

---

# **speedyfit**

***Release 0.0.2***

**Apr 19, 2023**



<b>1</b>	<b>Installation</b>	<b>3</b>
<b>2</b>	<b>Quickstart</b>	<b>5</b>
2.1	Setup and searching for photometry . . . . .	5
2.2	Fitting the SED . . . . .	6
2.3	Constraints . . . . .	10
<b>3</b>	<b>Command line options</b>	<b>15</b>
3.1	speedyfit setup . . . . .	15
3.2	speedyfit phot . . . . .	16
3.3	speedyfit fit . . . . .	16
3.4	speedyfit checkgrids . . . . .	17
3.5	speedyfit copy_catalogs . . . . .	17
<b>4</b>	<b>The setup file</b>	<b>19</b>
4.1	Photometry setup . . . . .	20
4.2	Fit and constraints setup . . . . .	21
4.3	MCMC setup . . . . .	22
4.4	Output and figures . . . . .	22
<b>5</b>	<b>Photometry catalogs</b>	<b>25</b>
5.1	GALEX GR6+7 (UV) . . . . .	25
5.2	Gaia EDR3 (Optical) . . . . .	25
5.3	Gaia DR2 (Optical) . . . . .	26
5.4	Skymapper DR1 (Optical) . . . . .	26
5.5	APASS (Optical) . . . . .	26
5.6	SDSS (Optical) . . . . .	27
5.7	Stroemgren-Crawford (Optical) . . . . .	27
5.8	2MASS (IR) . . . . .	27
5.9	WISE (IR) . . . . .	28
<b>6</b>	<b>Adding Photometry catalogs</b>	<b>29</b>
6.1	Vizier catalogs . . . . .	29
6.2	TAP catalogs . . . . .	30
<b>7</b>	<b>The photometry file</b>	<b>31</b>

<b>8</b>	<b>Model atmosphere grids</b>	<b>35</b>
8.1	Kurucz . . . . .	36
8.2	Munari . . . . .	37
8.3	TMAP . . . . .	38
8.4	Black body . . . . .	38
8.5	Model comparison . . . . .	39
8.6	Adding model grids . . . . .	40
<b>9</b>	<b>Making figures</b>	<b>41</b>
9.1	SED fit . . . . .	41
9.2	Constraint plot . . . . .	42
9.3	Distribution plot . . . . .	43
<b>10</b>	<b>Indices and tables</b>	<b>45</b>
	<b>Index</b>	<b>47</b>

Speedyfit is a python package to fit the photometric spectral energy distribution of stars. It uses Bayesian inference to correctly take known parameters with errorbars into account as priors, and uses a Markov chain Monte Carlo approach to fit the requested parameters and determine their final errors.

It is a command line tool written in Python 3 that allows you to search the most common online databases for photometric observations of your target, and fit theoretical atmosphere models to the obtained photometry. Speedyfit can deal with both single and binary stars, and allows for the inclusion of constraints from other sources, as for example the distance or reddening but equally so constraints from spectroscopic observations.



# CHAPTER 1

---

## Installation

---

The installation of speedyfit requires two steps, installing the python package, and downloading the required atmosphere models. The speedyfit package can be installed with pip from the pypi repository as follows:

```
pip install speedyfit
```

If you want install the current development version directly from github you can also use pip:

```
pip install git+https://github.com/vosjo/speedyfit.git#egg=speedyfit
```

The atmosphere models that speedyfit uses to fit the SEDs can be downloaded from:

<https://drive.google.com/file/d/1yKHB3uB7QUyhhkwa1hNMuWazxjLkLAjN/view?usp=sharing>

Download them and unpack them in a directory of your choice. The last step is to store the path to the atmosphere models in an environment variable so that Speedyfit will know where to get them. In a bash shell this is done as follows:

```
export SPEEDYFIT_MODELS="<path to extracted atmosphere models>"
```

Where the path could be something like: '/home/user/speedyfit/modelgrids/'. To not always have to type this in the terminal when running Speedyfit, you can add it to your '.bashrc' file.

To check that speedyfit can find all models run:

```
speedyfit checkgrids
```

Which if everything went well should give you the following output:

```
Checking which atmosphere models are available...
Checking for models in <path to extracted atmosphere models>
kurucz2
    raw: available
    integrated: available
munari
    raw: available
```

(continues on next page)

(continued from previous page)

```
        integrated: available
tmap
    raw: available
    integrated: available
blackbody
    raw: available
    integrated: available
```

If you get an error like:

```
SPEEDYFIT_MODELS environmental variable not set. CAN NOT find models!
Please point the SPEEDYFIT_MODELS variable to the directory where you stored models.
On bash use:
export SPEEDYFIT_MODELS='<path to extracted atmosphere models>'
```

Check that your “SPEEDYFIT\_MODELS” variable is correctly set up. If you get “NOT FOUND” instead of “available” for any of the models, check that the “SPEEDYFIT\_MODELS” variable points to the correct directory, and that the directory contains the extracted atmosphere models.

To uninstall Speedyfit, run:

```
pip uninstall speedyfit
```



To get you started immediately, here is a description of how to fit the SED of the red giant [HIP 4618](#). This system is in fact a red giant binary, but the companion is a brown dwarf and the SED is completely dominated by the primary.

## 2.1 Setup and searching for photometry

First we need to create a setup file containing all settings necessary for the fitting process, and we need to obtain photometry from the literature.

Speedyfit can automatically download photometry from several of the large surveys, and some smaller studies available on Vizier. By default only photometry from the following large and well calibrated surveys is obtained:

- Gaia DR2
- Sky Mapper DR1
- APASS DR9
- SDSS DR9
- Stromgren compilation catalog of Paunzen
- 2MASS
- WISE

You can create a default setup file and download photometry for a given object as follows:

```
speedyfit setup <object_name> -grid kurucz --phot
```

This will create a setup file called `HIP_4618_setup_kurucz.yaml` where you can setup the fitting parameters and a photometry file called `HIP_4618.phot` with the obtained photometry. The object name should be resolvable by simbad, or if that is not possible, a J-type coordinate. For example, to obtain photometry of the star ‘HIP 4618’ you could also have provided the coordinate as: ‘J005919.3-582417.5’

The ‘-grid’ option lets you determine which atmosphere model will be used for the fit. You can also change this by hand in the setup file later. Here we picked the ‘kurucz’ grid as it works well for giants.

The photometry file contains the following measurements:

band	flux	eflux
GAIA2.G	2.5868394603952325e-12	4.765134383579859e-16
GAIA2.BP	2.46735399163153e-12	3.636027212972875e-15
GAIA2.RP	2.55071491325932e-12	3.289013356019166e-15
APASS.B	1.7988585256582425e-12	1.7988585256582425e-13
APASS.V	2.942094782202632e-12	2.942094782202632e-13
APASS.G	2.336641341423516e-12	2.336641341423516e-13
APASS.R	3.0275544488430818e-12	3.027554448843082e-13
APASS.I	2.6320052051220464e-12	2.632005205122046e-13
2MASS.J	1.3478059822453636e-12	2.3586127760129225e-14
2MASS.H	7.890960003699894e-13	2.9071370348494806e-14
2MASS.KS	3.5720112040912657e-13	8.882848761004847e-15
WISE.W1	6.922838260043923e-14	4.399556963130625e-15
WISE.W2	2.357855173591615e-14	6.080661822738957e-16
WISE.W3	5.545248458540115e-16	7.661043691311934e-18
WISE.W4	4.532380303499413e-17	1.3358325527436886e-18

Appart from the columns given above, which are used by speedyfit, also the catalog measurment and error, unit, distance to the target coordinates and the bibcode of the source are given.

## 2.2 Fitting the SED

The setup command created a default setup file where you can define the parameters of your fit. The default setup file for a single star fit with the kurucz grid looks as follows:

```
# photometry file with index to the columns containing the photbands, observations_
↳and errors
objectname: HIP_4618
photometryfile: HIP_4618.phot
photband_index: band
obs_index: flux
err_index: eflux
photband_exclude: ['GALEX', 'SDSS', 'WISE.W3', 'WISE.W4']
# parameters to fit and the limits on them in same order as parameters
pnames: [teff, logg, rad, ebv]
limits:
- [3500, 10000]
- [4.32, 4.32]
- [0.05, 2.5]
- [0, 0.1]
# constraints on distance and mass ratio is known
constraints:
  parallax: [7.3467, 0.0996]
# added constraints on derived properties as mass, luminosity, luminosity ratio
derived_limits: {}
# path to the model grids with integrated photometry
grids:
- kurucz
# setup for the MCMC algorithm
nwalkers: 100      # total number of walkers
nsteps: 1000      # steps taken by each walker (not including burn-in)
nrelax: 250       # burn-in steps taken by each walker
a: 10             # relative size of the steps taken
# set the percentiles for the error determination
```

(continues on next page)

(continued from previous page)

```

percentiles: [0.2, 50, 99.8] # 16 - 84 corresponds to 1 sigma
# output options
resultfile: HIP_4618_results_kurucz.csv # filepath to write results
plot1:
  type: sed_fit
  result: pc
  path: HIP_4618_sed_kurucz.png
plot2:
  type: distribution
  show_best: true
  path: HIP_4618_distribution_kurucz.png
  parameters: ['teff', 'rad', 'L', 'ebv', 'd', 'mass']

```

There are a few things we would like to change here. By default the logg is fixed, because fitting the surface gravity from an SED is rather difficult, and usually won't result in useful results. Lets try to vary it between  $\log g = 2.5 - 4.0$  anyway. For this we set the 2nd line of the limits parameters to [2.5, 4.0].

Since we know that we are dealing with a giant star on the lower part of the RGB, we have to change the range of the radius to  $R = 1 - 10 R_{\text{sol}}$ . So set the 3rd line of the limits parameters to [1.0, 10.0]

Lets also remove all constraints for now by setting constraints to an empty dictionary.

We are not interested in a csv output file, so remove the 'resultfile' parameter.

Finally in the figure part, change the parameters of the distribution plot to: 'parameters : ['teff', 'logg', 'rad', 'ebv', 'd']' so that we also get the logg output.

The final file then looks like:

```

# photometry file with index to the columns containing the photbands, observations_
↪and errors
objectname: HIP_4618
photometryfile: HIP_4618.phot
photband_index: band
obs_index: flux
err_index: eflux
photband_exclude: ['GALEX', 'SDSS', 'WISE.W3', 'WISE.W4']
# parameters to fit and the limits on them in same order as parameters
pnames: [teff, logg, rad, ebv]
limits:
- [3500, 10000]
- [2.50, 4.00]
- [1.0, 10]
- [0, 0.1]
# constraints on distance and mass ratio is known
constraints: {}
# added constraints on derived properties as mass, luminosity, luminosity ratio
derived_limits: {}
# path to the model grids with integrated photometry
grids:
- kurucz
# setup for the MCMC algorithm
nwalkers: 100 # total number of walkers
nsteps: 1000 # steps taken by each walker (not including burn-in)
nrelax: 250 # burn-in steps taken by each walker
a: 10 # relative size of the steps taken
# set the percentiles for the error determination
percentiles: [0.2, 50, 99.8] # 16 - 84 corresponds to 1 sigma

```

(continues on next page)

(continued from previous page)

```

plot1:
  type: sed_fit
  result: pc
  path: HIP_4618_sed_kurucz.png
plot2:
  type: distribution
  show_best: true
  path: HIP_4618_distribution_kurucz.png
  parameters: ['teff', 'logg', 'rad', 'ebv', 'd']

```

Now lets run the fit:

```
speedyfit fit HIP_4618_setup_kurucz.yaml
```

You will get the following output:

```

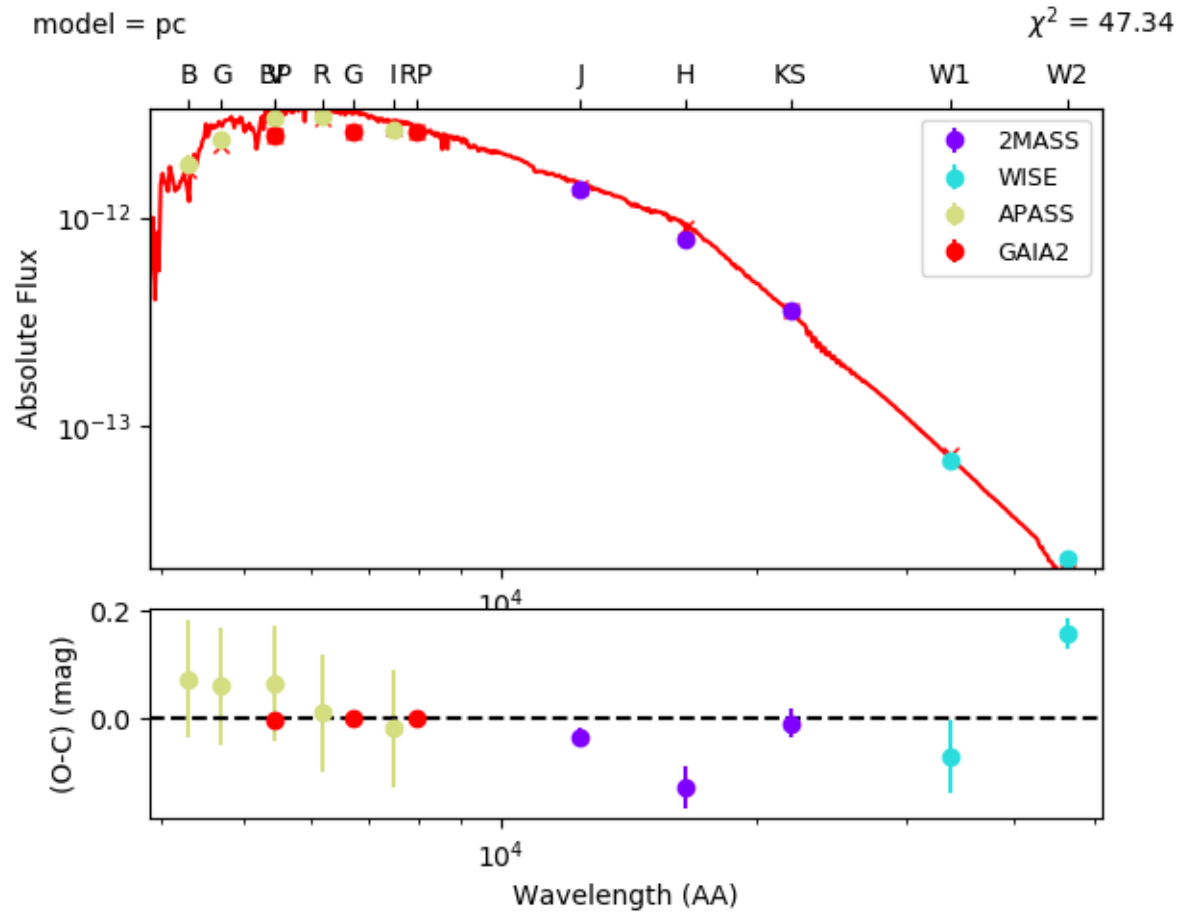
Applied constraints:
    distance = 136.11553486599425 - 1.845332907652827 + 1.845332907652827
100%|| 1250/1250 [00:51<00:00, 24.36it/s]
=====

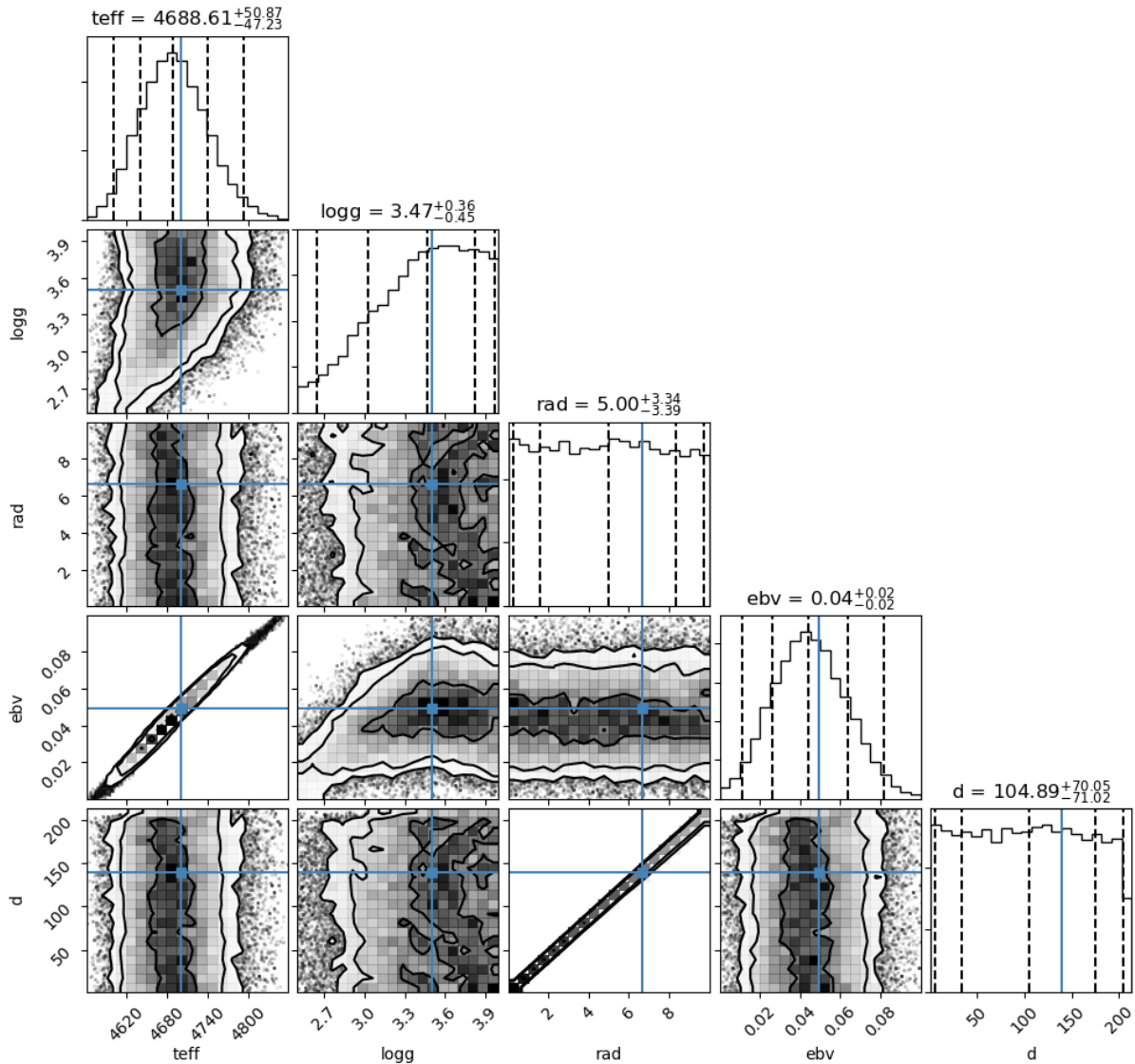
Resulting parameter values and errors:

```

Par	=	Best	Pc		emin		emax
teff	=	4699	4689	-	112	+	145
logg	=	3.49	3.48	-	0.97	+	0.51
rad	=	6.49	6.49	-	0.29	+	0.32
ebv	=	0.049	0.044	-	0.043	+	0.051
mass	=	4.72	4.67	-	4.15	+	10.81
d	=	136	136	-	5	+	6
L	=	18.38	18.26	-	1.72	+	2.19
scale	=	0.000	0.000	-	0.000	+	0.000
chi2	=	45.404	48.315	-	2.789	+	12.119

And the following two figures get created:





As you can see, only the effective temperature and reddening are determined with any precision. The other parameters, logg, radius and distance, are not constrained at all. The derived temperature is also well in line with that determined from spectroscopic observations:  $T_{\text{eff}} = 4750 \pm 100$  K [Jones et al. 2011 A&A, 536, A71](#).

## 2.3 Constraints

There is a spectroscopic solution of this system available, so we can use that to constrain the effective temperature and surface gravity. And there is a parallax measurement available from Gaia DR2 which was automatically added to the setup file, but we removed it before. Let's include it now. This is a good approach if you want to determine the radius and luminosity of the systems while also taking the errors on the spectroscopic parameters into account.

To do this we add the following lines to the constrained parameters in the setup file:

```
constraints:
  parallax: [7.3467, 0.0996]
```

(continues on next page)

(continued from previous page)

```
teff: [4750, 100]
logg: [2.91, 0.10]
```

We can also add the mass property to the output. Since we can derive the radius from the parallax and SED, and the surface gravity is included we can also determine the mass of the system.

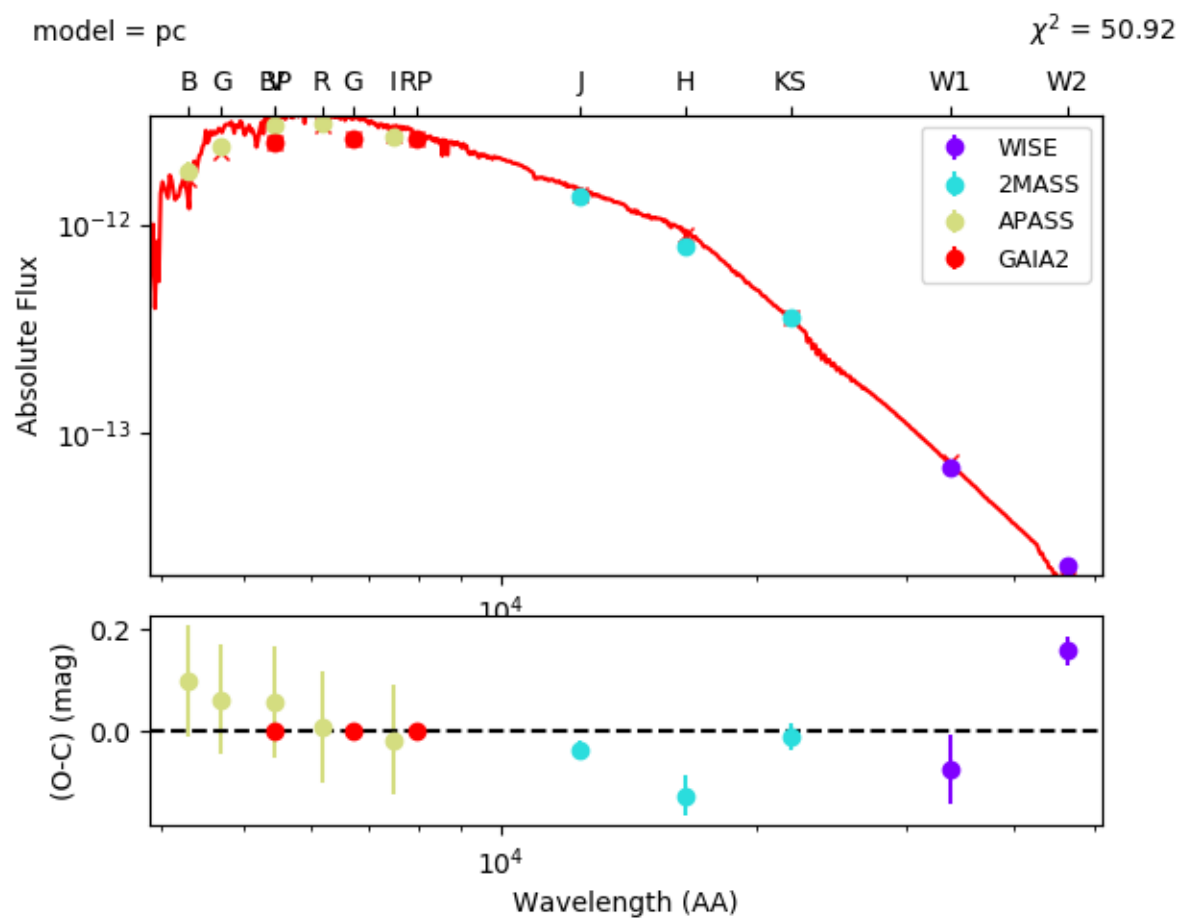
We can now run the fit again:

```
speedyfit fit HIP_4618_setup_kurucz.yaml
```

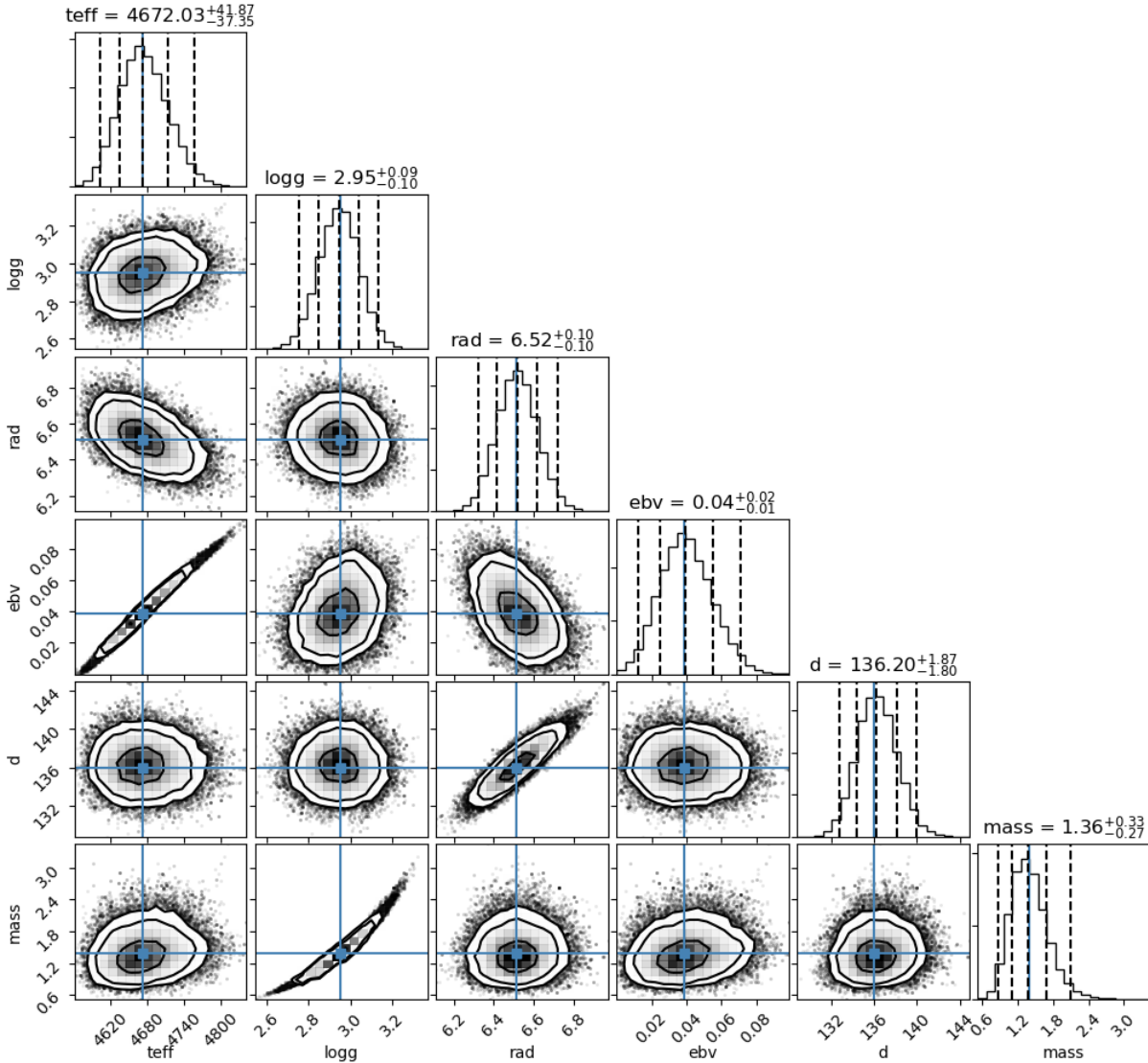
```
Applied constraints:
  teff = 4750 - 100 + 100
  logg = 2.91 - 0.1 + 0.1
  distance = 136.11553486599425 - 1.845332907652827 + 1.845332907652827
100%|| 1250/1250 [00:57<00:00, 21.66it/s]
=====

Resulting parameter values and errors:
```

Par		Best	Pc		emin		emax
teff	=	4671	4674	-	96	+	129
logg	=	2.94	2.95	-	0.28	+	0.27
rad	=	6.52	6.51	-	0.28	+	0.30
ebv	=	0.039	0.040	-	0.037	+	0.045
mass	=	1.36	1.36	-	0.65	+	1.19
d	=	136	136	-	5	+	6
L	=	18.12	18.13	-	1.55	+	1.93
scale	=	0.000	0.000	-	0.000	+	0.000
chi2	=	47.569	50.891	-	3.202	+	13.056







As you can see, the surface gravity is now constrained to more or less the same value as we provided for the prior. The same goes for the distance, as both these parameters can't really be constrained from an SED fit alone. The effective temperature however is more or less the same as in the unconstrained fit. This is because the SED required a much smaller distribution than we gave in the prior. If you look at the SED, you can see this is most likely caused by the Gaia photometry which has very small errors. The radius has very small errors now, as the Gaia parallax is very precise. The mass parameters (which is not fitted, but included as a derived parameter similar to luminosity) is now also quite well constrained.

If you noticed the difference in errors between the terminal output, and the distribution figure. This is caused because both use different parameters to set those. The terminal output (and is requested also the results file) use the percentiles parameter, while the distribution figure has different parameters to set the percentiles. Have a look at the [The setup file](#) and [Making figures](#) documentation for more info.



---

## Command line options

---

Speedyfit can run in three different modes:

- **setup**: create a setup file for an SED fit.
- **phot**: download photometry for an object.
- **fit**: fit a photometric SED based on a setup file.
- **checkgrids**: check which model atmosphere grids are installed.

Each mode has its own options, which are described below.

### 3.1 speedyfit setup

The ‘setup’ mode allows you to create a default setup file for the object that you want to fit. This setup file will contain the detailed setup for the SED fit including the parameters to fit, their limits, where to find the photometry, what model grids to use, the settings for the mcmc sampler and the output options. Speedyfit will create a default file based on a few arguments that you can then further edit to your liking. More details on the different possible options available in the setup file are given in *The setup file*.

**speedyfit** setup <object\_name> [options]

**object\_name**

The name of the object of which you want to fit the SED. This name needs to be resolvable by simbad, or it needs to be the coordinates in J<nnnnnn.n+-nnnnnn.n> format. As an example, consider the star Vega.

The following names for Vega (ra = 18:36:56.33635, dec = +38:47:01.2802) will be recognized:

- Vega
- any other of the 62 identifiers recognized by simbad: \* [alf Lyr: Identifiers](#)
- J183656.3+384701.2

**-grid** <model\_grid>

The name of the model grid that you want to use in the SED fit. This name is used to fill in the ‘grid’ parameter,

and the ‘limits’ parameter. By default the limits are set roughly at the limits of the grid. As some grids are not rectangular, this is not always possible.

If the option `-grid` binary is used, a binary setup file will be created with as default a hot component fitted with the TMAP grid, and a cold component fitted with the Kurucz grid.

#### **--phot**

If provided, speedyfit will also download photometry from Vizier and Tap archives. If you don’t specify this option, you can download the photometry in a separate step with ‘speedyfit phot’.

#### **--nopx**

If provided, speedyfit setup will NOT include the Gaia DR 2 parallax as a constraint in the fitting process. By default the parallax is always included.

## 3.2 speedyfit phot

The ‘photometry’ or ‘phot’ mode allows you do download photometry from several vizier and tap catalogs for the objects you are interested in. The default catalogs that Speedyfit will query can be found in [Photometry catalogs](#). More information on the formatting of the photometry file can be found in [The photometry file](#). This command only needs an object name to work.

**speedyfit phot** <object\_name> [options]

#### **object\_name**

The name of the object of which you want to download photometry. This name needs to be resolvable by simbad, or it needs to be the coordinates in J<nnnnnn.n+-nnnnnn.n> format. As an example, consider the star Vega.

The following names for Vega (ra = 18:36:56.33635, dec = +38:47:01.2802) will be recognized:

- Vega
- any other of the 62 identifiers recognized by simbad: \* [alf Lyr: Identifiers](#)
- J183656.3+384701.2

**-o** <output\_file>, **-output** <output\_file>

You can specify the name of the file in which speedyfit will store the downloaded photometry. If no name is given, the default name is: <object\_name>.phot

## 3.3 speedyfit fit

Finally, the ‘fit’ mode will use all information provided in the setup file together with the downloaded photometry to fit the SED and store and display the results. This command requires a setup yaml file, and requires you to have collected photometry of your target object and describe where it can be found in the setup file.

**speedyfit fit** <setup\_file> [options]

#### **setup\_file**

The name of the setup yaml file in which all parameters necessary for the SED fit process are specified. A default setup file can be created with the ‘speedyfit setup’ command.

#### **--noplot**

When provided, speedyfit fit will NOT display any of the plots, and will only save them to file. By default all plots requested in the setup file are shown when the fit is finished.

## 3.4 speedyfit checkgrids

The ‘checkgrids’ mode will check and print which model atmosphere grids are installed and can be used by speedyfit.

**speedyfit** checkgrids [*options*]

**--bands**

When provided, speedyfit will also list the photometric bands included in the integrated grids.

## 3.5 speedyfit copy\_catalogs

The ‘copy\_catalogs’ command will make a copy of the photometry catalog files in the SPEEDYFIT\_MODELS directory so you can edit them.

**speedyfit** copy\_catalogs

This command has no options. See *Adding Photometry catalogs* for more info.



## CHAPTER 4

---

### The setup file

---

All details regarding the setup for the SED fit are provided to Speedyfit using a setup file in yaml format. Yaml was chosen because it is easy to read by a human and by a computer. The default setup file with all its options are given below:

```
# photometry file with index to the columns containing the photbands, observations_
↳and errors
objectname: Vega
photometryfile: Vega.phot
photband_index: band
obs_index: flux
err_index: eflux
photband_include: ['GAIA', 'APASS', '2MASS']
photband_exclude: ['GALEX']
# parameters to fit and the limits on them in same order as parameters
pname: [teff, logg, rad, ebv]
limits:
- [3500, 10000]
- [4.31, 4.31]
- [0.01, 2.5]
- [0, 0.10]
# constraints on distance, mass ratio or any of the fitted parameters when known
constraints:
  parallax: [130.23, 0.36]
# added constraints on derived properties as mass, luminosity, luminosity ratio
derived_limits:
  mass: [1.5, 2.0]
# path to the model grids with integrated photometry
grids:
- kurucz
# setup for the MCMC algorithm
nwalkers: 100      # total number of walkers
nsteps: 1000      # steps taken by each walker (not including burn-in)
nrelax: 250       # burn-in steps taken by each walker
a: 10             # relative size of the steps taken
```

(continues on next page)

(continued from previous page)

```
# set the percentiles for the error determination
percentiles: [16, 50, 84] # 16 - 84 corresponds to 1 sigma
# output options
resultfile: Vega_results.csv # filepath to write results
plot1:
  type: sed_fit
  result: pc
  path: Vega_sed.png
plot2:
  type: distribution
  show_best: true
  path: Vega_distribution.png
parameters: ['teff', 'rad', 'L', 'ebv', 'd', 'mass']
```

## 4.1 Photometry setup

**objectname** (str)

Name of the system being fitted. Can be whatever you want, and doesn't need to be resolvable by Simbad. It is used for the figures and can help in recognizing what model this file belongs too, but doesn't serve any other purpose.

**photometryfile** (str)

The path to the file where the photometry of your object is stored.

**photband\_index** (str or int)

The name or index of the column in the photometry file that contains the names of the photometric bands. See [The photometry file](#) for more details.

**obs\_index** (str or int)

The name or index of the column in the photometry file that contains the flux. See [The photometry file](#) for more details.

**err\_index** (str or int)

The name or index of the column in the photometry file that contains the error on the flux. See [The photometry file](#) for more details.

**photband\_include** (list)

A list of all the photometric bands to include in the fit. Any band not mentioned in this list is excluded. Bands are named as: SYSTEM.BAND (e.g. GAIA2.G for the G band from Gaia DR2). When the system is used (e.g. GAIA2) all bands from that system are included. If a specific band (e.g. GAIA2.G) is used, only that band is included. When both [photband\\_exclude](#) and [photband\\_include](#) are included in the setup file, [photband\\_include](#) is processed first, after which bands to be excluded are removed.

**photband\_exclude** (list)

A list of which photometric bands to exclude from the fit. All bands mentioned here will be excluded. Bands are named as: SYSTEM.BAND (e.g. GAIA2.G for the G band from Gaia DR2). When the system is used (e.g. GAIA2) all bands from that system are excluded. If a specific band (e.g. GAIA2.G) is mentioned, only that band is excluded. When both [photband\\_exclude](#) and [photband\\_include](#) are included in the setup file, [photband\\_include](#) is processed first, after which bands to be excluded are removed.



## 4.2 Fit and constraints setup

### **pnames** (list)

A list of the parameters to include in the fit. If you are performing a single fit, this will be:

[teff,logg,rad,ebv]

If you are performing a binary fit this will be:

[teff,logg,rad,teff2,logg2,rad2,ebv]

Note that for a binary fit, the reddening parameter (ebv) will only be included once. Both components in the fit need to have the same reddening.

### **limits** (list of lists)

The limits on the parameters included in the fit. Limits have to be given in the same order as the parameters are given in *pnames*.

For example:

```
pnames = [teff,logg]
limits = [(3500,10000),(3.5,4.5)]
```

means that effective temperature is limited between 3500 and 10000 K, and logg is limited between 3.5 and 4.5 dex.

### **constraints** (dict)

A dictionary containing all the constraints that you want to apply in the fit. These are treated as priors in the Bayesian fit. You can apply a prior on the distance/parallax, any of the parameters you are fitting (teff,logg,rad,ebv), any of the derived parameters (mass and luminosity) and in case of a binary fit also the mass ratio.

A constraint is not the same as a limit. A constraint consists of a value and an error. You can provide a double sided error as well. e.g.

```
parallax: [130.23,0.36] -> parallax = 130.23 +- 0.36
teff: [5630,150,270] -> teff = 5630 -150 +270
```

Note: there constraints can be valuable if for example the effective temperature and surface gravity are known from a spectroscopic fit, and you want to derive the radius from the SED and the parallax while correctly propagating the errors.

### **derived\_limits** (dict)

A dictionary containing all the limits that you want to apply to derived parameters. These are also treated as priors, but as limiting priors were the prior is uniform between the lower and upper limit. You can apply limits on all derived parameters (mass and luminosity), on the distance and in case of a binary fit also on the mass ratio (q). e.g:

```
q: [0.5,0.9]
```

means that the mass ratio is limited between 0.5 and 0.9

### **grids** (list)

List of the model atmosphere grids used in the fit. For a single star fit only one grid name should be provided, for a binary fit, two grid names should be provided. You can provide the name of a grid as recognized by speedyfit, or the direct path to the grid. Recognized names are:

- kurucz
- munari
- tmap
- koester
- blackbody

For more info on the different available grids see *Model atmosphere grids*.

**percentiles** (list)

The percentiles to use when calculating the error on the fitted parameters. By default the percentiles are set to:  
[16, 50, 85]

The best fitting value is the median of all values, and the lower and upper error given by speedyfit will then correspond to 1 sigma. This means that 68% of all steps taken by the walkers are within the error given by Speedyfit. See the 68–95–99.7 rule for more information on this. Quick summary:

1 sigma: [16, 50, 85]  
2 sigma: [2.5, 50, 97.5]  
3 sigma: [0.15, 50, 99.85]

If you don't do any burn-in (via option *nrelax*), setting the percentiles to correspond with for example 3 sigma will result in very large unrealistic errors.

## 4.3 MCMC setup

The MCMC chain is implemented using the EMCEE package. See *emcee* for more information on the parameters governing the MCMC fitting process.

**nwalkers** (int)

Number of walkers used in the MCMC chain.

**nsteps** (int)

Number of steps that each walker takes in the MCMC chain.

**nrelax** (int)

Number of burn in steps that each walker takes. Walkers will start of at random locations in the parameter space and will slowly converge to an optimal solution. The nrelax steps are the first number of steps that will not be taken into account when determining the errors on the parameters from the walker chains.

**a** (int)

Relative step size taken by each walker.

## 4.4 Output and figures

**resultfile** (str)

The name of the file to store the results of the fit. This file will contain the best fitting value and error of each fitted and derived parameters in csv format.

**plot<n>** (dict)

In the result file you can also define what figures you want speedyfit to make. There are 3 types of figures that speedyfit can make:

- sed\_fit
- constraint
- distribution

In the setup file you can define up to 10 different configurations of those 3 figures. All figures will follow the same structure:

```
plot1: # number of the plot, up to 10
  type: <plot type> # which type of plot you want
  path: <figurename>.png # filename in which to store the figure
  parameter: value # any other parameters to configure this plot
```

For more information on available plots and their configuration see [Making figures](#)



---

## Photometry catalogs

---

Speedyfit can automatically obtain photometry from several large online databases, using both Vizier and TAP queries. It will use simbad to find the coordinates for your system, and will do a search based on those coordinates. The included catalogs are given below:

### 5.1 GALEX GR6+7 (UV)

Revised catalog of GALEX UV sources: <https://galex.stsci.edu/GR6/> Both GALEX DR5 and the revised GALEX DR6-7 are queried. Included photometric bands are:

- GALEX.FUV
- GALEX.NUV

This catalog is queried through Vizier: [II/312/ais](#) (DR5), [II/335/galex\\_ais](#) (DR6-7)

Cite as: [Bianchi et al. 2017, ApJS](#)

### 5.2 Gaia EDR3 (Optical)

The early third data release of the Gaia Spacecraft: <https://gea.esac.esa.int/archive/> Included photometric bands are:

- GAIA3E.G
- GAIA3E.BP
- GAIA3E.RP

This catalog is queried through Vizier: [I/350/gaiaedr3](#)

Cite as: [Gaia collaboration et al. 2020, A&A](#)

## 5.3 Gaia DR2 (Optical)

The second data release of the Gaia Spacecraft: <https://gea.esac.esa.int/archive/> Included photometric bands are:

- GAIA2.G
- GAIA2.BP
- GAIA2.RP

This catalog is queried through Vizier: [I/345/gaia2](https://vizier.cfa.harvard.edu/vizier/?source=I/345/gaia2)

Cite as: [Gaia collaboration et al. 2018, A&A](#)

## 5.4 Skymapper DR1 (Optical)

The first public data release of the Skymapper survey: <http://skymapper.anu.edu.au/> Included photometric bands are:

- SKYMAPPER.U
- SKYMAPPER.V
- SKYMAPPER.G
- SKYMAPPER.R
- SKYMAPPER.I
- SKYMAPPER.Z

This catalog is queried through TAP at: <http://api.skymapper.nci.org.au/public/tap/>

Cite as: [Wolf et al. 2018, PASA](#)

## 5.5 APASS (Optical)

The 9th data release of The AAVSO Photometric All-Sky Survey: <https://www.aavso.org/apass>. The APASS survey is valid from roughly 7th to 17th magnitude. Included photometric bands are:

- JOHNSON.B
- JOHNSON.V
- SLOAN.U (only available for very few stars)
- SLOAN.G
- SLOAN.R
- SLOAN.I
- SLOAN.Z

Even though these bands are supposed to be Johnson and Sloan, they differ slightly from the standard bands, and speedyfit uses specific APASS passbands.

This catalog is queried through Vizier: [II/336/apass9](https://vizier.cfa.harvard.edu/vizier/?source=II/336/apass9)

Cite as: [Henden et al. 2015, AAS](#)

---

**Note:** The 10th data release of APASS is public, but can not be automatically queried. You can include this data by hand by obtaining it from: <https://www.aavso.org/apass-dr10-download>

---

## 5.6 SDSS (Optical)

The 9th data release of the Sloan Digital Sky Survey: <http://www.sdss3.org/dr9/>. Included photometric bands are:

- SDSS.U
- SDSS.G
- SDSS.R
- SDSS.I
- SDSS.Z

This catalog is queried through Vizier: [V/139/sdss9](http://vizier.u-strasbg.fr/viz-bin/V/139/sdss9)

Cite as: [Ahn et al. 2012, ApJS](#)

## 5.7 Stroemgren-Crawford (Optical)

This is the new catalogue of Strömrgren-Crawford  $uvby\beta$  photometry created by Paunzen, E. by combining Stromgren photometry from all existing catalogs. Included photometric bands are:

- STROMGREN.Y
- STROMGREN.B-Y
- STROMGREN.M1
- STROMGREN.C1
- STROMGREN.HBN-HBW (Limited availability)

These measurements are automatically converted to Stromgren  $u$ ,  $v$ ,  $b$ ,  $y$  and  $\beta$ .

This catalog is queried through Vizier: [J/A+A/580/A23/catalog](http://vizier.u-strasbg.fr/viz-bin/J/A+A/580/A23/catalog)

Cite as: [Paunzen et al. 2015, A&A](#)

---

**Note:** As this is a compilation catalog, you should reference the original catalog that contains the measurements for the system that you are studying.

---

## 5.8 2MASS (IR)

The Two Micron All Sky Survey: <https://irsa.ipac.caltech.edu/Missions/2mass.html>. Included photometric bands are:

- 2MASS.J (Magnitude limit = 15.8)
- 2MASS.H (Magnitude limit = 15.1)
- 2MASS.KS (Magnitude limit = 14.3)

This catalog is queried through Vizier: [II/246/out](#)

Access to the images is obtained via isra: <https://irsa.ipac.caltech.edu/Missions/2mass.html>

Cite as: [Skrutskie et al. 2006, AJ](#)

## 5.9 WISE (IR)

The Wide-field Infrared Survey Explorer: <http://wise.ssl.berkeley.edu/>. This is the ALLWISE version of the WISE catalog. It is a combination of the data from the original WISE mission, extended by the NEOWISE mission which continued observing in band W1 and W2 after W3 and W4 ran out of coolant. Included photometric bands are:

- WISE.W1
- WISE.W2
- WISE.W3
- WISE.W4

This catalog is queried through Vizier: [II/328/allwise](#)

Access to the images is obtained via isra: <https://irsa.ipac.caltech.edu/applications/wise/>

Cite as: [Cutri et al. 2013, yCat](#)



---

## Adding Photometry catalogs

---

Speedyfit is by default configured with a few standard catalogs that cover many different uses. If this is not sufficient for you, you can activate other catalogs, or add your own. This is an easy process.

The catalog files are stored with the program data. Changing those is annoying, and an update would overwrite them. Therefore, the first step is to copy them to your user space:

```
speedyfit copy_catalogs
```

This will copy two photometry catalog files: `tap_cats_phot.cfg` and `vizier_cats_phot.cfg` to the `SPEEDYFIT_MODELS` directory where you can edit them. Speedyfit will first check the `SPEEDYFIT_MODELS` directory, and if it can't find any catalogs there, it will use the ones in its installation directory. So even when updating speedyfit, the catalogs in your `SPEEDYFIT_MODELS` will not be affected.

You can now activate any of the other catalogs in those files by uncommenting them. Or you can add your own catalogs.

### 6.1 VizieR catalogs

The vizier catalogs are stored in the file: `vizier_cats_phot.cfg`. Most catalogs that you are going to use will be available through [VIZIER](#). This is by far the easiest system to work with. Speedyfit uses the [astroquery](#) library to query vizier.

The syntax to use is straightforward, see e.g. the APASS catalog:

```
[II/336/apass9] # APASS all sky survey
Bmag = APASS.B
Vmag = APASS.V
g_mag = APASS.G
g_mag_unit = ABmag
r_mag = APASS.R
r_mag_unit = ABmag
i_mag = APASS.I
i_mag_unit = ABmag
bibcode = 2015AAS...22533616H
```

in square brackets is the vizier name of the catalog: [II/336/apass9], and then for each included band there is a mapping between the vizier name and the band name used by speedyfit e.g.: Bmag = APASS.B. You can also specify units that are used to convert magnitudes to fluxes if necessary e.g.: g\_mag\_unit = ABmag. A bibcode can be added for informational purposes: bibcode = 2015AAS...22533616H. For most catalogs you will only need to include the band mappings to have a functional system.

If the error keyword in the catalog is not the default, you can specify it with e.g.: <error\_key\_in\_catalog> = e\_<band\_name\_in\_catalog>. See e.g. the J/A+A/600/A50/sdcat catalog below for an example of that:

```
[J/A+A/600/A50/sdcat] # subdwarf catalog Geier+2017: galex for sdB stars
FUVmagc = GALEX.FUV
FUVmagc_unit = ABmag
FUVmagc_err = e_FUVmag
NUVmagc = GALEX.NUV
NUVmagc_unit = ABmag
NUVmagc_err = e_NUVmag
bibcode = 2017A&A...600A..50G
```

Here the error on the FUV band in the catalog is given by FUVmagc\_err, instead of the default e\_FUVmag.

## 6.2 TAP catalogs

There is one other way of obtaining photometry, which is through the tap protocol. The Table Access Protocol specified by the International Virtual Observatory Alliance: <http://www.ivoa.net/documents/TAP/>. It uses the Astronomical Data Query Language which is based on SQL. This is harder to work with as there are many more possibilities for querying data. Catalogs that use this format are stored in the tap\_cats\_phot.cfg file.

Currently there is only 1 catalog that uses the TAP protocol. You can try to add others, but there is no guarantee that it will work.

## CHAPTER 7

### The photometry file

Speedyfit can download photometry from Vizier and use Tap queries to obtain photometry from Skymapper. It saves the downloaded photometry to a text file using the `astropy.io.ascii.write` function using the ‘fixed\_width’ format.

A photometry file will look like:

	band	meas	emeas	unit	distance	bibcode	flux	
↪	eflux							↪
↪	SKYMAPPER.U	11.37	0.01	mag	0.209	2018PASA...35...10W	2.502e-13	2.
↪	996e-15							
↪	SKYMAPPER.V	11.34	0.00	mag	0.209	2018PASA...35...10W	2.110e-13	1.
↪	166e-15							
↪	SKYMAPPER.G	10.64	0.00	mag	0.209	2018PASA...35...10W	2.427e-13	6.
↪	708e-16							
↪	SKYMAPPER.R	10.32	0.01	mag	0.209	2018PASA...35...10W	2.203e-13	3.
↪	856e-15							
↪	SKYMAPPER.Z	10.04	0.01	mag	0.209	2018PASA...35...10W	1.258e-13	2.
↪	203e-15							
↪	GAIA2.G	10.29	0.00	mag	0.144	2018A&A...616A...1G	1.896e-13	6.
↪	987e-17							
↪	GAIA2.BP	10.65	0.00	mag	0.144	2018A&A...616A...1G	2.219e-13	2.
↪	658e-16							
↪	GAIA2.RP	9.77	0.00	mag	0.144	2018A&A...616A...1G	1.619e-13	1.
↪	193e-16							
↪	2MASS.J	9.00	0.02	mag	0.126	2003yCat.2246....0C	7.621e-14	1.
↪	895e-15							
↪	2MASS.H	8.52	0.04	mag	0.126	2003yCat.2246....0C	4.474e-14	1.
↪	895e-15							
↪	2MASS.KS	8.40	0.01	mag	0.126	2003yCat.2246....0C	1.857e-14	3.
↪	250e-16							

Where the columns have the following meaning:

- band: The photometric pass band of the measurement provided in SYSTEM.BAND format.
- meas: The measurement as how it is included in the catalog. Usually magnitudes, but can be fluxes as well.

- `emeas`: The error on the catalog measurement.
- `unit`: The unit of the catalog measurement.
- `distance`: The distance between the target and the catalog source in arcsec.
- `bibcode`: The bibcode of the article describing the source when available.
- `flux`: The flux of the measurement in `erg/s/cm2/AA`.
- `eflux`: The error on the flux in `erg/s/cm2/AA`.

The only columns that are important for the fit, and the only ones that speedyfit will actually read are:

- `band`
- `flux`
- `eflux`

The names of these columns are the default values. You can change them, if you also update the corresponding names in the setup file of the fit: `photband_index`, `obs_index`, `err_index`. See [The setup file](#)

You don't have to use photometry provided by speedyfit, you can use your own photometry as long as you have the photometric bands in a format that speedyfit can read, the flux and the errors. You can have any other number of columns in the photometry file, but they are not needed and are ignored by speedyfit.

---

**Note:** The flux and its error have to be given in units of : `erg/s/cm2/AA`! The fitting part of Speedyfit will not perform any unit conversions.

---

Speedyfit allows two possible formats for the photometry file. The 'fixed\_width' format described above, and a headerless format that is read by [astropy.io.ascii.write](#) with the default format value, and assuming no header (`data_start=0`, `header_start=None`). In the headerless format, the position of the photometry band, flux and error needs to be specified as an integer.

An example of a headerless photometry file is:

```
#      meas  e_meas  flag  unit  photband  source      cmeas      e_cmeas      cunit
↪      bibcode      comments
# float64 float64 |S20 |S30 |S30 |S50      float64      float64      |S50
↪      |S19      |S74
11.0835  0.0014 nan  mag  GAIA2.G  I/345/gaia2  9.37669e-14  9.37669e-16  erg/s/
↪cm2/AA  2018yCat.1345....0G -
11.0258  0.0055 nan  mag  GAIA2.BP  I/345/gaia2  1.61477e-13  1.61477e-15  erg/s/
↪cm2/AA  2018yCat.1345....0G -
11.0571  0.001 nan  mag  GAIA2.RP  I/345/gaia2  4.96983e-14  4.96983e-16  erg/s/
↪cm2/AA  2018yCat.1345....0G -
11.123   0.012 nan  mag  APASS.B  II/336/apass9  2.2233e-13  2.45728e-15  erg/s/
↪cm2/AA  2015AAS...22533616H -
11.17    0.009 nan  mag  APASS.V  II/336/apass9  1.23896e-13  1.23896e-15  erg/s/
↪cm2/AA  2015AAS...22533616H -
11.063   0.033 nan  ABmag  APASS.G  II/336/apass9  1.86108e-13  5.65658e-15  erg/s/
↪cm2/AA  2015AAS...22533616H -
11.251   0.013 nan  ABmag  APASS.R  II/336/apass9  8.73103e-14  1.0454e-15   erg/s/
↪cm2/AA  2015AAS...22533616H -
11.377   0.028 nan  ABmag  APASS.I  II/336/apass9  5.41327e-14  1.39603e-15  erg/s/
↪cm2/AA  2015AAS...22533616H -
```

Where the columns of interest are the `photband`, `cmeas` and `e_cmeas`. To use this file you would specify: `photband_index = 4`, `obs_index = 6`, `err_index = 7` in the setup file.

---

**Note:** Headerless does not necessarily mean without a header. As long as the header is commented it is ignored by speedyfit, and the file is considered headerless.

---



---

### Model atmosphere grids

---

Speedyfit is provided with 4 grids of model atmospheres. Below the range in effective temperature and surface gravity of all grids is shown. You can make these plots with the following python code:

```
import pylab as pl
from astropy.io import fits
from speedyfit.model import get_grid_file

grid = 'blackbody'

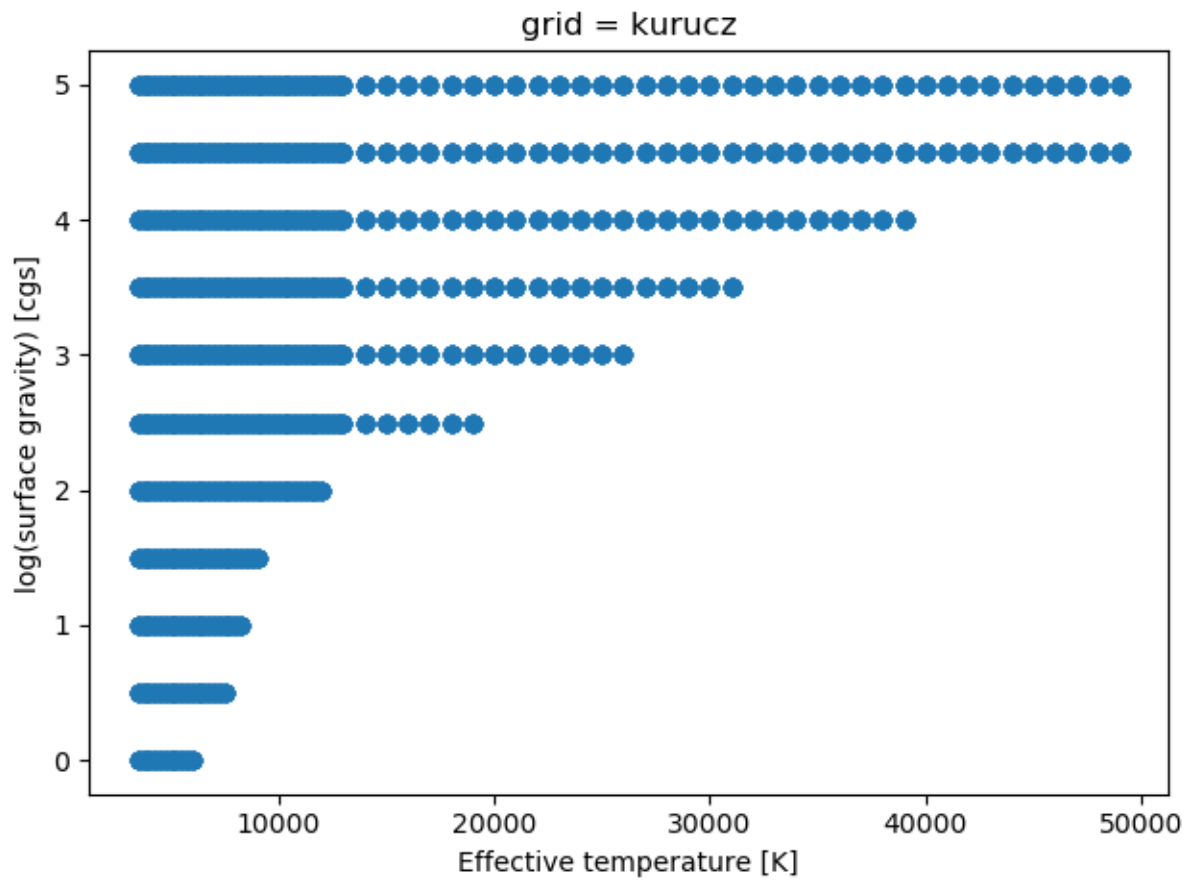
hdu = fits.open(get_grid_file(grid=grid, integrated=True))

teff = hdu[1].data['teff']
logg = hdu[1].data['logg']

pl.figure()
pl.plot(teff, logg, 'oC0')
pl.xlabel('Effective temperature [K]')
pl.ylabel('log(surface gravity) [cgs]')
pl.title('grid = ' + grid)
pl.tight_layout()

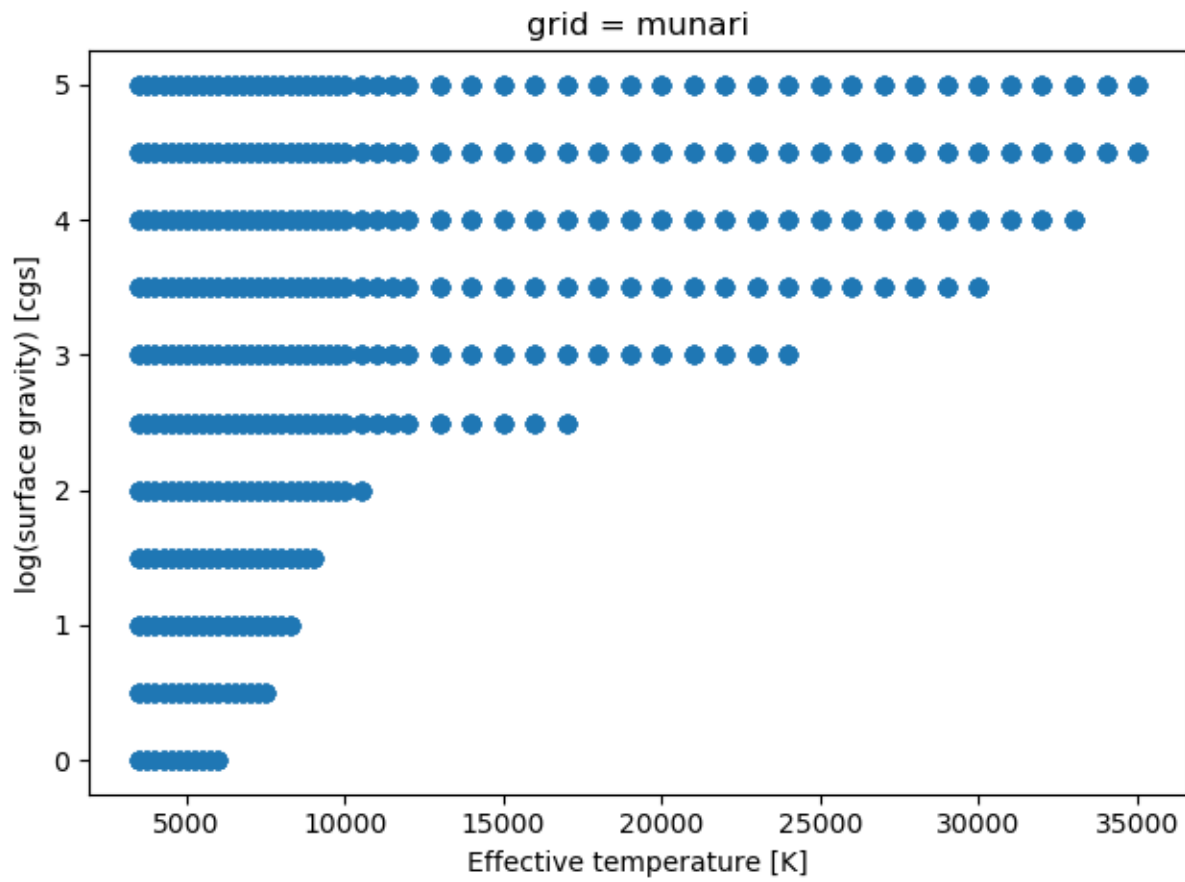
pl.show()
```

## 8.1 Kurucz

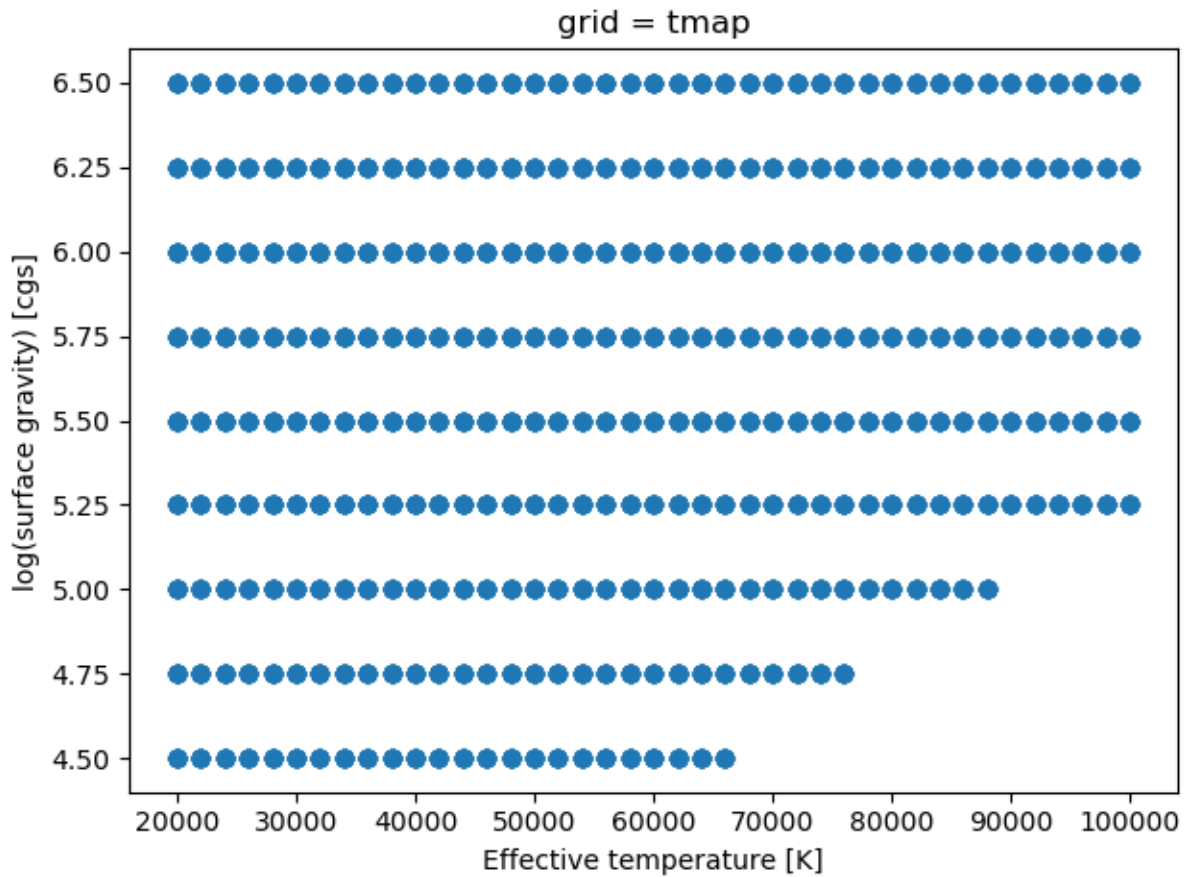




## 8.2 Munari

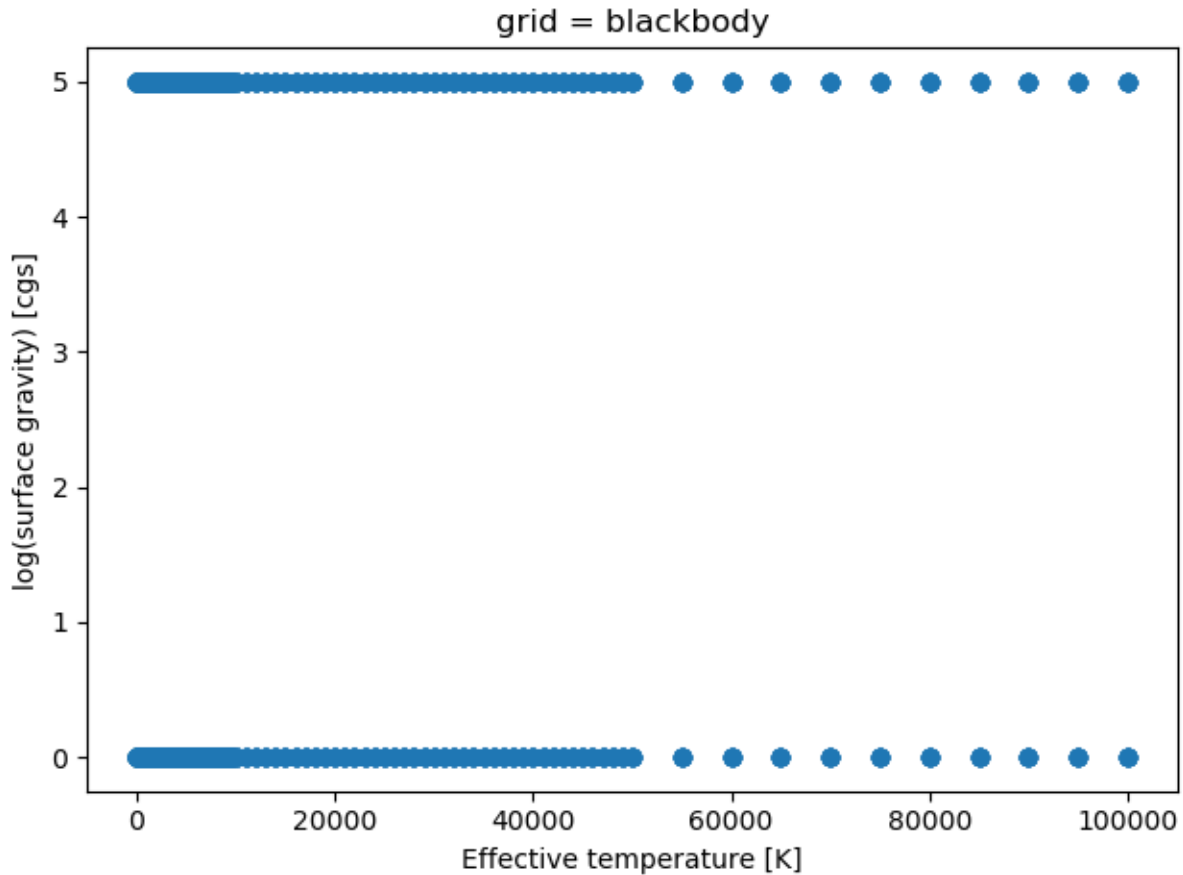


## 8.3 TMAP



## 8.4 Black body

A black body obviously doesn't have a surface gravity, but due to the way that the code is build, it is easier to create a pre integrated grid of black body model atmospheres at different temperatures and 2 surface gravities then to change the code and implement the black body analytically.



## 8.5 Model comparison

Lets compare a spectrum of each of these models at the same effective temperature and surface gravity:

```
import pylab as pl
import numpy as np
from speedyfit.model import get_table_single

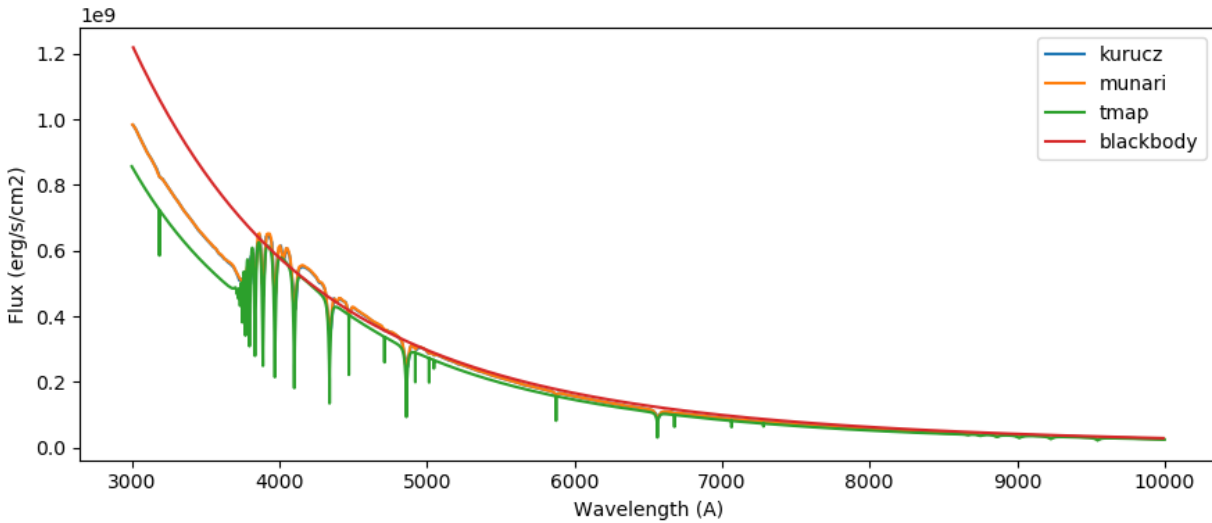
grids = ['kurucz', 'munari', 'tmap', 'blackbody']

pl.figure(figsize=(10,5))

for grid in grids:

    wave, flux = get_table_single(teff=20000,logg=5.0,ebv=0.0,grid=grid)
    s = np.where((wave > 3000) & (wave<10000))
    pl.plot(wave[s], flux[s], label=grid)

pl.xlabel('Wavelength (A)')
pl.ylabel('Flux (erg/s/cm2)')
pl.legend(loc='best')
pl.tight_layout()
pl.show()
```



## 8.6 Adding model grids

You can add your own model atmosphere grids to speedyfit. For this you need a fits file containing the model atmospheres, and a separate file containing the models integrated over the pass bands that you want to use. This grid also needs have the reddening pre calculated for the range you want to include in the fit.

The file name for the non integrated grid is not important. The filename of the integrated grid has to follow: <gridfile-name>\_lawfitpatrick2004\_Rv3.10.fits

Then add the grid to the file 'grid\_description.yaml' located in the same folder as the model grids: '\$SPEEDY-FIT\_MODELS'

The new grid should then show up when calling:

```
speedyfit checkgrids
```

---

## Making figures

---

Speedyfit gives you the option to make three different kinds of plots:

- sed fit
- constraint plot
- distribution plot

You can setup all of these plots and if wanted multiple of each in the setup file for the fit. They all follow the same basic structure, but have different option arguments:

```
plot1: # number of the plot, up to 10
      type: <plot type> # which type of plot you want
      path: <figurename>.png # filename in which to store the figure
      parameter: value # any other parameters to configure this plot
```

The different plotting types are discussed in detail below.

### 9.1 SED fit

Type keyword: sed\_fit

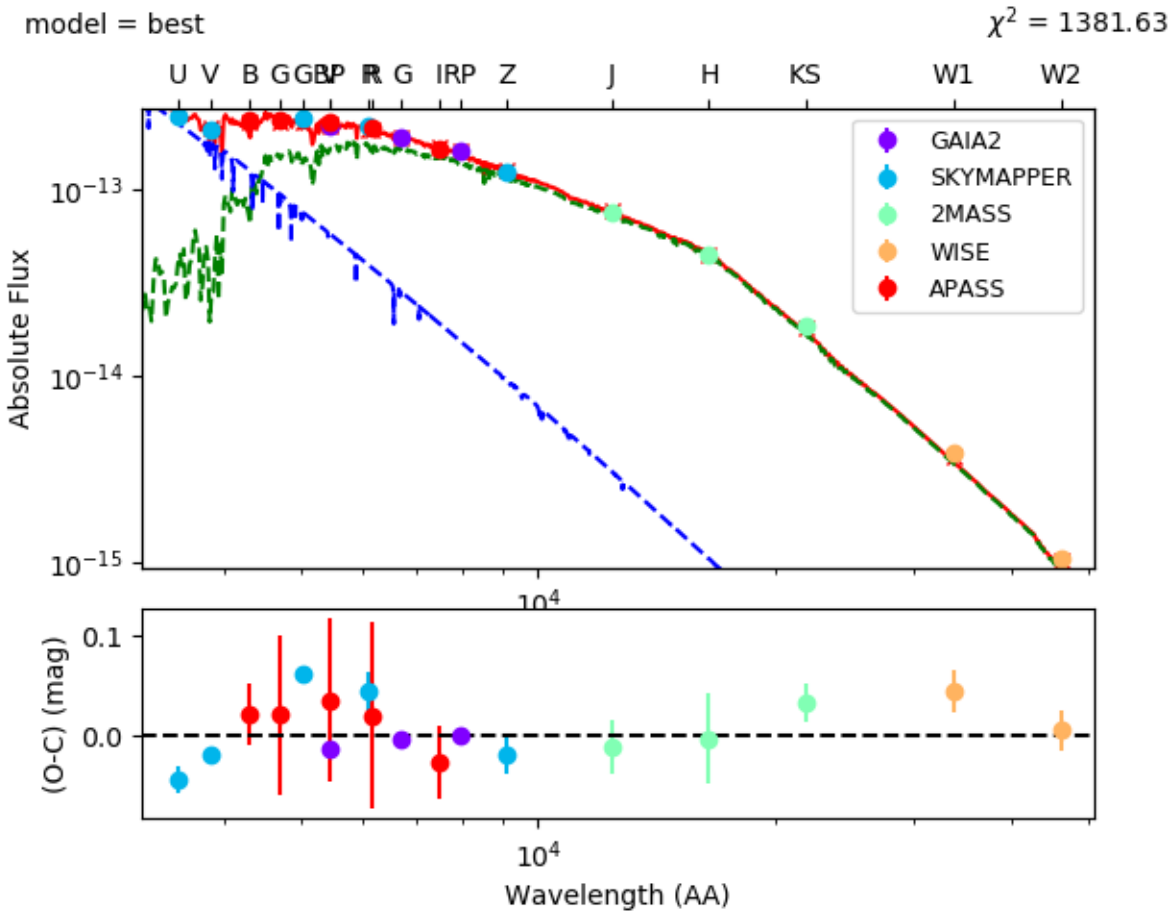
The SED fit, displays the used photometry, together with the integrated photometry of the best fitting model and the full atmosphere model for the best fitting parameters. If a binary fit is performed, the full atmosphere model of both components will be shown.

Optional arguments:

**result** ('best' or 'pc')

What model to consider the result. Pick the model with the lowest Chi2 value ('best') or the median model ('pc').

An example of this plot for the binary system BD-7 5977, an hot subdwarf binary with a sub giant companion:



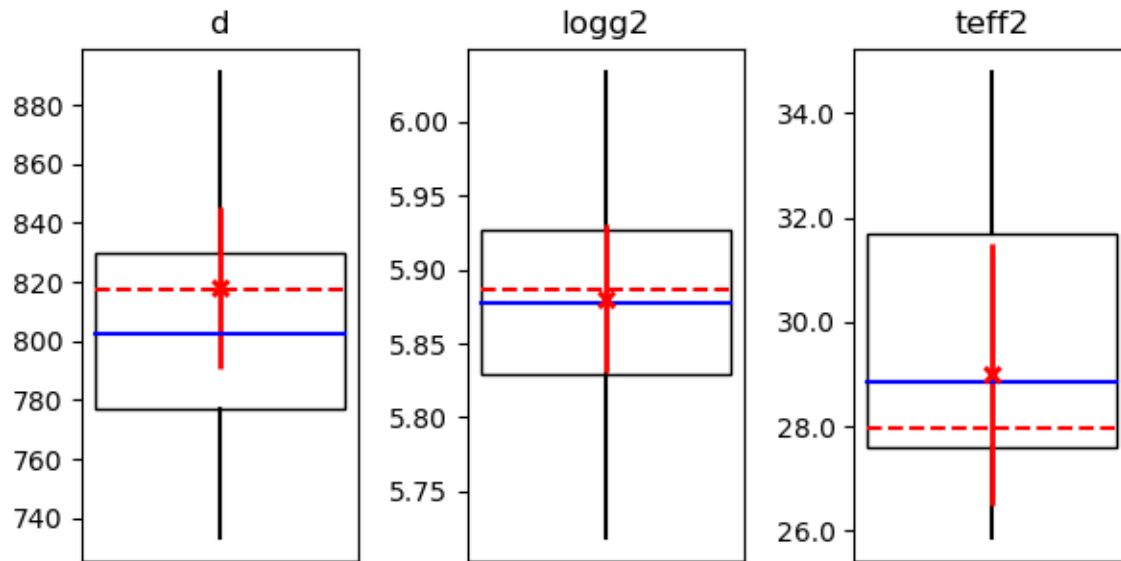
## 9.2 Constraint plot

Type keyword: constraint

The constraint plot shows the applied constraints in red crosses with the error bars, together with the best fitting model (lowest Chi2) in red dashed line, the median model in blue full line and the distribution of the walkers as a whisker plot.

This plot type does not have option arguments

An example of this plot for the binary system BD-7 5977, with constraints on the distance, surface gravity and effective temperature from spectroscopy is shown:



### 9.3 Distribution plot

Type keyword: `distribution`

The distribution plot shows the distribution for all fitted and derived parameters. This plot type is build on top of the [corner plot](#) (Daniel Foreman-Mackey 2016).

**parameters** (list)

A list of parameters (fitted or derived) to be included in the distribution plot

**show\_best** (bool)

When true, indicate in red line the parameter value with the lowest chi2 (default = False)

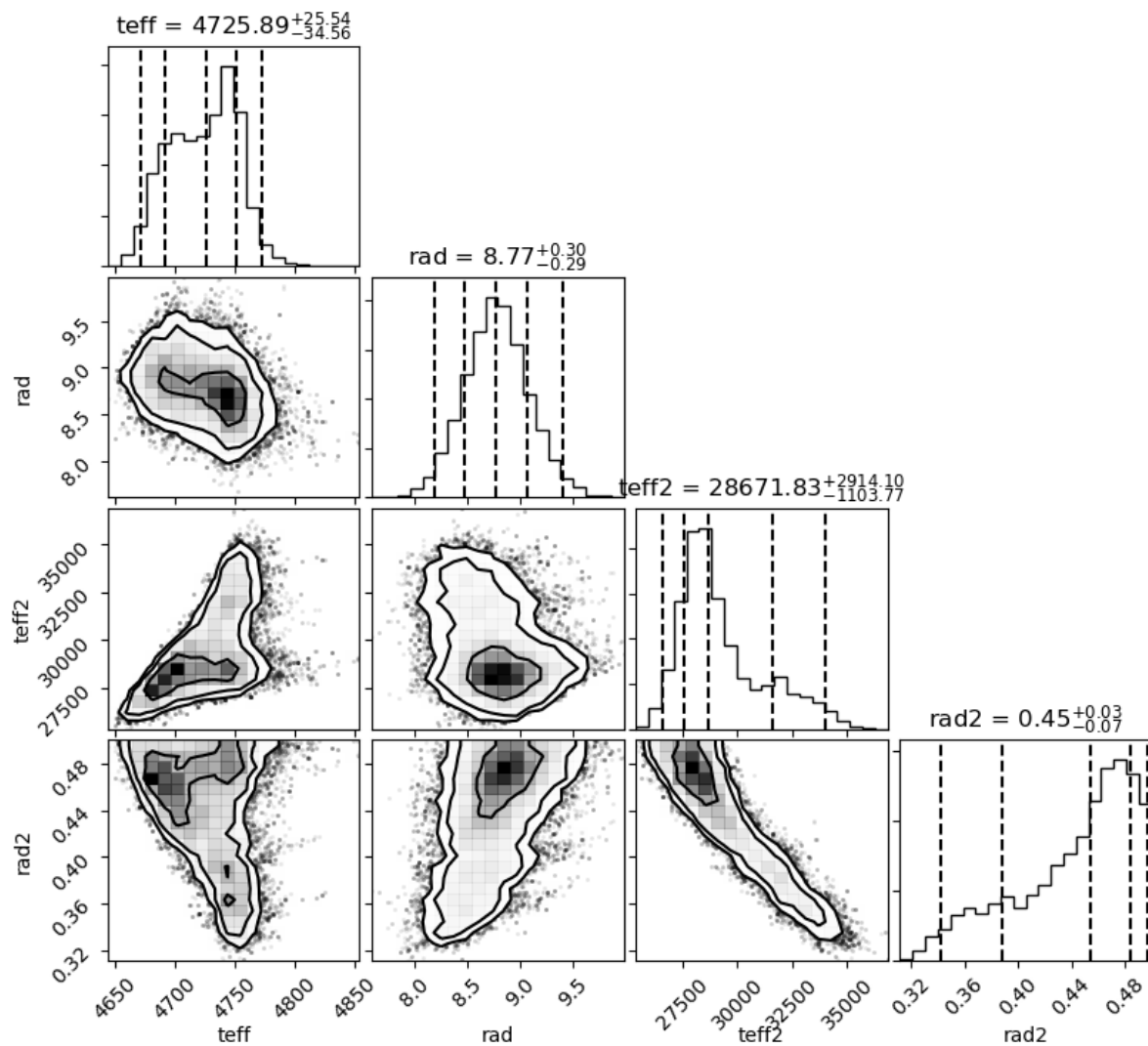
**quantiles** (list)

List of the quantiles to mark on the corner plot (default = [0.025, 0.16, 0.5, 0.84, 0.975], corresponding to 1 and 2 sigma). See the [API documentation](#) of corner plot for more details.

**levels** (list)

List of the contour levels to draw on the 2D histograms. (default = [0.393, 0.865, 0.95], the latter two corresponding to 1 and 2 sigma). See the [API documentation](#) of corner plot for more details.

An example of this plot for the binary system BD-7 5977 with the distribution of the walker results for the effective temperature and radius of both components is shown:





## CHAPTER 10

---

### Indices and tables

---

- `genindex`
- `modindex`
- `search`



## Symbols

-bands  
     speedyfit-checkgrids command line  
     option, 17

-noplot  
     speedyfit-fit command line option,  
     16

-nopx  
     speedyfit-setup command line  
     option, 16

-phot  
     speedyfit-setup command line  
     option, 16

-grid <model\_grid>  
     speedyfit-setup command line  
     option, 15

-o <output\_file>, -output  
     <output\_file>  
     speedyfit-phot command line option,  
     16

## A

a (*int*)  
     command line option, 22

## C

command line option  
     a (*int*), 22  
     constraints (*dict*), 21  
     derived\_limits (*dict*), 21  
     err\_index (*str or int*), 20  
     grids (*list*), 21  
     levels (*list*), 43  
     limits (*list of lists*), 21  
     nrelax (*int*), 22  
     nsteps (*int*), 22  
     nwalkers (*int*), 22  
     objectname (*str*), 20  
     obs\_index (*str or int*), 20

parameters (*list*), 43  
     percentiles (*list*), 22  
     photband\_exclude (*list*), 20  
     photband\_include (*list*), 20  
     photband\_index (*str or int*), 20  
     photometryfile (*str*), 20  
     plot<n> (*dict*), 22  
     pnames (*list*), 21  
     quantiles (*list*), 43  
     result ('best' or 'pc'), 41  
     resultfile (*str*), 22  
     show\_best (*bool*), 43  
 constraints (*dict*)  
     command line option, 21

## D

derived\_limits (*dict*)  
     command line option, 21

## E

err\_index (*str or int*)  
     command line option, 20

## G

grids (*list*)  
     command line option, 21

## L

levels (*list*)  
     command line option, 43  
 limits (*list of lists*)  
     command line option, 21

## N

nrelax (*int*)  
     command line option, 22  
 nsteps (*int*)  
     command line option, 22  
 nwalkers (*int*)

command line option, 22

## O

object\_name

speedyfit-phot command line option,  
16

speedyfit-setup command line  
option, 15

objectname (*str*)

command line option, 20

obs\_index (*str or int*)

command line option, 20

## P

parameters (*list*)

command line option, 43

percentiles (*list*)

command line option, 22

photband\_exclude (*list*)

command line option, 20

photband\_include (*list*)

command line option, 20

photband\_index (*str or int*)

command line option, 20

photometryfile (*str*)

command line option, 20

plot<n> (*dict*)

command line option, 22

pnames (*list*)

command line option, 21

## Q

quantiles (*list*)

command line option, 43

## R

result (*'best' or 'pc'*)

command line option, 41

resultfile (*str*)

command line option, 22

## S

setup\_file

speedyfit-fit command line option,  
16

show\_best (*bool*)

command line option, 43

speedyfit-checkgrids command line  
option

-bands, 17

speedyfit-fit command line option

-noplot, 16

setup\_file, 16

speedyfit-phot command line option

-o <output\_file>, -output  
<output\_file>, 16

object\_name, 16

speedyfit-setup command line option

-nopx, 16

-phot, 16

-grid <model\_grid>, 15

object\_name, 15