \frac{\int_0^\infty f_\lambda(\lambda) \cdot R(\lambda) \cdot \lambda \, d\lambda}{\int_0^\infty R(\lambda) \cdot \lambda \, d\lambda} \tag{6}$$

where $f_\lambda(\lambda)$ is the model SED, and $R(\lambda)$ is the response filter curve. This "monochromatic" flux is assumed to be emitted at a given wavelength (the *effective* λ or, in this case the "pivot" λ), λ_{Pivot}.

From version 1.4.0, SINOPSIS allows for the possibility of using data in a variety of units: Johnson magnitudes, AB magnitudes, fluxes in F_λ units ($[erg/s/cm^2/\text{Å}]$), and fluxes in F_ν units (both Jy and $[erg/s/cm^2/Hz]$). Declaring the photometric data units is done with a string keyword, which must be specified and has to be placed as the second item of the header in the catalog file. The allowed keywords are, at the moment:

1. `Fl` assumes the band flux is given in units of $erg/s/cm^2/\text{Å}$ (f_λ);
2. `Fnu` assumes the band flux is given in units of $erg/s/cm^2/Hz$ (f_ν);
3. `Jy` assumes the band flux is given in Jansky, that is, in units of $10^{23} erg/s/cm^2/Hz$ (f_ν);
4. `Jon` assumes the band flux is given magnitudes in the Johnson system;
5. `AB` assumes the band flux is given in AB magnitudes.

The units keyword is assumed to hold also for the normalization magnitudes, when used. Apart from the first case, all the others need a conversion which will bring them to the same units of the model (f_λ), and that is performed as follows:

2. $f_\lambda = \dfrac{c}{\lambda^2_{Pivot}} \cdot f_\nu$;
3. $f_\lambda = 10^{-23} \cdot \dfrac{c}{\lambda^2_{Pivot}} \cdot f_\nu$;
4. $f_\lambda = f_0(\lambda) \cdot 10^{-0.4 \cdot m}$;
5. $f_\lambda = \dfrac{c}{\lambda^2_{Pivot}} \cdot 10^{-0.4 \cdot (m_{AB} - 48.6)}$;

where c is the speed of light in $\text{Å}/s$, f_0 is the zero-magnitude flux, calculated based on the SED of Vega (from Kurucz 1993 models), m is the Johnson magnitude, and m_{AB} is the AB magnitude.

Important note: using photometric data is currently only possible for spectra given in `asciii` or single `fits`. The datacube option will be soon available, while the 2D is currently under testing.

8 Mid- and Far-Infrared as constraints.

This is an option which will be introduced at some point in the future.

9 Outputs

SINOPSIS is designed in such a way that it allows you to plot and have a look at the results while you are still running the fits. Each time you run the code on a dataset, it produces a number of files, some of which are used as log/record files, some contain your results, and some contain models or modelled quantities. Files containing information for all objects (e.g. the main output results catalog) are updated as SINOPSIS goes through the fit. In this section we will go into the details of each output file.

9.1 The main catalog

The main catalog file contains a summary of all the results from the fit for all the spectra. It is called "`listfile.out`" (where "`listfile`" is the name of the input file containing the list of the observed spectra). More specifically, it contains the derived properties for all the galaxies, each row containing values which refer to the reference model for a given observed spectrum. An explanation has been already given for most of the values which are given in each column in this file, so here I will only briefly summarize their meaning. I have kept the order in which each column is given:

1. `objname`: filename (without extension) of the spectrum;
2. `Dl`: luminosity distance;
3. `z`: redshift;
4. `redchi`: χ^2 of the reference model;
5. `Z`: best–fit metallicity;
6. `nrun`: run identification of the reference model;
7. `neqw`: number of measured equivalent widths;
8. `Av_y`: extinction of the youngest stellar populations (age $\leq 2 \cdot 10^7$ yr);
9. `Av_y_m`: minimum value of Av_y;
10. `Av_y_M`: maximum value of Av_y;
11. `Av`: average extinction value for all the stellar populations;
12. `Av_m`: minimum value of Av;
13. `Av_M`: maximum value of Av;
14. `sfr1`: SFR in the first main age-bin;

15. `sfr1_m`: minimum value for SFR1;
16. `sfr1_M`: maximum value for SFR1;
17. `sfr2`: SFR in the second main age-bin;
18. `sfr2_m`: minimum value for SFR2;
19. `sfr2_M`: maximum value for SFR2;
20. `sfr3`: SFR in the third main age-bin;
21. `sfr3_m`: minimum value for SFR3;
22. `sfr3_M`: maximum value for SFR3;
23. `sfr4`: SFR in the fourth main age-bin;
24. `sfr4_m`: minimum value for SFR4;
25. `sfr4_M`: maximum value for SFR4;
26. `Mb1_3`: percentage of stellar mass in the first age bin (mass calculated according to definition n.3);
27. `Mb2_3`: percentage of stellar mass in the second age bin (mass calculated according to definition n.3);
28. `Mb3_3`: percentage of stellar mass in the third age bin (mass calculated according to definition n.3);
29. `Mb4_3`: percentage of stellar mass in the fourth age bin (mass calculated according to definition n.3);
30. `Mb1_2`: percentage of stellar mass in the first age bin (mass calculated according to definition n.2);
31. `Mb1_2_m`: minimum value of Mb1_2;
32. `Mb1_2_M`: maximum value of Mb1_2;
33. `Mb2_2`: percentage of stellar mass in the first age bin (mass calculated according to definition n.2);
34. `Mb2_2_m`: minimum value of Mb2_2;
35. `Mb2_2_M`: maximum value of Mb2_2;

36. `Mb3_2`: percentage of stellar mass in the first age bin (mass calculated according to definition n.2);

37. `Mb3_2_m`: minimum value of Mb3_2;

38. `Mb3_2_M`: maximum value of Mb3_2;

39. `Mb4_2`: percentage of stellar mass in the first age bin (mass calculated according to definition n.2);

40. `Mb4_2_m`: minimum value of Mb4_2;

41. `Mb4_2_M`: maximum value of Mb4_2;

42. `Mb1_1`: percentage of stellar mass in the first age bin (mass calculated according to definition n.1);

43. `Mb2_1`: percentage of stellar mass in the second age bin (mass calculated according to definition n.1);

44. `Mb3_1`: percentage of stellar mass in the third age bin (mass calculated according to definition n.1);

45. `Mb4_1`: percentage of stellar mass in the fourth age bin (mass calculated according to definition n.1);

46. `AMass3`: Stellar mass, calculated according to definition n.3, normalized to aperture (magnitude or spectrum);

47. `AMass3_m`: minimum value of AMass3;

48. `AMass3_M`: maximum value of AMass3;

49. `TotMass3`: Total stellar mass, calculated according to definition n.3;

50. `AMass2`: Stellar mass, calculated according to definition n.2, normalized to aperture (magnitude or spectrum);

51. `AMass2_m`: minimum value of AMass2;

52. `AMass2_M`: maximum value of AMass2;

53. `TotMass2`: Total stellar mass, calculated according to definition n.2;

54. `AMass1`: Stellar mass, calculated according to definition n.1, normalized to aperture (magnitude or spectrum);

55. `AMass1_m`: minimum value of AMass1;

56. `AMass1_M`: maximum value of AMass1;
57. `TotMass1`: Total stellar mass, calculated according to definition n.1;
58. `lVwage`: logarithm of the luminosity–weighted age (V–band);
59. `lVwage_m`: minimum value of lVwage;
60. `lVwage_M`: maximum value of lVwage;
61. `lwage`: logarithm of the luminosity–weighted age (bolometric);
62. `lwage_m`: minimum value of lVwage;
63. `lwage_M`: maximum value of lwage;
64. `mwage`: logarithm of the mass–weighted age;
65. `mwage_m`: minimum value of mwage;
66. `mwage_M`: maximum value of mwage;
67. `Parameters::` these columns contain either the values of the SFR for each SSP (in the *FF* option) or the parameters of the star formation law.

In case the input data are in the form of a datacube (e.g. integral field data), this file will be given as a `.fits` cube, with the name in the form `listfile_out.fits`. Each plane of the cube will contain the information described above, pixel by pixel.

9.2 The equivalent width catalog

One of the main products of SINOPSIS consists in a catalog of measurements of equivalent widths for the most prominent spectral lines. Equivalent width values are used as constrains for the model spectrum, and the list of lines measured is defined in the file `spec_lines_param.dat` located in the directory `data/`.

An arbitrary number of lines to be measured can be added, provided the list follows the same exact format of the original one. Note that not all of the line are used as constraints, some are just measured (e.g. the [OIII] line at 5007 Å). The measurements are contained in the `listfile.eqw` (where "`listfile`" is defined as above). The structure of the file is as follows: the first column contains the name of the spectrum, while in the following columns the observed (i.e. NOT rest-frame; so, they need to be corrected by dividing by a $(1 + z)$ factor before making any analysis) values of the equivalent width are reported, expressed in Å (in order of increasing wavelength). The default set contains 14 lines (see Fritz et al., 2014, for a complete list and characterization of them).

Then, the value of the D4000 and D_n4000 indexes (see Bruzual A., 1983; Balogh et al., 1999, respectively, for their definition), are reported and, after those, the uncertainties on

the lines' measurement (these values follow the same order as the measurements). Last column contains the Signal-to-Noise calculated over the entire spectrum.

If the input data are in the form of a datacube, the table described above will be given in a `.fits` file, named `listfile_eqw.fits`, with a similar format to the input datacube, with all the aforementioned values in different planes, for each pixel.

9.3 The (model) magnitudes catalog

Using an SSP set which spans the whole electromagnetic spectrum from far-UV to radio wavelength, allows us to calculate the predicted magnitudes, both observed and absolute, of a galaxy. We are basically performing a somehow physical motivated extrapolation of the optical spectrum to longer wavelength. The advantage of this approach is that you do not need to apply a K-correction when calculating absolute magnitudes. While the values calculated in this way are still dependent on the derived SFH which, in turn, only relies on the information carried by the optical spectrum and is anyway prone to model degeneracies, they do not have to rely on the morphological classification to estimate the amount of K-correction.

The model magnitudes, both observed and absolute, are computed by convolving the filters response curves with the spectrum. Changing the number and the list of filters for which the magnitudes are calculated can be done by simply adding them in the file `filter_list.dat`, which is located in `SINOPSIS/data/filters/`. In case new filters are added, please make sure the format is the same as those already provided with this version of the code.

The user can choose whether to calculate these quantities directly from the configuration file (see Sect.4.2). If she/he chooses to calculate them, a file called "`listname.mag`" will be created.

Again, if the input data are in the form of a datacube, these quantities will be in the same format, with each plane containing the pixel by pixel magnitudes as described above.

9.4 The model spectra

When observed spetra in `ascii` or 2D `fits` format are used, SINOPSIS will produce, for each spectrum in a separate file, the panchromatic model calculated from the best-fit (or, better, from the reference model). These are store in a file called "`specname.spec`" (where "`specname`" is the file name of the observed spectrum in the fitted sample, without extension) which contains the (observed) wavelength (in Å) in the first column and the flux (in units of erg/s/cm^2/Å) in the second column. These files can be used to plot the best fit model overlapped to the observed spectrum. These fluxes are those which are used to calculate the model magnitudes. If the corresponding option is selected, a third column is added two the first two ones just described, containing the model spectrum but without emission lines. This can be used to correct the gas mission for the underlying absorption.

When the observed spectra are taken from a datacube, the model spectra will be given in the same format. The mode spectra without emission lines will be found in a different file. In this case, when 2 files are produced, the names will be in the form `listfile_model.fits` and `listfile_model_noline.fits`.

9.5 The smoothed observed spectra

On option recently introduced in SINOPSIS is the possibility of smoothing either the SSPs or the observed spectra to a lower resolution, in order to match the spectra at the worse resolution. In case the SSPs are smoothed, the final model spectra will be written in the "`specname.spec`" files, at the degraded spectra. In case the observed are those put at a lower resolution, the use can choose to have them written as well (see Sect.4.2) in files containing two columns: the first is the wavelength (in Å) and the second is the flux (at the smoothed resolution). The file names will be in the form `specname.smooth`

9.6 The log file

A log file (`listfile.log`) is also produced, containing a short summary of the simulation, such as the main parameters (the type of star formation history pattern, the extinction curve, the number of runs and of metallicity, etc.) and the starting and ending time and date.

9.7 The fitted pixel mask (datacube only)

In a file with the name of the form `listfile_fitmask.fits`, if the relative option is chosen in the `config.sin` file, a mask of the spectra which have been fitted is saved. It has three possible values:

- 1, for those pixels whose spectrum was possible to fit;
- 0, for those pixels whose spectrum was not fitted due to the absence of a redshift determination;
- -1, for those pixels having a spectrum with a signal too low to be significant (e.g., these spectra typically have 40% or more of the flux datapoints equal or below zero.

9.8 The skipped spectra file (single file spectra only)

If, for any reason, one or more spectra are not processed by SINOPSIS (e.g. because they display too noisy or negative fluxes for too many wavelengths points), a record will be kept in a file with the name of the form `listfile.skipped`. This is an equivalent to the fitted pixel mask.

9.9 The contribution to the total flux as function of the stellar age (datacube only)

When the relative option in the `config.sin` file is chosen, a mask whose name will be `listfile_fluxcont.fits`, will be created, with the same size of the original image, and having 4 planes, each containing the percentage of the flux contribution of the SSP in the main four age bins. This is useful when trying to determine whether a spectrum really contains a significant contribution from stellar populations of a given age, or if it is instead an artifact of the fitting procedure. This option is, for the moment, limited to the use of datacubes only.

9.10 The single run models

As explained in Sect. 2, several spectral fitting runs are performed in order to obtain a best fit model and to get uncertainties on the physical quantities. The spectral fits can be performed using up to 3 different SSPs set, each of them at a different metallicity (typically Z=0.004, Z=0.02, and Z=0.05, but this of course dependes on the SSP set you are using). This number and the metallicity values to be used, are a user's choice (see Sect. 4). The code will create 3 (but they can be less or even more, depending on the setup and on the adopted SSP set) directories named according to the metallicity values used (e.g. `Z004/`, `Z02/`, `Z05/`), each containing other subdirectories named with number from 1 to the number of runs used for calculating the uncertainties (`1/`, `2/`, ..., `11/`). These directories contain other files, `specname_out.dat`, where the characteristics of the fits are written.

9.11 The age bin file

This file is used for plotting purposes only, and contains the definition of the main age bins that are considered meaningful for the star formation history reconstruction.

9.12 The uncertainty calculation

As explained in Fritz et al. (2007), and as it might be obvious for the reader, the solution of the problem is highly degenerate. Not only, in the free–free form case, we have up to 24 parameters, but the constraints (observed flux and equivalent widths of lines) are also non–independent. The fact that we might have more parameters than degree of freedom, does not necessarily make the problem solvable by default (by "solvable" here I mean that there are one or more combination of the parameters which will give a reasonable good fit). If, for example, the galaxy under scrutiny is a type–1 AGN, there is no way SINOPSIS can adequately reproduce its spectral features.

Calculating the uncertainties in the derived physical quantities becomes something more like looking for the most extreme values of these parameters which allow a good

fit. To do so, we exploit one of the characteristics of the ASA algorithm, that is the fact that each random move in the parameter space is decided based on a) the χ^2 value and b) the previous set of parameters. Hence, beginning the parameters space exploration from a different starting point, means that the algorithm will follow a completely different exploration path, ending up in a different final point (solution). We do this 11 times, and chose as a "best fit model" the model with the median total stellar mass among those explored.

10 Advanced Features

There is a number of options you might want to use to further tweak the spectral fitting. Here, I will go into the details of what you can change, and how the changes will affect the code, the fits, and the outputs. I have divided the changes according to what they will affect.

Note that, in some cases, when the changes deeply affect the behaviour of the code, you will need to recompile.

10.1 SSPs tweaking

There are several options, parameters and characteristics that can be changed of the Simple Stellar Population models and how they are treated.

10.1.1 Adding a new set of theoretical spectra

The first, most obvious option is using another set of theoretical spectra (e.g., with different IMF, isochrones, etc.). For this, you should make sure that:

1. the file format of the new set is the same as the one currently used. This implies that each SSP spectra, for a given age and a given metallicity, should be found in a single ascii files, with the first column containing the wavelength array, in units of Å, and the second column containing the luminosity array in units of 10^{30}[erg/s/Å/$M_\odot$];
2. there is no restriction on the file name;
3. the new SSP files can be placed in the same directory as the ones originally provided by SINOPSIS;
4. a new `ssplist.dat###` should be created, for each value of the metallicity, containing the list of the new SSP files, and some other infos (see below).

As mentioned above, the code will first read the files containing the list of available SSP. The name of the file that SINOPSIS recognizes, is `ssplist.dat###`, where the symbols "`###`" correspond to numbers used to identify the metallicity of the set. E.g., `ssplist.dat2` will contain the list of SSP spectra at solar ($Z = 0.02$) metallicity, while `ssplist.dat001` will contain the list of SSP spectra with $Z = 0.0001$.

Each ssplist file should start with a one-row header containing the spectral resolution (expressed as the FWHM in Å), between a specified wavelength range (the extremes of the range are given in the same row, and in units of Å). Hence, a comment line follows, and then 4 columns, whose content should be the following (the numbers identify the column number):

1. name of the SSP file;

2. age of the corresponding SSP, in [yr]

3. fraction of the mass of the core–burning stars;

4. fraction of the mass of stars+remnants.

Note that the latter two columns are used to calculate the total stellar mass according to definition 3 and definition 2, respectively.

If you used the `free-free` star formation option, the code will perform an age–binning, to reduce the number of the model spectra used. The age limits of the binning are defined in the `ssp_setup_ff_***.dat` file. They are usually 12, and they are defined according to the criteria explained in section 2.1 of Fritz et al. (2007).

When a new set of SSP is introduced, this file should be changed making sure that the values of the age (first column) correspond to actual age values of the SSP set. In Section 10.1.2 we will see the meaning of the rest of the column in this file.

10.1.2 SFH parameters

The Star Formation History parameters which are derived will depend, of course, on the kind of SFH prescription that you assume. As already explained in Sect. 5.1, SINOPSIS supports 2 different kinds of star formation history: Free–Form and Analytic. The first one in particular, allows the star formation rate of the stellar populations of each age, to assume any value, without them being correlated. The same goes for extinction.

It is possible to, at least partially, change this. Let us consider, for example, the 4 SSPs with the youngest stellar ages. They range, in the Jacoby–Miles models, from 2×10^6 to 2×10^7 years. Normally, when you calculate the SFR from the observed luminosity of recombination lines such as Hα, a model with a constant SFR over about 10^7 years is assumed. Hence, if you would like your results to be comparable with such estimates, you might want to have a constant SFR over this period as well. This can be very easily achieved by modifying the `ssp_setup_ff_jm.dat` file which is located in the `data/` directory. This file consists in a one–line header, followed by 9 columns. The meaning of each column, from the first to the last one, is as follows:

1. age of the SSP that are considered. Actually, these are the values of the age binning;

2. in the second column, an integer number tells you whether the corresponding parameter will be included in the minimization, and it is, hence, free to vary. These numbers, in the second column, refer to the extinction (the $E(B-V)$) value for the stellar population of the corresponding age. If any of these integer is set to 0, than the corresponding extinction value is **not** included in the minimization process and its value will be kept fixed to the initial value, which is reported in the fourth column for each age. As another option, if you set two of these integers to the same value, e.g. if you put the first and the second one equal to 1, then their values will be taken

to be the equal during the minimization process. Hence, the starting value (column 4) will have to be set to the same value.

3. the third column contains the minimum allowed value for extinction for each age;
4. the fourth column is the starting point value for extinction, to build the very first model in the minimization process;
5. the fifth column is the maximum allowed value for extinction;
6. column 6 to 9 contain the same parameters as column 2 to 5, except that they refer to the star formation rate value for each SSP.

One of the most obvious tweaking you can do, is to set a constant star formation rate over a given age range. Like this, for example, you can enable a comparison between the SFR as derived by SINOPSIS, and that calculated by other observables, i.e. extinction corrected emission lines luminosity (such as Hα or [OII]), which assume a constant SFR over about 10^7 years.

Let's say that you want to have a similar star formation history pattern. To have a constant SFR over a the aforementioned period, just set the first 4 numbers of column 6 to the same value, e.g., 13 changing, if necessary, the starting SFR value accordingly (i.e. so that the value is the same for all three SSPs).

In a similar way, if you want to force the code not to use SSP displaying emission lines, set those same number to 0, and the starting point (column 8) to 0 as well.

Of course, the same holds in case you want to play a similar game for the extinction parameter.

10.2 ASA tweaking

The core of the best–fit model search is the Adaptive Simulated Annealing (ASA) algorithm. It requires some input values, which are taken from the file `constants.f90`, namely:

1. *iter* is number of iterations after which the systems will undergo a decrease in its temperature;
2. *maxrun*: this is the maximum number of tries which the algorithm perform in the search of the minimum χ^2. When the number of tries exceed this value, the code automatically exits the minimization;
3. *idum0* is one seed for the random number generator (must be an integer), to decide about the following point where the model will be constructed;
4. *idum1* is a second seed for the random generator number (must be an integer as well) to decide whether to accept an uphill move (i.e. a model with a nhigher χ^2 value with respect to the current one) or not;

5. *eps*: this value is used after each model evaluation to decide, together with the merit function (the χ^2 in our case) whether the conditions for ending the optimization are met;

6. *temptr*: this is the initial "temperature", a parameter which controls the width of the change of each parameter value: the higher its value, the higher the allowed change;

7. *rt*: this is the change in the temperature which occurs after *iter* iterations. It regulates the "cooling" of the system;

Note that *idum0* and *idum1* are given as vectors, of size matching those of the number of runs used to estimate uncertainties. This is to allow the code to start from a different point and to follow a different path towards the best fit value for each different runs.

If you want a finer exploration of your parameter space, try to increase *temptr* and *iter*, and decrease *eps*. You will most likely get slightly better χ^2 value (but never significantly better), at the cost of much higher computation time. The default values of these parameters have been carefully chosen so to balance between the finding of a best–fit, and an acceptable CPU time. Increasing *rt* (note it must be lower than 1), will produce similar effects. Decreasing this value accelerates the convergence of the method, providing some gain in computational time and can be used if your dataset is of very high quality.

You might want to try to explore the uncertainties better. One of the possibilities for doing so, is to increase the number of runs. To do this, you should increase *maxrun* (remember it has to be an odd number), and add more values to the *idum0* and *idum1* vectors.

Note that modifications to this file will need a recompilation of the code to be effective.

11 Changes and modifications tracks

This section will be constantly updated with the changes on the code. The first version, 1.0.0, has been released to collaborators in 2015 on April, the 2^{nd}. Each following version will carry a different identification number. Minor changes to the main code, module or subroutines and function will result in an increment of the last figure of the version (e.g., from 1.0.0 to 1.0.1), with respect to the last one. The index will be reset to 0 in case of an increment of the preceding figure (e.g., from 1.0.10 to 1.1.0). Inclusion of new subroutines, function and/or modules marginally affecting the usage of the code, will be identified with an increment in the second figure in the version numbering. Any major change affecting the whole code is identified with an increment on the first figure.

11.1 Version 1.1.0

This version includes:

1. extinction curves in the `config.dat` are now specified through keywords (see 5.2);
2. a log file is now produced at the end of each simulation, containing a summary of the main parameters used for the fits;
3. the name of the file containing the observed values of the equivalent widths is now changed from `obs_eqw.dat` to `listfile.eqw`, where "`listfile`" is the name of the input file containing the catalog of spectra (extension not included).

11.2 Version 1.1.1

This version includes:

1. version number now displayed at the beginning of the run;
2. the `.log` file contains now more information;
3. more checks when reading the `config.dat` file, and specific feedback is given in case of incorrect format or wrong keywords.

11.3 Version 1.2.0

This version includes:

1. the `.eqw` files contains now also the average S/N of the spectrum. This is calculated in a new routine, `snr.f90`, following the method described in the ST-ECF Newsletter, Issue #42 (`www.spacetelescope.org/about/further_information/newsletters/html/newsletter_42.html`[3]). This number is reported in the last column of the

[3]This link is not freely available anymore.

`listfile.eqw` file, and is calculated over the entire observed spectrum. As a consequence, 2 other subroutine are added to the code package: `sort.f90` and `median.f90`;

2. a new optional keyword is added to be used in `config.dat`, beside the "`basic`" and "`advanced`": "`eqw`", which allows you to only measure the equivalent widths of the lines, the D4000 and D_n4000, and the newly introduced S/N. The only output file will hence be `listfile.eqw`. Note that, in case this option is chosen, the `config.dat` file is read only up to line 3, while the rest is skipped;

3. corrected the prescription for the lognormal SFH description. From now on, T_0 (see Eq. 4) enter directly the definition of the Star Formation History law. The boundaries for the allowed values have been changed accordingly (see the file `ssp_setup_lgn.dat` in the `data/` directory). τ_i (see same equation), will be limited in the range $4 \cdot 10^{-2}$ to 2.64 (range also set in `ssp_setup_lgn.dat`, and taken from Gladders et al., 2013);

4. added `elapsed_t` as a module variable. This represents the elapsed time since the big-bang at the age of each SSP. Used to speed up computation in `chi2comp` when using the lognormal SFH prescription;

5. a slight modification to the computation of rms in the continuum, to more properly calculate the uncertainties (see the `sigma_comp.f90` routine);

6. eliminated a double output message("`Best metallicity:  Z=`");

7. header names added for all the parameters of the log-normal prescription in the `listfile_OUT.DAT`;

8. change in the routine which calculates the SFR and the stellar mass in bins (final model and uncertainties), now done more efficiently;

9. some subroutine have been renamed for consistency issues.

11.4 Version 1.3.0

This version includes:

1. the header of the `basic`–form catalog has now a slightly different setup. The first line will still have to contain 3 integers, but now they will represent, respectively: the flag stating whether the wavelength array in the observed spectra is linear (0) or logarithmic (1), the same flag but for the flux array, and the number of photometric points which are available for the fits. After this the list of all the photometric bands should follow.

2. upgrade for using photometric data points in combination with the spectrum. The option is currently available only when the `basic` option is selected for the catalog. See Table **??** for an example of how to properly include photometry in the input catalog file.

3. it is now possible for the user to define the number and the ages of the "main age bins" based on which the star formation history is calculated. These are defined in the file `parameters.f90`: `nsfr_bin` is the number of bins (4 is default) and the age limits are defined in the `largebins` vector. IMPORTANT CAVEAT: **all of the age limits in the former vector MUST coincide with ages in the stellar populations vector.** These can be checked in the one of the files containing the list of SSPs name and ages (e.g. `ssplist.dat2`, located in one of the directories inside `data/ssp/`).

4. in order to be able to properly plot the Star Formation Rate values in these main bins (e.g. with the `sinopsis.m` macro) a new file is created, named with the same as the main input catalog, and having a `.bin` extension.

11.5 Version 1.3.1

This version includes:

1. corrected a bug which caused the code to crash if the '`eqw`' option is used;

2. added, in the main module, the subroutine `readdataspec3`, which initializes the data for the observed spectrum in case the '`eqw`' option is used.

11.6 Version 1.4.0

This version includes:

1. a bug is corrected in the subroutine `snr.f90`, which would cause the code to crash at random;

2. when launching SINOPSIS in "equivalent width measurement mode", this is now stated in a terminal message at the beginning of the process;

3. the code takes now into account for the possible presence of an header in the files containing the spectra. All the header's elements are currently identified by the character: #;

4. new lines are added to the list of the standard ones, mainly in the UV part of the spectrum: [NeV]a ($\lambda = 3345.9$ Å), [NeV]b ($\lambda = 3425.8$ Å), [NeIII]a ($\lambda = 3869$ Å), and [OIII]b ($\lambda = 4959$ Å). The new lines are added to the list in the `spec_lines_param.dat`, located in the `data/` directory;

5. a more precise method is now used to compute photometry (see details in Sect. 7);

6. different units are now accepted for photometric datapoints (a list of the accepted units and needed flag is provided in Sect. 7). A subroutine (`convert.f90`) has been included to perform this conversion;

7. the central band wavelength definition is slightly changed, and it refers now to the "pivot wavelength" (see Sect. 7). A new function (`lam_pivot.f90`) is added to calculate it.

Apart from these changes, an optimisation option is added to the compiler file, `compile.sh` (i.e. the `-Ofast` option; see, e.g., https://gcc.gnu.org/onlinedocs/gcc/Optimize-Options.html for further details), which enables most of the optimisations schemes.

11.7 Version 1.4.1

This version includes:

1. a bug is corrected which causes some output files to have an extra dot in their name. A new subroutine –`makename.f90`– is added, which creates the root name for output files from the input files;

2. a bug is corrected which causes, in some random cases and for a specific set of options, the total extinction, the luminosity and mass-weighted ages to be 0 in the output file;

3. different spectra normalization options, apart from that based on photometry, are re-introduced. They are currently 3:

   - `none`: no normalization applied, so no absolute quantities are derived (used, e.g., when fitting stacked spectra; this was formerly option `0`);
   - `spec`: all physical quantities are normalized so that the match the observed spectrum flux (this was formerly option `1`). When using this option there is no need of providing any information about photometric data, unless they are used as a further constraint (see also Sect 4);
   - `phot`: all physical quantities are normalized to a given photometric datapoint (this was formerly option `2`).

   As a consequence, in the configuration file `config.dat`, the $5th$ line must not an integer anymore, but it should instead be a 4-character string.

4. a new Vega SED is added, taken from Colina et al. (1996) (`vega_colina.dat`). As a consequence, the old Vega SED file (`vega.dat`) is now renamed to `vega_kurucz.dat`. At the moment it is possible to switch between the two of them only by commenting the proper line in the function `zero.f90`, and then recompiling the code again;

5. the continuum bands at each side of the spectral lines entering the χ^2 calculation (`minibands` in the code) are now eliminated, as they do not add significant constraints and, furthermore, might be source of noise in the χ^2 value;

6. more fun, with the new SINOPSIS random banners, popping up at the beginning of a simulation.

11.8 Version 1.5.0

This version includes:

1. the file `config.dat` does not exist anymore, now, and it has been substituted by a more appropriate `config.dat` file;

2. the `config.dat` has a structure which is basically the same as the formerly used `config.dat`, with the exception of two lines which are added after the number of runs per metallicity (i.e. after line 10; see following points for a description of these new features);

3. it is now possible to smooth the SSPs spectra to a resolution matching that of the observed spectra. To this aim, a line has been added in the header of the `ssplist.dat#` file, specifying the spectral resolution (i.e. the full width at half maximum, in Å) in a given range. For example, the SSPs currently supported have a resolution of 4.5 Å in the optical, more precisely between 3536 and 7412 Å, while outside this range it is much lower ($\sim$ 20 Å). These data are included in the files containing the list of available SSPs at a given metallicity. Hence, the first line of these files is something like the following:

```
4.5    3536.    7412. # fwhm between 3536 and 7412 ang.
```

   To this purpose, a subroutine is added, `gauss smooth.f90` has been added to the code, which takes a spectrum at a given resolution, and smooth it to a lower, user-defined, resolution, using a gaussian with a full-width at half maximum (FWHM) defined as:

$$FWHM_G = \sqrt{FWHM_N - FWHM_O} \tag{7}$$

   where $FWHM_N$ is the final FWHM, and $FWHM_O$ is the original one. Like this, it is possible to automatically degrade the theoretical spectra when (and IF) the resolution is higher compared to that of the observed spectra. In a similar fashion, it is possible to degrade the observed spectra (e.g. in case they are very high resolution, higher than that of the SSPs), so that they match the resolution of the theoretical models. In the `config.dat` file, the eleventh line is now an integer flag telling whether

(1) or not (0) the SSPs spectra should be smoothed. The following line will have to contain 3 numbers: an integer flag, telling whether the observed spectra should be smoothed (1) or not (0), a second number providing the value of their FWHM, and the third one is again a flag to allow writing the smoothed observed spectrum within the files containing the model spectra. Note that this number has to be provided in case any of the two smoothing processes is performed. If both the aforementioned flags are set to 0 there is no need to provide the spectral resolution of the observed spectra;

4. it is now possible to have a table, in output, containing the both the "observed" and absolute magnitudes calculated from the model spectra. The resulting table is contained in a file called `magnitude.dat`. The magnitudes can be both Johnson and AB. See Sect. 4.2 to learn how to use this option. A subroutine (`photmodel.f90`) has been added to perform this calculation;

5. it is now possible to deal also with very nearby objects, for which the redshift cannot be use as a mean to calculate their distance. In order for SINOPSIS to be able to deal with such objects a further row has been added to the configuration file, at the 15th line (one but last), containing the name of the file with the (redshift-independent) distances (e.g.: `mydistances.dat`) and the redshift value above which those distances should be used (e.g.: 0.01). A new subroutine (`distread.f90`) is added to read the appropriate distance from the aforementioned file;

6. corrected a mislabeling in the main output file (upper limit on the star formation rates and on the percentage of the stellar mass in the age bins where labelled as `_m`. Now they are `_M`;

7. now, when the "FF" approach for the star formation history prescription is adopted, in the main output file of the physical properties, each of the SFR for the single stellar populations is labelled;

8. added labels in the `.eqw` file for the uncertainties on EW and for the S/N measure;

9. added labels in the `magnitudes.dat` file for absolute magnitudes;

10. in the configuration file, from this version, choosing the SFH prescription is done through string flags, not integer anymore.

11.9 Version 1.5.1

This version includes:

1. More funny banners;

2. when the "`none`" option for the normalization of the spectra is chosen, now also a redshift has to be provided in the catalog input file. Also, when this option is chosen, the output catalog is more consistent: aperture and total masses are now equal to 1, and the SFR values are such that they are consistent with this mass normalization;

3. corrected a bug because causing the distance to the galaxy to not be written in the output catalog;

11.10 Version 1.5.2

This version includes:

1. Thanks to Steve Bickerton, the code is now much faster. The changes are mainly in the `extinflux` routine which now performs operations only when needed.

11.11 Version 1.5.3

This version includes:

1. changed the format of the χ^2 value on the final, general, table. Now expressed with exponential notation (now is `es10.4`; formerly was `f10.4`);

2. the output file containing the catalog with the magnitudes calculated by the model is now named from the catalog input file, to which the `.mag` suffix is added;

3. a '#' character is added to the header of the `.eqw` file, so that the file is now "topcat" ready;

4. when using a physical normalization, if an object has a redshift ≤ 0, and the file with optional distances (`mydistances.dat`) is not used, then the object is skipped and not analyzed and it does not appear in any of the results. This prevents SINOPSIS from crashing;

11.12 Version 1.6.0

This version includes:

1. SINOPSIS can now deal with spectra in fits format. Both single files, 2-D fits file (1 spectrum per row), and cubes (such as for integral field data) are allowed as input data;

2. there is now a completely new setup file which is more easy to read. The mandatory name for this setup file is `config.sin`;

3. the new setup file is fairly free in format and does not need to have all the keywords specified. To this aim, a new subroutine, `set_defaults.f90`, has been written, which initialise all the parameters to some default (not necessarily the best, though) ones;

4. the tarfile of the code includes now an `example/` directory where some examples of catalog files, and the configuration file as well, can be found. These correspond to different kind of input catalogs;

5. the code is now compatible with a new set of SSP models (Charlot & Bruzual, 2016, Private Communication), which will soon be publicly available (and hence included in SINOPSIS);

6. a new option allows the user to chose to write, within the model spectra files (the `.spec` ones), a third column containing the model spectrum without emission lines;

7. an option is introduced to prevent SINOPSIS writing all of the best models parameters for all runs and metallicity values. The use of this option (which can be activated by setting to `no` the "`Output all the best fits for all runs and metallicities`" input field) is particularly desirable for large number of spectra. In case this option is used, only the details of the best–fit, reference model will be written out (in an `.out` file, as before);

8. as a complement to the option described above, the user can now choose not to write these `.out` file as well (by setting to `no` the "`Write output file for each reference model`" input field). Particularly useful if there are several thousands of spectra to analyze (e.g. as in the case of MUSE data), to limit the amount of files written in output;

9. when using the option for analyzing datacubes, the results will be outputted as cube as well, in fits format.

11.13 Version 1.6.1

This version includes:

1. SINOPSIS is now available with a MIT License Copyright (c) (thanks to Marco Riello for the suggestion!). The details are included both in the source code directory and in this manual;

2. in case the `cube` option is used, it is now possible to choose a "memory saving" option, which will very marginally slow down the speed of the code. In case this option is NOT used, SINOPSIS could use up to about 7 Gb of RAM, in case very large cube are used (e.g. such as for MUSE data);

3. a bug is fixed which enormously slowed down the code, or even gave a segmentation fault problem, in case an observed spectrum had the flux in one or more band equal or less than zero;

4. a new directory, `macro/` is now included in the code tarfile. Here you can find some python scripts to plot the results of the fits.

11.14 Version 1.6.2

This version includes:

1. a bug is corrected which prevented to use user–defined values of the distances, read-in from a catalog, instead of calculating them from the redshift. This is particularly crucial for nearby galaxies;

2. the `config.sin` file found in the `example` directory has been changed to include the keyword which enables reading the redshift value below which z–independent distances are used for a given galaxy;

3. somehow more detailed instructions to install the `cfitiso` libraries in a linux machine are included;

4. a new subroutine, `checkhome.f90` is added, to check whether SINOPSIS paths coincides which the one which is declared in the module `user.f90` (like this SINOPSIS can basically check whether the user changed the default path to the correct one);

5. a further keyword is introduced in the `config.sin` file, which allows the code to skip a user–specified number of lines when the spectra are given in ascii format, and when there is no header identifier character;

6. it is possible now to use a RAM–saving option (only available, and needed, for the datacube option). By choosing to do so, SINOPSIS does not keep the results for each spectrum in the memory, but writes them in a hidden file, which is at the end read and converted into a `fits` file;

7. some clean up and checks were performed.

11.15 Version 1.6.3

This version includes:

1. more bug fixes and stability check were performed;

2. two new optional outputs are introduced, currently working with the datacube option only. a) a "fitting flag" file in `.fits` format, with a similar format as the original datacube (i.e. having the same value for the `NAXIS1` and `NAXIS2` keywords), that has for each pixel value of 1, when the corresponding spectrum is fitted, 0, when it is not fitted as the redshift is missing, and -1 when it is not fitted as 40% or more of the flux values are below or equal to 0. b) a file containing the contribution to the total flux as a function of the age (the four main bins are considered for this). It is a `.fits` file and has the same format as the previously described file;

3. it is now possible to easily define a new set of continuum bands, by just creating a file containing them, and specifying its name on the configuration file;

4. when the datacube option is chosen, it is possible to use two redshift masks, one derived from the stellar velocities (absorption lines), and the other from the gas velocities (emission lines). To enable this option, just write the names of the files containing the two masks one after the other, in the `listfile`. In this way, the code will use the emission–lines redshift to measure the equivalent width of emission lines;

5. a fourth normalization band is introduced, centered at $\sim$ 6900 Å, restframe. This adds to the three previously defined bands. The decision of which of the fourth to use to normalize the model spectrum, is based on the observed spectral range, with priority given to the reddest bands.

6. a further option is added to the `config.sin` file which allows to use a fixed–width band to measure the Hβ line. See the `config.sin` file definition in Sect. 4.2;

7. as an "under the hood" feature, it is now possible, when using spectra from a datacube, to perform the fits on a rectangular subsection of the whole dataset. To do so, the coordinates, in pixel, of the lower and upper corner of the defined box, should be given when launching SINOPSIS as in the following example:
   `prompt> sinopsis 120 140 180 160`
   where $x_1 = 120$, $y_1 = 140$, $x_2 = 180$ and, $y_2 = 160$, and x and y are the coordinates of the lower–left (1) and upper–right (2) corner of the selected box, respectively. This will produce maps and outputs having the very same size of the original data, but the fits will be performed only on spectra in the selected area.

11.16 Version 1.6.4

This version includes:

1. fixed a bug which would prevent the use of analytical SFH;

2. fixed a bug which prevented to use the option of smoothing both observed a model spectra to a common resolution;

3. fixed a bug in the calculation of the times left in the SINOPSIS run.

11.17 Version 1.6.5

This version includes:

1. from this version on, customizable options which formerly required changing files in the `SINOPSIS/data/` directory, will have to be placed in the directory where the fits are performed. This modification has been pursued in order to avoid confusion. The customizable files recognized in this version are:

   - the file containing the cosmological parameters;
   - the file containing the definition of the continuum bands;
   - the file containing the parameters of the stellar populations;

   If the default files are used, then the corresponding keywords can be omitted, or it will have to be set to `default`. The `config.sin` file has been updated correspondingly.

2. the option of choosing where to read the continuum bands definition is now gone from the `config.sin`, as it was useless;

3. it is now possible to choose at which wavelength the normalization of observed and model spectrum is performed in the fitting. Two options are available: a "default" set of values (the code chooses the reddest one available in the observed spectrum), or a custom option. If the custom wavelength is out of range, SINOPSIS chooses the reddest on from the pre-defined list.

4. it is now possible to choose the age of the oldest SSP used by the code. Three different options are given:

   - the user defines a maximum age for all the spectra of the sample under analysis;
   - the age is chosen as the oldest age possible based on the galaxy's redshift (`zage` keyword in the `config.sin` file);
   - the age is chosen based on the galaxy's redshift and on a value of the galaxy formation redshift, the latter being provided by the user.

   Note that the distinction is automatically made by the code between the first and the last case, based on the value that is provided. Once the maximum age is chosen (or calculated), the maximum SSP age will be smaller or equal to this value.

5. SINOPSIS now can write, if chosen to do so, a file containing the spectra of the SSP at each age that are used to compose the final best fit model (i.e.: the model spectra weighted by their SFR value and with dust extinction applied). The files have the extension `.ssp` and are composed by a number of columns equal to the number of SSP spectra plus the wavelengths column (the first one);

6. it is now possible to account for a possible contribution from the [NII] lines to Hα, when the latter is measured in emission. This is useful when the spectral resolution is such that the three lines are blended, and hence their equivalent width is measured as if it was a single line. The number to be used here is meant to be the percentage of flux due to the nitrogen lines (this value needs to be between 0 and 1).

11.18 Version 1.6.6

This version includes:

1. We introduce from this version the option of measuring the equivalent width of all the spectral lines in the optical spectrum, using a fixed bandwidth to define the continuum level. The parameters are defined in the file `fix_lines_param.dat` in the `data/` directory. In this same file, the meaning of each parameter is explained. The `config.sin` file needs to be updated accordingly, and the default option will be **not** to use a fixed band, but performing the equivalent width trend curve analysis (see Fritz et al., 2007, 2014, for details);

2. related to the above point, it is possible to specify, in the `config.sin`, the name of a file with custom-defined parameters for the lines' measurement at fixed band-width. The file will have to be located in the directory where SINOPSIS is running. The `default` option will use the default file, `fix_lines_param.dat`, located in the `data/` directory;

3. from this version, the χ^2 value that is now written in the final output files will be the one of the reference model (i.e. the model, for the metallicity value having the minimum χ^2, that has the median stellar mass from the n models that have been run to calculate uncertainties);

4. a bug is fixed causing the extinction curve to be wrongly interpolated. Note that this does not affect results obtained with previous versions of the code, it only makes the code internally self-consistent;

5. fixed a bug in the equivalent width measurement when using a datacube;

6. changing the order of some parameters read from the `config.sin` file, in order to keep consistency. The `config.sin` file (given in the `example/` directory is also changed (the order of the keywords in the config file is not important any way);

7. completed the parameters that are set as default (`set_default subroutine`).

11.19 Version 1.6.7

This version includes:

1. Added the possibility, when fitting data from a IFU cube, to use a mask to specify which spaxels will be fitted or not, This mask is given as an argument to the `sinopsis` command when launching the code. Must be a fits file, of the same dimension of the spectra datacube, of integer format. Pixels with values larger than 0 in this mask will be fitted, others will be skipped;

2. introduced the possibility of using the EW of a spectral line which is out of the observed range, but for which a measurement is available. The `config.sin` file now contains one more line used to specify that this option is used. Note: this option is currently only available when spectra are given as single files, plus it will automatically force the use of fixed bands for the EW measurements;

3. some changes have been introduced (by Dani Dıáz) in the config procedure and in the makefiles, to facilitate the installation;

4. when using the non-parametric sfr option, the values of the $E(B-V)$ are now also written in the main output file (or in the output cube). These are located after the sfr values as las planes;

5. the definition of the main age bins to which the SFH is believed to be reliable (also called "main age bins") is now more easy to change. In the `data/` directory there is now a file in which these age bins are defined: `ssp_final_bins_jm.dat` or `ssp_final_bins_cb20.dat` (depending on which set of SSP is used).

11.20 Version 1.6.8

This version includes:

1. It is now possible to specify the file containing the parameters used to measure the equivalent widths of line not only in the "fixed' but also in the "variable" method. A line has been consequently added to the `config.sin` file.

References

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

Bertelli, G., Bressan, A., Chiosi, C., Fagotto, F., & Nasi, E. 1994, A&A Supl., 106, 275

Bressan, A., Granato, G. L., & Silva, L. 1998, A&A, 332, 135

Bruzual A., G. 1983, ApJ, 273, 105

Bruzual G., Charlot S., 2003, MNRAS, 344, 1000

Calzetti, D., Kinney, A. L., & Storchi-Bergmann, T. 1994, ApJ, 429, 582

Cardelli, J. A., Clayton, G. C., & Mathis, J. S. 1989, ApJ, 345, 245

Cava, A., Bettoni, D., Poggianti, B. M., et al. 2009, A&A, 495, 707

Colina, L., Bohlin, R. C., & Castelli, F. 1996, AJ, 112, 307

Ferland, G. J. 1996, Hazy, a Brief Introduction to CLOUDY, in University of Kentucky, Department of Physics and Astronomy Internal Report

Fritz, J., Poggianti, B. M., Bettoni, D., et al. 2007, A&A, 470, 137

Fritz, J., Poggianti, B. M., Cava, A., et al. 2011, A&A, 526, A45

Fritz, J., Poggianti, B. M., Cava, A., et al. 2014, arXiv:1402.4131

Gladders, M. D., Oemler, A., Dressler, A., et al. 2013, ApJ, 770, 64

Jacoby G.H., Hunter D.A., Christian C.A. 1984, ApJS, 56, 257

Liu, G., Calzetti, D., Hong, S., et al. 2013, ApJ Letters, 778, L41

Osterbrock, D. E. 1989, Research supported by the University of California, John Simon Guggenheim Memorial Foundation, University of Minnesota, et al. Mill Valley, CA, University Science Books, 1989, 422 p.

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