

VADEMECUM*
How to use SP_Ace V1.4

Corrado Boeche
corrado.boeche@gmail.com

July 7, 2020

*"Tutorial" in latin. Literally it means "follow me".

1 Foreword and some advices

SP_Ace is a software written in FORTRAN95 that analyses stellar spectra to derive stellar parameters (T_{eff} and $\log g$) and chemical abundances [E/H]. SP_Ace is a user friendly tool, designed for large spectroscopic surveys (but can also be used for just one or few spectra) and provide results of high accuracy even at low spectral resolution. However, like any other tool, SP_Ace must be employed with some knowledge and common sense. If used blindly, SP_Ace may give meaningless (or even wrong) results. SP_Ace is free (GPL licence) software provided without any warranty.

I take care of fixing the bugs as soon as I get to know them. However, there may be bugs I have not met yet. In case you find one, please let me know and I will try to fix it.

Before to use SP_Ace on a sample of spectra, it may be a good idea to test it on spectra of standard stars to verify if there is any systematic error introduced by the normalization applied or by the wavelength range employed (some internal systematic errors were discussed in the papers Boeche & Grebel, [2016] and Boeche, Vallenari, Lucatello [2020], hereafter reported as Paper I and Paper II).

2 Basic information

The present version (v1.4) of SP_Ace performs the spectral analysis under LTE assumption and based on 1D atmosphere models. The stellar parameter ranges covered by SP_Ace are

$$3600 < T_{\text{eff}}(\text{K}) < 7400$$

$$0.2 < \log g < 5.4$$

$$-2.4 < [\text{m}/\text{H}](\text{dex}) < 0.4$$

SP_Ace does not converge for stars with stellar parameters out of these intervals.

3 Updates on the recent versions

3.1 v1.1

With respect to the SP_Ace version v1.0 employed in Paper I, the version v1.1 had the following changes:

- Numerical Recipes routines has been replaced with equivalent free software routines. With this change, SP_Ace is now a free software code which source code is released under GPL licence.
- the order of the iteration loops has significantly changed. Now, the internal re-normalization is done inside a first “normalization loop” (which serves also as first parameter estimation), and in the “outer loop” is performed the refinement of the parameters (and no re-normalization). The “inner loop” of the version v1.0 has been removed.
- SP_Ace can now handle spectra with variable dispersion
- SP_Ace can now handle spectra with S/N that varies along the wavelenghts (as it happen for the echelle orders because the blaze function, for instance)
- SP_Ace now neglects *all* the Paschen HI lines in the near infrared but the HI line at 8750Å. Test at medium resolution ($R \sim 7000$) showed that it brings improvements in gravity estimation. Thus, we decided to keep it.

- a new internal re-normalization that can take in account the pseudo-continuum created by strong blended lines. This change is particularly effective for low resolution spectra ($R \sim 2\,000$, where it is needed), while for higher resolution it has no effect (and it is not needed).
- The uncertain estimation routine has been changed (although the main idea of estimation is still the same). From the tests performed by us, the errors are consistent with the ones given by the previous routine.

With this changes, SP_Ace is a little slower than before (but not the error estimation, which is faster!) but now the continuum placement is more robust and nearly independent from the normalization done by the user and from the S/N ratio (in particular at low spectral resolution).

3.2 v1.2

With respect to the version v1.1, the version v1.2 had the following changes:

- fixed an imprecision on the computation of the normalized χ^2 reported in the output file. (This has no effect on the parameter computation, it's only an output issue).
- added one column on the same file that report information on the convergence of the measurement and the reason of the failure (if any). See more in Sec. 7.1
- a study on the error estimation revealed that when `ABD_loop=.false.` the the estimated errors can be inconsistent with the resulting stellar parameters. The problem can be fixed only by changing in the GCOG library. This is work in progress and it will be implemented in the next version of SP_Ace.
- the error routine has been slightly changed to fix a bug. The present results are consistent with the previous one, although in some cases they can be different.
- the interval on which the $H\alpha$ line is constructed has been enlarged to avoid some discontinuity in the χ^2 hypersurface observed in the previous version. This fix remove this discontinuity. As a consequence, the resulting stellar parameters are slightly (but not significantly) different. A bug in the main program has been found and fixed. The resulting stellar parameters of the fixed version are different from before, but the difference is not significant. This is showed with tests on synthetic and real spectra reported in Sec. 11.

3.3 v1.3

With respect to the version v1.2, the present version v1.3 has the following changes:

- The new optional keyword `RV_ini` has been implemented. It sets the initial RV (that was previously set to zero). See Sec. 6.2.
- The normalization of the parameters has been implemented. The results have now negligible differences with respect to the previous version of the code. I implemented some optimizations in the routines that build the models. These changes make the code faster.
- A new way to compute uncertain has been implemented. The code is faster and the errors estimation look reliable, but see discussion in Sec. 10

3.4 v1.4

With respect to the version v1.3, the present version v1.4 has the following changes:

- computed a new GCOG library (version v2.2) based on the synthesis code SPECTRUM (Gray & Corbally, [1994]). The new GCOG library covers the wavelength range 4800-6860Å. The opacity of the neighbour lines has been computed with a new method (see Paper II).
- introduced the keyword `Salaris_MH` that applies the Salaris formula (Salaris et al. , [1993]) to internally compute the metallicity to adopt when reading the GCOG library. This become necessary for the way the GCOG library v2.2 has been built (see Boeche, Vallenari, Lucatello [2020]).
- optimized the computation of the polynomial EW to make the code faster
- as option, it takes the name of the parameter file from the command line. If no name is given, it assumes that the name is “space.par”. (This is useful when `SP_Ace` is run in parallel in the same directory, so that every process has a different parameter file name).
- the output header format has been changed to avoid the `N_lines` column names repetition
- with the present version, when the code do not converge in 30 iterations during the first estimate (the so called “quick loop”), the estimate goes to the end, but output `conv=1` (instead of `conv=0`)
- because the new line list, new element abundances (up to 22 elements) are allowed to be output.
- a new line profile has been implemented.
- a Graphical User Interface (GUI) is provided to run `SP_Ace`.

4 The source code

The source code can be downloaded from http://dc.g-vo.org/SP_ACE together with the GCOG library v2.2.

To compile `SP_Ace`, move into the source directory and type “make”. If everything goes well, you should find the binary “SPACE” that can be moved into the working directory or in a directory where the system can find the executable file. If you have a `bin/` directory in you home directory, you can just type “make install”.

5 Warnings

`SP_Ace` can handle spectra *that have these features*:

- Spectral resolution between ~1 000 and ~40 000.
- the spectra must be in the wavelength ranges 4800-6860Å.
- the spectra must have less than 128 000 pixels.

- the spectra must be wavelength calibrated, continuum normalized, and radial velocity corrected at rest frame¹.
- The spectra should have constant (or nearly constant) FWHM across the wavelength interval covered.

If these conditions are not matched, you may have unreliable results, although there is some flexibility. For instance, if the radial velocity correction has an error of a fraction of FWHM, SP_Ace can correct this and everything will be fine. If the resolution (i.e. FWHM) is not perfectly constant but varies of a small fraction between the bluest and the reddest part of the spectrum, this will not significantly affect the results (but if it is double, this will! The fit of a variable FWHM as a function of wavelength may be implemented in the future. If you have this concern, please let me know).

Another variable that may create concern is the rotational velocity V_{rot} . SP_Ace does not derive V_{rot} and it cannot properly fit the line profile of a high V_{rot} star. However, there is some flexibility because a wider line profile due to high V_{rot} can be reasonably fit by a Voigt profile with a wider FWHM. This is true only if the extra width of the line caused by stellar rotation does not exceed the width of the instrumental line profile. This means that the V_{rot} that SP_Ace tolerates is higher for lower spectral resolution. If the V_{rot} is exceedingly high for SP_Ace, this may derive unreliable results that one should recognised because the resulting normalized χ^2 should be significantly bigger than 1 (but see Sec. 8).

6 The parameter file

When launched, SP_Ace looks for the parameter file, starts the analysis of the spectrum declared in the parameter file and exits. The parameter file holds several information necessary to run SP_Ace like the name of the spectrum to analyse, the address of the library, and other parameters. Some of them are absolutely necessary to run SP_Ace, some others are optional and SP_Ace can happily run without them.

When launch SP_Ace with no argument, the parameter file must called `space.par` (this is the name that SP_Ace look for by default). Otherwise, one can name the parameter file with another name (say, “`param.space.dat`”) and then launch SP_Ace followed by the name of the file with the command `SPACE param.space.dat`.

Each line of the parameter file starts with one keyword and can be followed by a string (the name of the spectrum, for instance), by one or more numbers, or, for some keywords, nothing. If the keyword is followed by a string or a number, they must be separated with a space. The keywords can be listed with any order. Comment lines have the symbol # as first character.

The following is an example of the `space.par` file holding the four necessary keywords only:

```
obs_sp_file '/home/corrado/spectra/spectrum.asc'
GCOGlib '/home/corrado/SPACE/GCOG_library/'
fwhm 0.4
wave_lims 5000 6500
# this is a comment
```

¹With the present version v1.4 it is possible to give an approximate initial radial velocity as input, see the keyword `RV_ini` in Sec. 6.2.

6.1 Necessary keywords

These are four keywords without which SP_Ace cannot run. They are:

- `obs_sp_file`: as suggested by the name, this keyword refers to the name of the spectrum to analyse. Give the absolute address of the spectrum if the spectrum is not in the working directory. The name of the spectrum must be bracketed by apexes ”.
- `GCOGlib`: this refers to the General Curve-Of-Growth Library on which SP_Ace relies on. Give the absolute address bracketed by apexes ”.
- `fwhm`: here give a first guess of the Full-Width-Half-Maximum of the instrumental profile of the spectrum. A precise FWHM is not necessary, because SP_Ace refines the FWHM itself. So, just give a rough estimation as starting point, for instance give 3.0 for a spectrum with resolution $R \sim 2000$ or 0.4 for $R \sim 12\,000$. This will be good enough.
- `wave_lims`: put the wavelength limits on which the analysis is performed. As showed in the previous example, the first number is the begin and the second is the end of the wavelength interval under analysis. It is possible to give multiple wavelength ranges (up to five) with the rule

```
wave_lims ini1 end1 ini2 end2 ini3 end3
```

for instance:

```
wave_lims 4800 5800 6200 6860
```

In this example the analysis is performed on the wavelength ranges 5212-5800Å and 6200-6860Å, while the range 5800-6200Å is neglected.

Be aware that every wavelength interval contain different lines that may cause different systematic errors.

- `Salaris_MH`: when not explicitly called, this keyword is called by default because it is necessary (as said before) for the way that the GCOG library v2.2 has been built. Then also the keyword `ABD_loop` is called internally. Therefore *when the GCOG library v2.2 is used, the keyword Salaris_MH must be called (or it is called by default)*.

6.2 Optional keywords

The optional keywords are not necessary, but even when the are not declared in the file `space.par`, some of them are internally defined with a default value. The optional keywords and their default values are:

- `sn_ratio`: if you know precisely the signal-to-noise ratio of the spectrum you can declare it here and the whole analysis will be done with this parameter fixed. For instance

```
sn_ratio 100
```

If this keyword is absent, SP_Ace estimates the S/N for you. Keep in account that two analyses performed with different S/N can lead to different results (albeit they should not be dramatically different).

- `ele2write`: SP_Ace always measures all the elements that have at least one measurable absorption line in the chosen wavelength range (they can be up to 30 elements), but many of them give (or might give) not reliable abundances. With the present version SP_Ace outputs the results of 22 elements. Sometime it happens that one or more of the ten elements have a poor estimation and we know we don't want to use it. In this case we declare which elements we want to print in the output by writing the atomic number of the wanted elements after the keyword `ele2write`. The following example

```
ele2write 12 14 20 22 28
```

writes in the output the chemical abundances of the element Mg, Si, Ca, Ti, Ni (and Fe, which is always written by default, it is not necessary to declare it). In case one element could not be estimated, SP_Ace returns the null value for this element. When the keyword `ele2write` is missing, SP_Ace write the results for all the ten elements by default.

- `error_est`: this keyword switch on the error estimation. It has no argument (it is not followed by any number or string). When present, SP_Ace computes the confidence interval limits that bracket the 64% probability of the three stellar parameters (T_{eff} , $\log g$, and $[m/H]$) and chemical abundances of elements declared with the `ele2write` keyword. Be aware that it can significantly slow down the speed of SP_Ace. By default, the error estimation is switched off.
- `null_value`: you can choose how the null value has to look like. The three possibilities are: -9.99, 'null', or 'NaN'. For instance

```
null_value '-9.99'
```

The default value is 'null'.

- `l1ist_rej`: with this keyword the user can exclude from the analysis absorption lines or small wavelength intervals. The centers of the lines (or the intervals) with their semi-widths must be listed in a file and the name of this file must follow the keyword. Example: suppose that your spectrum has some defects at $\lambda = 5234.5$, 5512.0 , and 5606.2\AA and you want SP_Ace to neglect them during the analysis. Then, you must prepare a file called (for instance) "lines2rej.dat" that contains the following rows

```
5234.5 2.0
5512.0 1.0
5606.2 5.0
```

The first column contains the central wavelengths and the second column the semi-width (expressed in \AA) of the interval you want to neglect during the analysis. To these intervals, SP_Ace gives `weight=0`. The keyword is used as follow

```
l1ist_rej 'lines2rej.dat'
```

By default, SP_Ace rejects some small intervals that contain lines that suffer from NLTE effects or problems in the opacity correction of the neighbor lines in the GCOG library. You can identify these intervals because their pixels have `weight=0` in the output file `space_model.dat` (see Sec. 7.2).

- `T_force`: SP_Ace can take T_{eff} as input and derives the other parameters. Suppose that your spectrum has a low S/N and SP_Ace's T_{eff} is of poor quality (or, even worse, SP_Ace cannot converge for this spectrum). Suppose that you have a good photometric estimation of T_{eff} (say 5777). Then you can use it and put in the parameter file the following

```
T_force 5777
```

With this keyword SP_Ace keep the T_{eff} constant at the value 5777 during the whole analysis, and estimates $\log g$, $[m/H]$, and chemical abundances only.

- **G_force**: like before, you may want to give $\log g$ as input. Then, if you know the $\log g$ (say 4.44), you write in the parameter file

```
G_force 4.44
```

and SP_Ace estimates T_{eff} , [m/H], and the chemical abundances only. **T_force** and **G_force** can be used together, letting SP_Ace to estimate chemical [m/H] and chemical abundances only. For instance

```
T_force 5777
```

```
G_force 4.44
```

- **norm_rad**: it is followed by a number that affect the re-normalization of the spectrum performed by SP_Ace. A unwise value of this parameter can lead to wrong results. You are invited to not to use this keyword unless you know very well what you are doing. Anyway, I think the user has the right to try. If you really want to play around with this parameter, you need to know what this parameters represent. In short, **norm_rad** is the semi-width (expressed in Å) of the wavelength interval centered on the i -th pixel employed to estimate the continuum of that pixel (see point 1 of Sec. 7.4 of Paper I).

In practice, if the **norm_rad** is a small number (10 or 30, for instance) the line that fit the continuum is more “flexible” and capable to follow a curvy continuum. The larger the number, the more stiff the fitting line become. If you give a value that is larger than the spectral range covered by the spectrum (expressed in Å), the line that fit the continuum will be a straight line which can only shift up or down. If your spectrum suffers of fringes that the previous normalization could not fully remove, you may try to use a small value, with the recommendation to avoid too small values (5-10 should be the lower limit, depending on the spectrum). The idea behind this limit is that **norm_rad** should not be smaller than the width of the largest line seen in the spectrum. Because some strong line have broad wings spanning up to 10Å to avoid that the continuum fits the shape of these lines, the **norm_rad** value must be larger than the width of the largest line and $\sim 5 - 10$ times larger than the instrumental FWHM. So that, for a spectrum of resolution $R=2000$ one may want to use **norm_rad**=30 or larger, while for $R=20\,000$ one may try to use **norm_rad**=10. All the tests illustrate in Paper I and Paper II have been performed with the default value

```
norm_rad 30.
```

which proved (up to now) to be good enough for most of the cases. SP_Ace has an internal lower limit that is 5. Any value smaller than that is taken as it was 5.

- **no_norm**: this keyword switches off the internal re-normalization of SP_Ace. In general, this is not recommended because SP_Ace proved to do a better job than a manual normalization. In particular, this is not recommended for low resolution spectra, where the lines are instrumentally blended and the normalization commonly done (with or without eye inspection) with the IRAF task *continuum* cannot count for the pseudo-continuum generated by the blended lines. If you really want to use this keyword, it make sense only if your spectra have a high resolution and are good enough that you can clearly recognize the continuum. Otherwise let SP_Ace to do the job.
- **alpha**: this keyword compels SP_Ace to treat the elements as there were only two elements: “metal” and “alpha”. In other words, the abundance of the alpha elements (Mg, Si, Ca, Ti) are derived as they were the same element “alpha”, and the abundances of all the other elements are derived as they were only one element “metal”. The elements C,N,O, and the molecule CN are not considered part of the “alpha” nor of the “metal” elements. This option is useful

for spectra that hold little information because handling only two "elements" dramatically decrease the degeneracy in the parameter space. This can help to get meaningful results from poor quality spectra when SP_Ace cannot converge to solution by looking into the full parameter space. In my tests I found this useful for $R \sim 2000$ but not necessary for $R \gtrsim 5000$. When employed, the `alpha` keyword give an output that contain the "metal" abundances $[m/H]$ and the "alpha" abundances $[\alpha/H]$.

- `ABD_loop`: it switches on the loop between the routine that search for the best stellar parameters and the one that derives the chemical abundances. With the previous GCOG library (v1.0, released in Paper I) `ABD_loop` was recommended to be switched off (see discussion in Sec.8.5.1 of Paper I). However, with the new GCOG library v2.2 `ABD_loop` is switched on by default when the keyword `Salaris_MH` is called (and it must be called with the recent GCOG library! see Paper II).
- `RV_ini`: it sets the initial radial velocity in case the spectrum has not been radial velocity corrected and in case the user know the radial velocity with reasonable approximation (inside 1FWHM of shift in wavelength). To use it, just added to the file `space.par` something like this:

```
RV_ini -5.0
```

where the keyword is followed by a number that represent the RV expressed in km s^{-1} .

7 How to read the results

SP_Ace outputs four data files called "`space_TGM_ABD.dat`", "`space_model.dat`", "`space_ew_meas.dat`", and "`space_msg.txt`"².

7.1 The output file "`space_TGM_ABD.dat`"

In `space_TGM_ABD.dat` you find a header (in the first line) and the derived stellar parameters and chemical abundances (in the second line). If you did not choose to estimate errors, the first 14 columns of the file looks like this

```
conv  RV  FWHM  S/N  chisq  Teff  T_l  T_h  logg  L_l  L_h  MH  MH_l  MH_l
0  0.2  3.0  115.8  0.82  5740  null  null  4.00  null  null  -0.00  null  null
```

If you chose the error estimation (by setting the keyword `error_est`) it looks like this

```
conv  RV  FWHM  S/N  chisq  Teff  T_l  T_h  logg  L_l  L_h  MH  MH_l  MH_l
0  0.2  3.0  115.8  0.82  5740  5659  5851  4.00  3.89  4.20  -0.00  -0.06  0.05
```

Here you find the parameters that SP_Ace tries always to estimate. The first column "`conv`" says if the convergence was successful (`conv=0` or `conv=1` or failed (`conv>1`). In Tab. 1 the meaning of every possible value for `conv` is reported. The next columns are radial velocity RV, Full-Width Half-Maximum FWHM, Signal-To-Noise S/N, the final (normalized) χ^2 between the observed spectrum and the best matching model. Then follows the stellar parameters T_{eff} , $\log g$, and metallicity MH. Beside the stellar parameters there are two columns with suffix "`_l`" and "`_h`" which represent the

²This is when you give to the parameter file the default name "`space.par`". If you call it, say, "`mypar.dat`" and give it as argument when SP_Ace is called, then the output files will be named "`mypar_TGM_ABD.dat`", "`mypar_model.dat`", "`mypar_ew_meas.dat`", and "`mypar_msg.txt`".

Table 1: Values of the *conv* parameter are here outlined with their meanings. Only for values 0 or 1 SP_Ace produces output results.

<i>conv</i> value	meaning
0	SP_Ace successfully converged
1	normalization loop did not converged in 30 loops (but SP_Ace produces results anyway)
2	stellar parameters out of the limits covered by the GCOG library
3	FWHM does not converge
4	RV too far from zero (beyond 1FWHM in wavelength)
5	improper input parameters in minimization routine lmdif1
6	TGM minimization routine exceeded the maximum number of iterations
7	ABD minimization routine exceeded the maximum number of iterations
8	TGM outer loop did not converge
9	SP_Ace cannot open the spectrum (maybe wrong name or address?)
10	SP_Ace cannot open the GCOG library (maybe wrong name or address?)
11	SP_Ace cannot open the space_6degpoly.dat file (maybe wrong address of the GCOG library?)

lower and the upper limit of the 64% probability confidence interval. In case you chose no error estimation, the upper and lower limits has always the null value. In case you chose to estimate errors, you may find that sometimes one limit has the null value. This means that the errorbar goes beyond the limits of the parameter grid and it cannot be estimated.

After these 14 columns you find the derived chemical abundances. The columns for the chemical abundances depend on how many elements you decided to write with the keyword `ele2write`. For the present example I used the keyword

```
ele2write 12 14
```

(remember that the element Fe is always reported and there is no need to specify it). The “chemical” columns look like this

```

Fe      Fe_l  Fe_h  Fe_N      Mg      Mg_l  Mg_h  Mg_N      Si      Si_l  Si_h  Si_N
-0.01  -0.06  0.03  1164     -0.05  -0.33  0.02   14      0.04   0.01  0.13  176

```

The abundance of the element (with respect to the Sun) is reported with the name of the element only and the lower and upper limits have suffix “_l” and “_h”. Like before, they represent the 64% probability confidence interval. The suffix “_N” reports the number of absorption lines used to derive the chemical abundance. For instance

```

Mg      Mg_l  Mg_h  Mg_N
-0.05  -0.33  0.02   14

```

says that the abundance of Magnesium is -0.05 dex, that the confidence interval (probability 64%) spans the interval [-0.33,+0.02] dex, and the estimation has been obtained by using 14 Mg lines. (The larger the number of lines the more reliable is the estimation). A null value in the “_l” or “_h” columns means that the lower limit and the upper limit of the abundance are beyond the grid (which in this case spans from [El/m]=-0.4 dex to [El/m]=+0.6 dex).

7.2 The output file “space_model.dat”

This file contains 5 columns which are (from left to right): wavelength, original flux, re-normalized flux, best matching model, continuum fit to the original flux, pixel weights, and pixels signal-to-noise.

The first two columns represent the spectrum given by the user (and not modified by SP_Ace). The third column is the flux re-normalized by SP_Ace. The fourth column is the continuum level found by SP_Ace for the original flux, and this is used to re-normalize the spectrum. This means that the third column is obtained by dividing the second column by the fourth column. The fifth column gives the weight for every pixel. This means that the pixels with weight equal to zero have been neglected during the analysis. The sixth column reports the signal-to-noise estimated for each pixel.

With this file you can overplot the best matching model with the re-normalized spectrum to check the consistency of the two. Similarly, you can verify the continuum found by SP_Ace by overplotting it over the original flux.

7.3 The output file “space_ew_meas.txt”

This file contains three columns: the wavelength, the atomic number, and the EW corrected by the opacity of the neighbor lines (in mÅ) of the lines that SP_Ace used in the best matching model. Checking this file can be useful to verify the goodness of the spectrum model and identify the lines.

Warning: the EWs reported *are not* the EW of the observed lines, but *are the values of the polynomial GCOGs for the best matching spectrum model.*

7.4 The output file “space_msg.txt”

When SP_Ace cannot converge on a result or something goes wrong, SP_Ace stops and writes a message in this file, reporting the name of the spectrum that fails and an indication on what happens. This is an example:

```
### /home/corrado/spectra/Sun_spectrum.asc  
FWHM does not converge, SP_Ace exits with no results!
```

When the spectrum has low S/N, too narrow wavelength range or it has very low metallicity, there are no information enough for SP_Ace to converge in one or more of the wanted parameters.

8 Imprecision

To date, SP_Ace has an imprecision in the computation of the χ^2 that, although does not affect the goodness of the final parameters, I want the user to be aware of.

When no S/N is given as input by the user, SP_Ace computes the expected residual for each pixel (σ) as average residual between the best matching model and the observed spectrum over a fixed interval (51 pixels) centred on the i -th pixel with a 3σ clip to remove the deviant points and computes the S/N as $1/\sigma$. (The σ is then used in the computation of the $\chi^2 = \sum_i \left(\frac{\text{model}(i) - \text{spectrum}(i)}{\sigma(i)} \right)^2$). On the assumption that the best matching model is a good representation of the real spectrum, the discrepancy between model and spectrum is due to the pure shot noise. Conversely, the differences between the model and spectrum (differences that can be due to physical or chemical reasons) would be regarded as “noise” by SP_Ace and the derived $\sigma(i)$ would not represent the shot noise. In the real world, the differences between model and observed spectrum are due to both shot noise and imprecision of



Figure 1: The SP_Ace GUI.

the model, leading to the conclusion that σ and discrepancies model minus spectrum hold in the χ^2 formula are not fully independent. This has no effect in the minimization process because the σ is computed after every minimization loop and it is kept constant when the minimization routine runs. Because the partial dependence between residuals and σ , the final normalized χ^2 may be closer to 1 than it should be. Still, the normalized χ^2 is not equal to one because σ is not fully dependent to the discrepancies observed minus model.

To conclude, we can say that the χ^2 is a “pure” one when the S/N is given as input by the user, otherwise it should be considered more like a parameter representing the goodness of fit. This *does not affect the robustness of the results* (because it does not affect the search for the minimum χ^2 .)

9 SP_Ace Graphical User Interface

We provide a Python Graphical User Interface (GUI) to simplify the use of SP_Ace. By launching the python code `python space_GUI.py` the window showed in Fig. 1 opens. From the window, the user can control all the SP_Ace options. One can run a single spectra process or provide a list of spectra that can be run one per time or all at once if the `batch mode` tick is given. In this case, the code runs the spectra launching parallel jobs. To save the results in a file it is necessary to click the `Save results` butto. The window is self explanatory.

10 About the errors estimates

With the version v1.3 I implemented a different method to estimate the uncertain (method kept in version v1.4). Instead of using a walking amoeba (employed in the previous versions, something similar to the Nelder and Mead minimization method) I decided to fit the 3D surface of constant χ^2 with an ellipsoid and search for the max and min extensions of its surface in the parameter space³. The shape of the 3D surface of constant χ^2 is not an ellipsoid but this is often a good approximation.

³As outlined in Paper I, SP_Ace derives the uncertain by measuring the extension of the χ^2 3D surface that satisfy at the equation $\chi^2 - \chi_{min}^2 = 3.53$ where χ_{min}^2 is the minimum χ^2 found (i.e. the convergence point). The method is outlined in “Numerical Recipes, Press et al., 1992).

As showed in Sec. 12 the uncertain obtained in this way are reasonably good. The new method is not expected to be better (in precision) than the old one but it is definitely much faster.

I would like to spend few more words about it because we don't have to expect great precision from the uncertain provided. In fact, the error of a stellar parameter in a multidimensional space (such as the stellar parameter space) cannot be expressed with a simple error bar. Usually we use the error for each parameter expecting that the error bars frames the interval that holds the 68% probability to have the right value. In the case of a stellar parameter (say T_{eff} for example) the upper and lower error bars provided by SP_Ace give the extension of the constant χ^2 3D surface that hold the 68% of the probability in the parameters T_{eff} , $\log g$, and $[m/H]$. Be aware that it is the 3D surface that hold a volume containing the 68% probability! When we consider its extension in one dimension, the other dimensions collapse and the error bars provided hold a probability that is larger than 68%. In other words, it is not possible to frame the 68% probability distribution of one parameter with error bars because the probability distribution is correlated with other parameters and the shape of the volume containing such probability is unknown (unless one explores it all with a monte carlo simulation, for instance). To be more informative we should provide the correlation matrix of the uncertain which is difficult to handle because it contains many values (in the case of SP_Ace these are 9 different values because SP_Ace computes the errors by considering 3 variables at time). To conclude, the error bars are a first approximations of the size of the volume that hold 68% of the probability. It can be a good approximation but it will never be a precise one. Please see the tests in Sec. 12.

11 Tests on synthetic and real spectra using SP_Ace v1.4

Here we repeated the tests on synthetic spectra and ELODIE spectra seen in Paper II with spectral resolution $R=2\,000$ and $R=20\,000$, and $S/N=100$ (please see Paper II for how the samples were constructed) using the present SP_Ace v1.4.

For the real spectra, we used the same spectra used in Paper II (ELODIE, benchmark and S4N spectra) degraded to lower resolution.

The results reported from Fig. 2 to Fig. 5 show that, respect to the previous version, there is a small improvement in gravity accuracy for $\log g > 3.5$. For other stellar parameters, the differences are not significant.

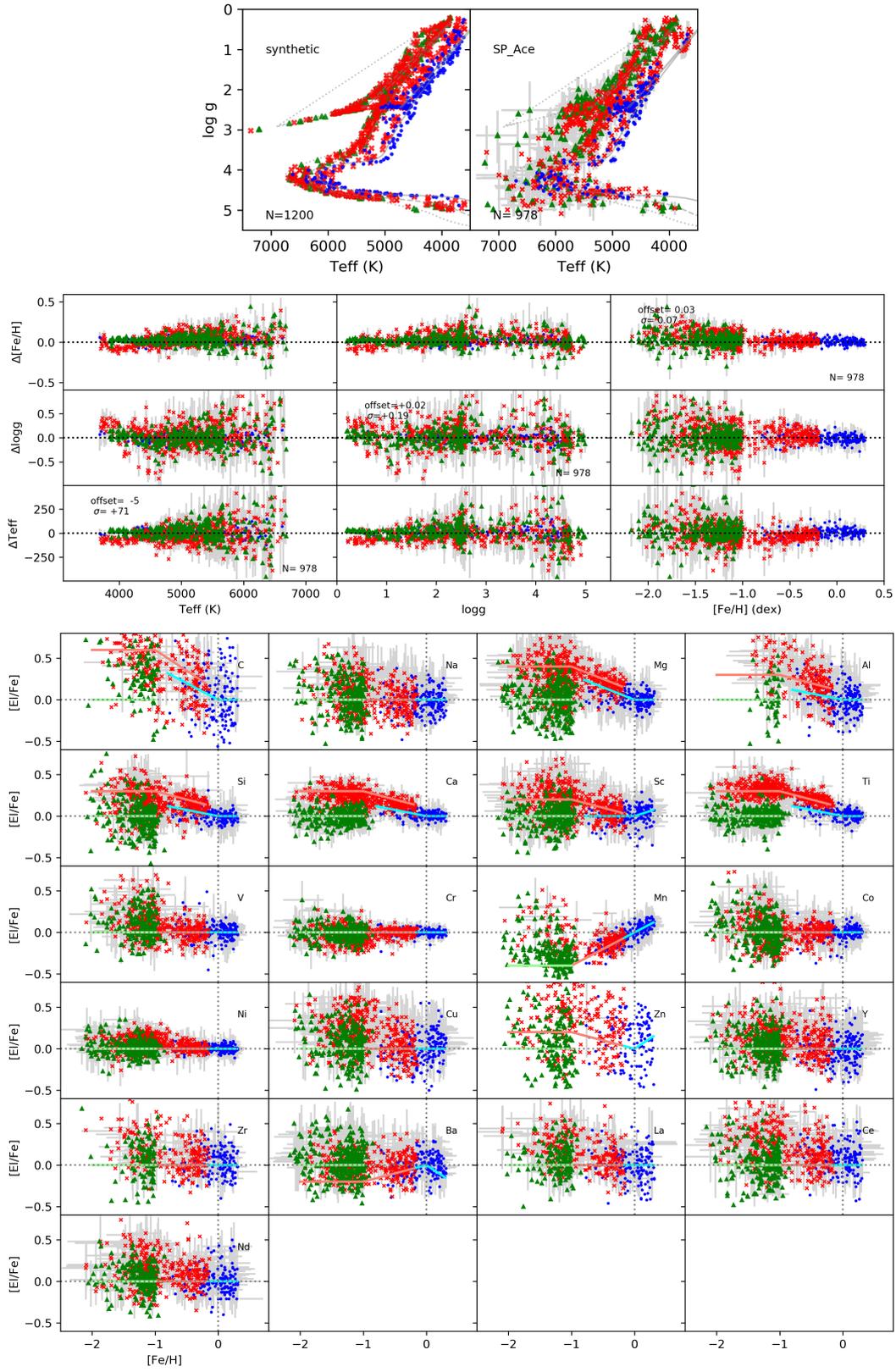
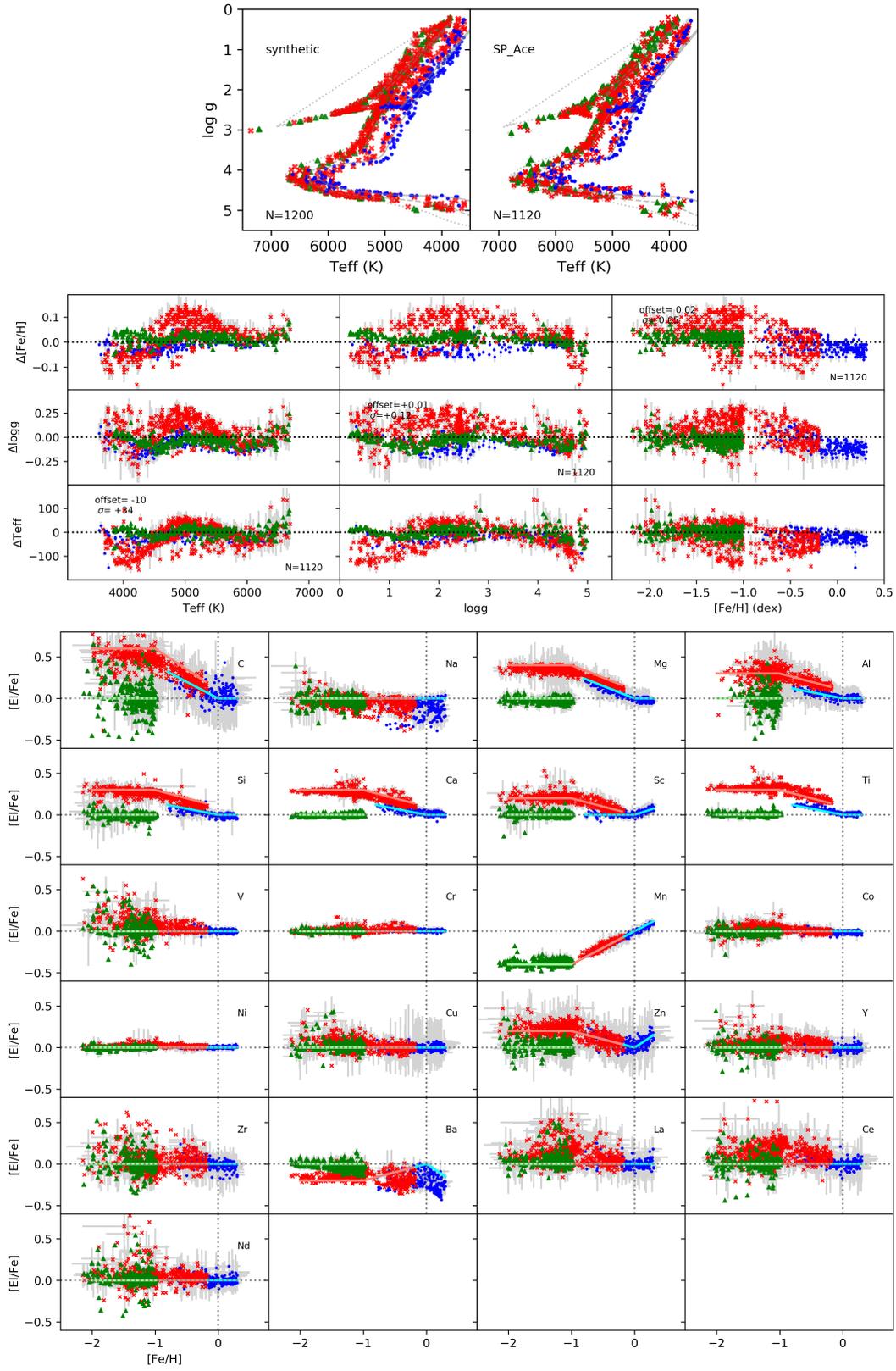


Figure 2: As in Figs. 6, C1, C2 of Paper II, but with $R=2000$.



16
Figure 3: As in Fig.2 but with $R=20\,000$.

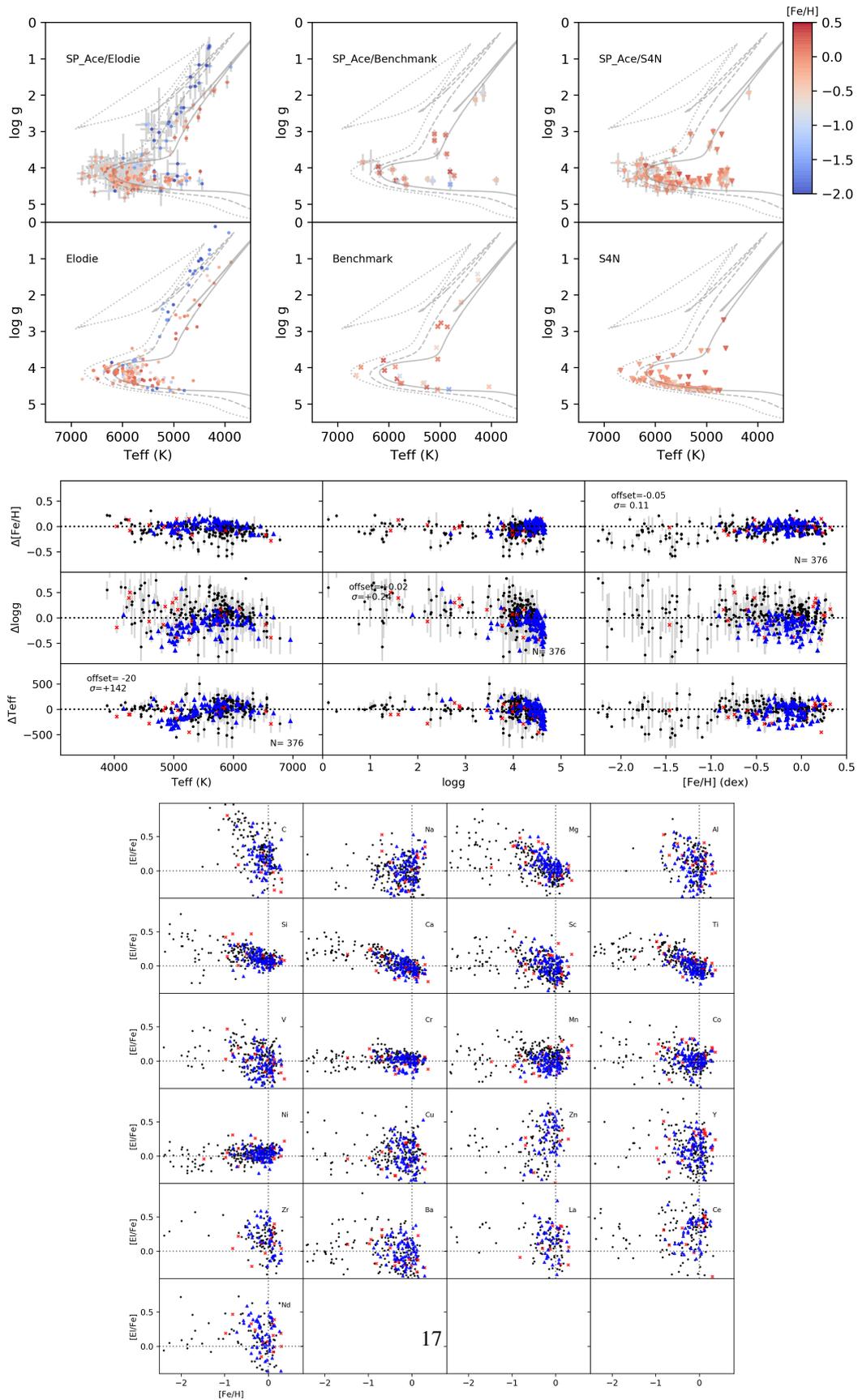


Figure 4: As in Fig. D1 of the paper, but with $R=2000$.

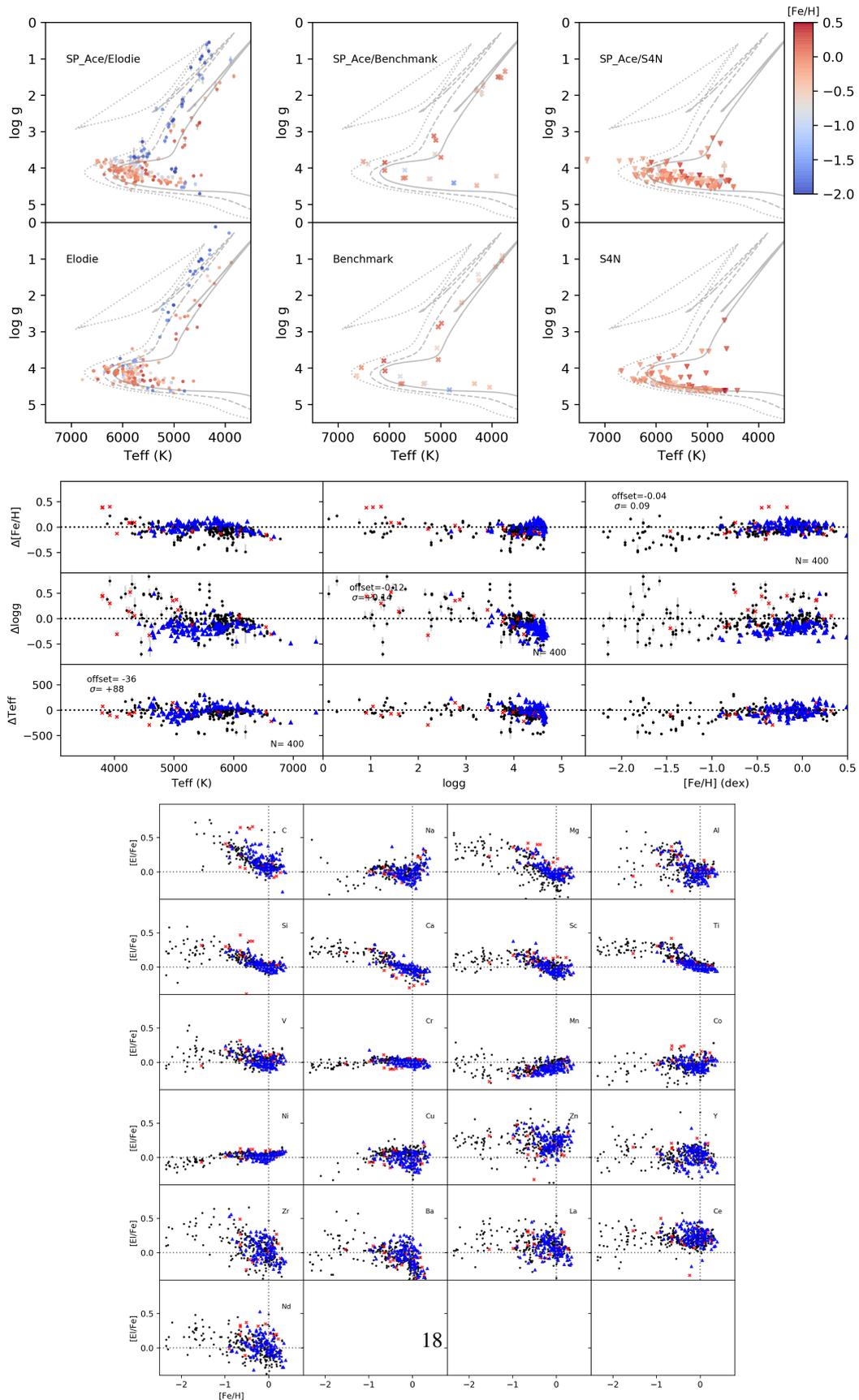


Figure 5: As in Fig.4, but with $R=20\,000$.

12 Tests on the error estimation accuracy using SP_Ace v1.4

The following tests were done with SP_Ace v1.4 and aim at verify the accuracy of the internal errors computed by SP_Ace.

I considered three synthetic spectra with different stellar parameters and I applied to them synthetic noise to reproduce three different values of $S/N=20, 50, 100$ (I here report $S/N=50$ only). Following a monte-carlo simulation, the stochastic noise was applied for 500 times for each spectrum and S/N value in order to map the shape of the error probability distribution and compare it with the internal error estimates computed with the method of the constant χ^2 3D-surface employed by SP_Ace.

In the following figures I show the results for the case of resolution $R=12\,000$ and $S/N=50$.

The size of the error bars (for the sake of clarity, we report these for only three of them) are of the same order of the extension of the distribution, although sometimes they appear to be too small (es example, see $[M/H]$ axis on top panels of Fig. 8). In Fig. 8 we can clearly see how the internal confidence intervals can only esteem the stochastic errors, while the systematic errors are not seen.

From these plots we can infer that the errors estimates provided by SP_Ace are reliable although not precise (but see the discussion in Sec. 10).

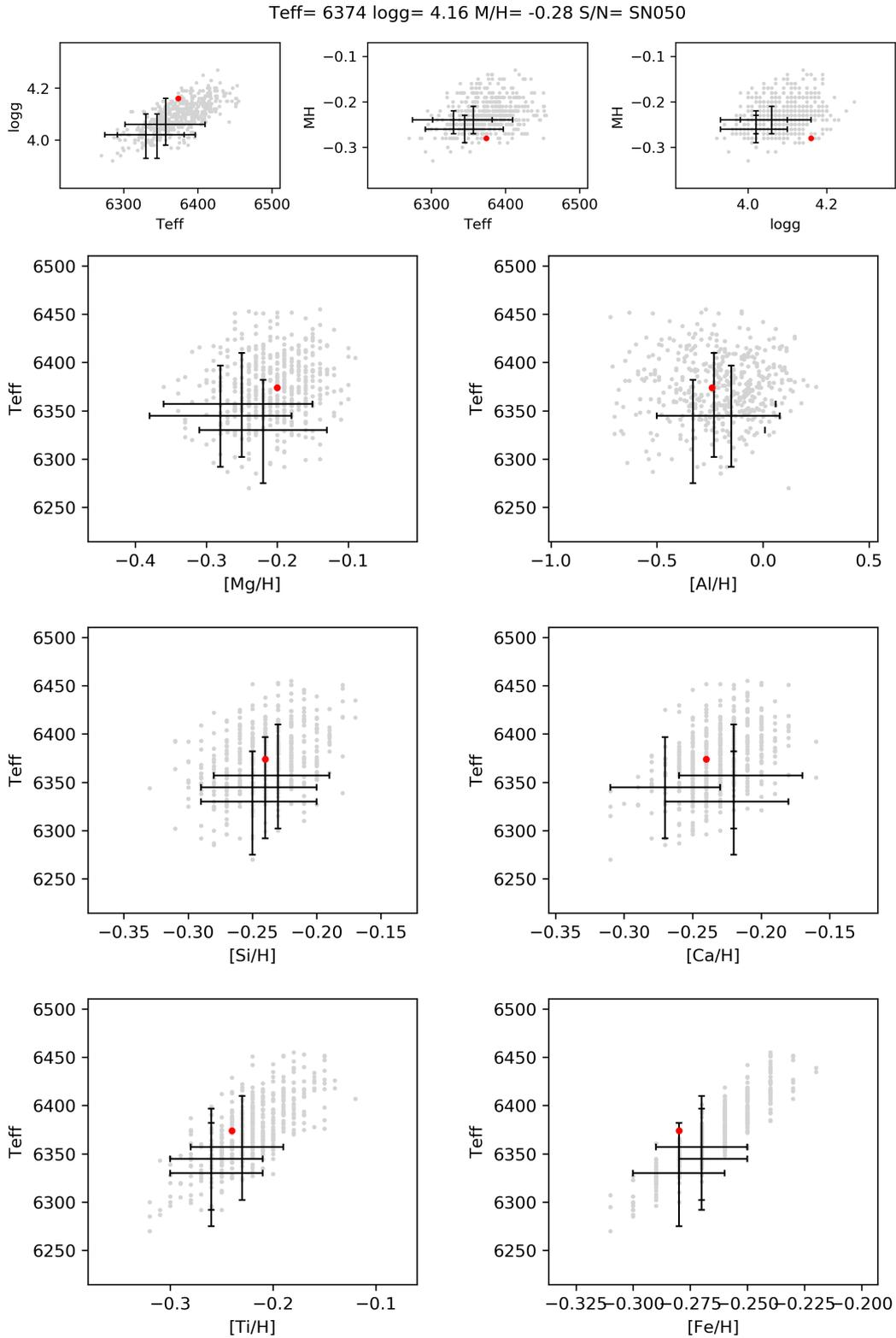


Figure 6: **Top:** SP_Ace measurements distribution in a monte carlo simulation of a synthetic spectrum with $T_{\text{eff}}=6374$ K, $\log g=4.16$, and $[m/H]=-0.28$ dex, $R=12\,000$ and $S/N=50$. The gray dots are the measurements of the 500 monte carlo realizations, the three of them are reported with internal error bars. The red dots are the reference parameters. **Bottom:** like before for six different elemental abundances as a function of T_{eff} .

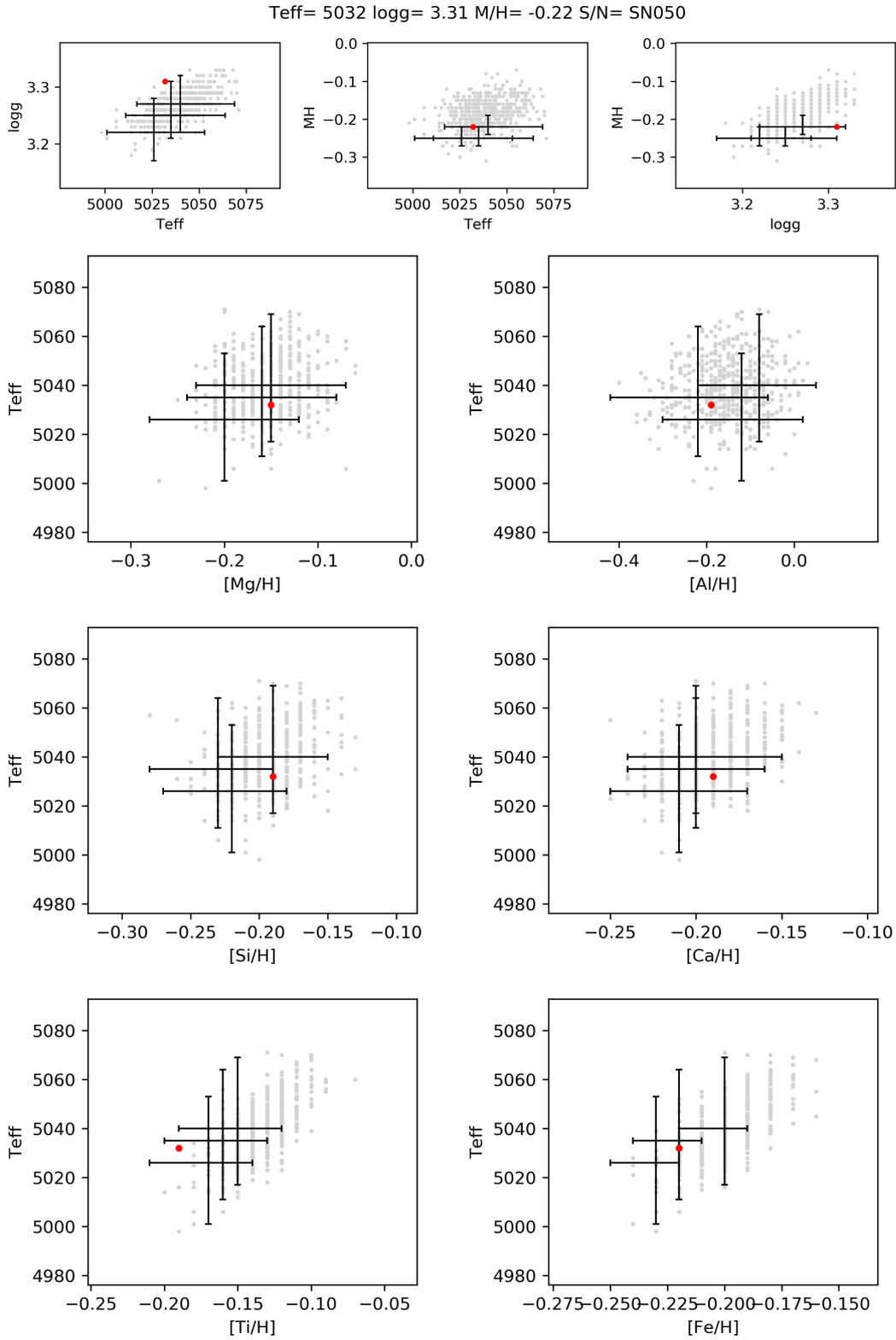


Figure 7: Like Fig. 6 but for a synthetic spectrum of $T_{\text{eff}}=5032$ K, $\log g=3.31$, $[m/H]= -0.22$ dex.

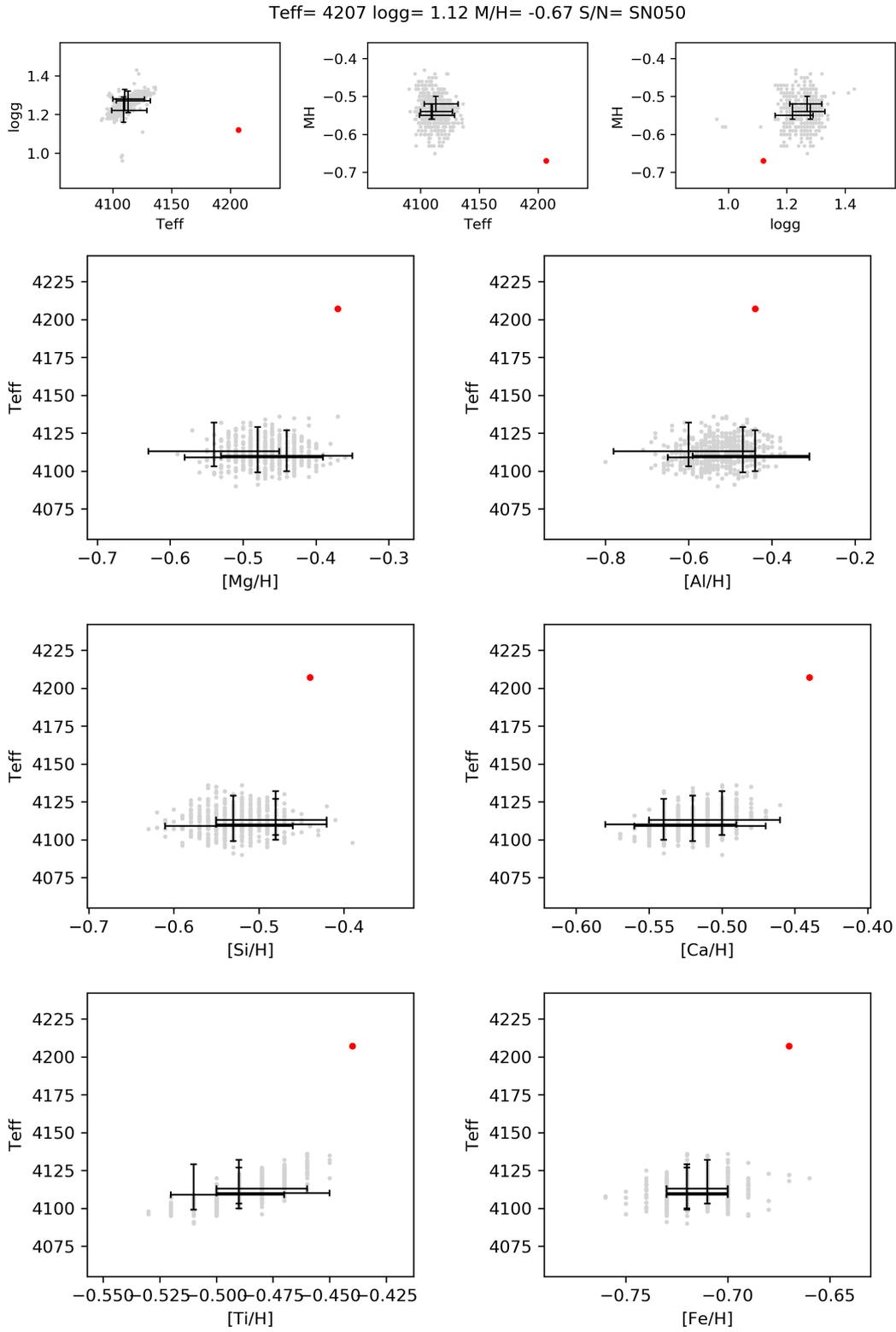


Figure 8: Like Fig. 6 but for a synthetic spectrum of $T_{\text{eff}}=4207$ K, $\log g=1.12$, $[m/H]= -0.67$ dex.

References

- [2016] Boeche, C., & Grebel, E. K. 2016, A&A, 587, A2
- [2020] Boeche, C., Vallenari, A., Lucatello, S., 2020, A&A submitted
- [1994] Gray, R. O., & Corbally, C. J. 1994, AJ, 107, 742
- [1993] Salaris M., Chieffi A., Straniero O., 1993, ApJ, 414, 580