



Introduction

Over 20 years ago Valenti & Piskunov developed the stellar spectral synthesis library Spectroscopy Made Easy (SME) [8,11] to simplify stellar spectral analysis, and it has been used in hundreds of works since.

PySME is a new Python frontend to the same library, developed with the same goal in mind: Making spectroscopy easier for everyone.

PySME creates accurate, high-resolution synthetic spectra, based on a range of stellar parameters, a linelist and a model atmosphere. It can also be used the other way around. Give PySME an observation and it will determine the best fit stellar parameters for that spectrum.

This is also useful for Exoplanet research. It is important to understand the host star, to properly understand the workings of the orbiting planet.



Fast and Reliable

PySME uses the latest version of the spectral synthesis code that incorporates 20 years of development



Open Source

PySME (and all its components) are completely open source



Easy to Use

PySME comes with a custom GUI to make inspecting your spectra easy

Features

Elemental Abundances

PySME doesn't just use the "classical" stellar parameters of temperature, surface gravity, and metallicity, it also accounts for individual elemental abundances. That also means that it can determine the best fit abundance of one or several elements to your observations.

It also handles a variety of different abundance formats, and converts them internally. You no longer need to convert your abundance data from one format into another, just specify which one you want to use.

VALD Linelists

Line lists are essential for any spectral synthesis as they describe which lines are in the spectrum, and which properties they have. Creating a good linelist is therefore crucial for a good analysis. Especially since you also don't want to waste time calculating lines, that are too weak to be seen in the spectrum anyway. Thankfully VALD [4,5,7,9,10] makes this extremely easy. Simply specify your estimated stellar parameters and your noise level, to get a comprehensive linelist that is optimized for these parameters. Plus various formats of VALD line lists can be directly read by PySME.

Model Atmospheres

PySME requires a model atmosphere to perform the radiative transfer calculations. For this purpose it provides access to a number of grids from MARCS [3] and ATLAS, most recently the MARCS 2014 grid, which covers temperatures from 2500K to 8000K, surface gravities from -0.5 to 5.5 log(cgs), and metallicities from -5 to 1 dex. Of course it is also possible to use your own grids or fixed atmospheres.

To avoid huge distributions the files are stored on an Uppsala server until they are requested for the first time, at that point they will be downloaded and stored locally. And all available files will be shown in the GUI.

Non-LTE Corrections

Thanks to ever more precise instruments, it is often not enough any more to assume Local thermal equilibrium (LTE) in your calculations. Therefore PySME supports the use of non-LTE departure coefficient grids. This significantly improves the accuracy for lines, that are affected by Non-LTE effects.

PySME provides access to a number of non-LTE grids, for various elements, that have been calculated on the MARCS 2014 grid [2]. To avoid huge distributions the files are stored on an Uppsala server until they are requested for the first time, at that point they will be downloaded and stored locally. And all available files will be shown in the GUI.

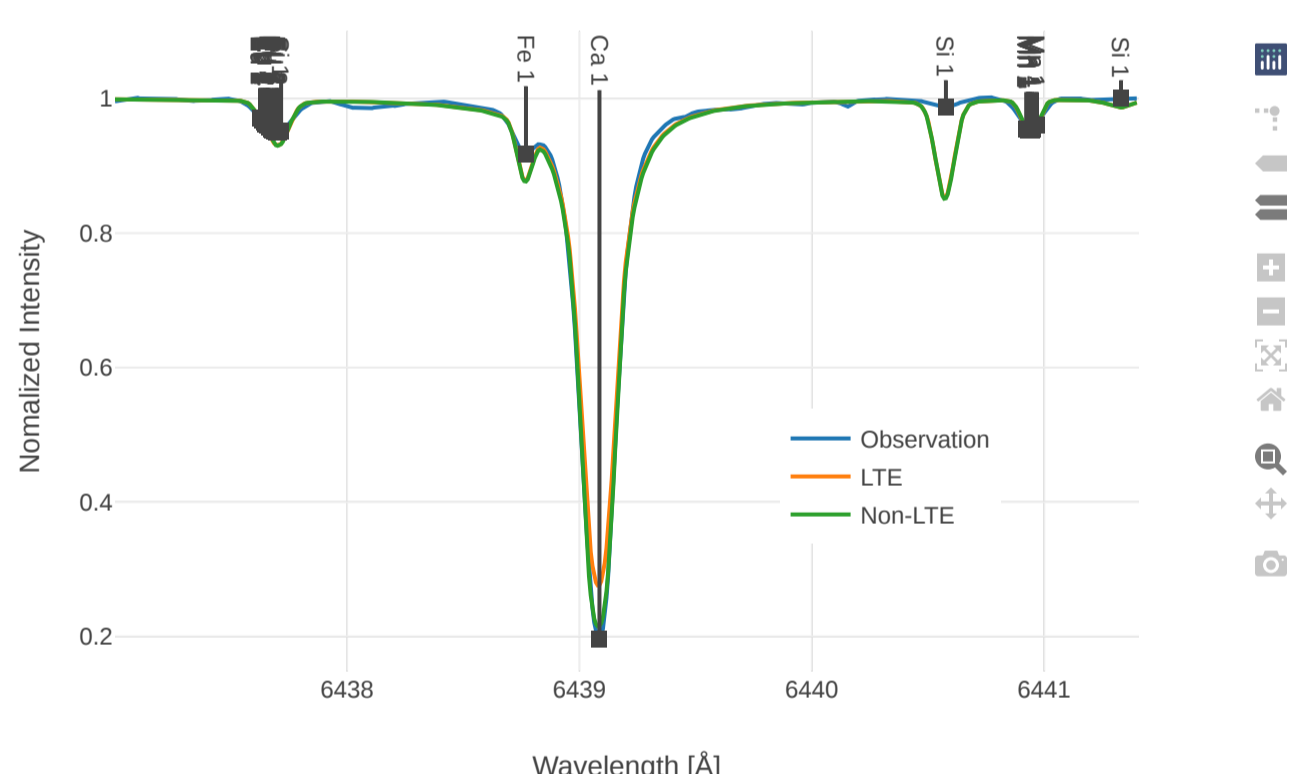
Backwards Compatibility

For easy adaptation it is possible to import existing SME input and output files from the original IDL implementation. Simply load the files in your script or in the GUI and PySME will take care of the conversion.

It is also possible to save the new PySME files as IDL files, provided that an IDL installation is available.

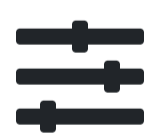
Open Source

PySME is completely Open Source, this includes the underlying SME library, that performs the radiative transfer calculations and is written in C++ and Fortran. This means that PySME can easily be run in parallel processing applications, since no licenses are required for it to run. Furthermore it can also be used by projects that require a fully open source approach, which is required by some funding agencies.

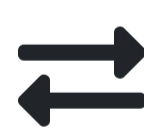


Example: The Sun

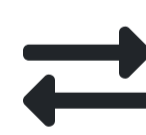
Here we display the capabilities of PySME on a small sample of the solar spectrum. For this we used a VALD linelist as input, set the abundances to solar abundances of Asplund 2009, and started with solar temperature and surface gravity. To illustrate the difference between LTE and non-LTE calculations both are shown in this plot. Non-LTE calculations are performed for Ca, Mn, and Si. Notice that the Non-LTE Ca line is much deeper than the LTE calculation.



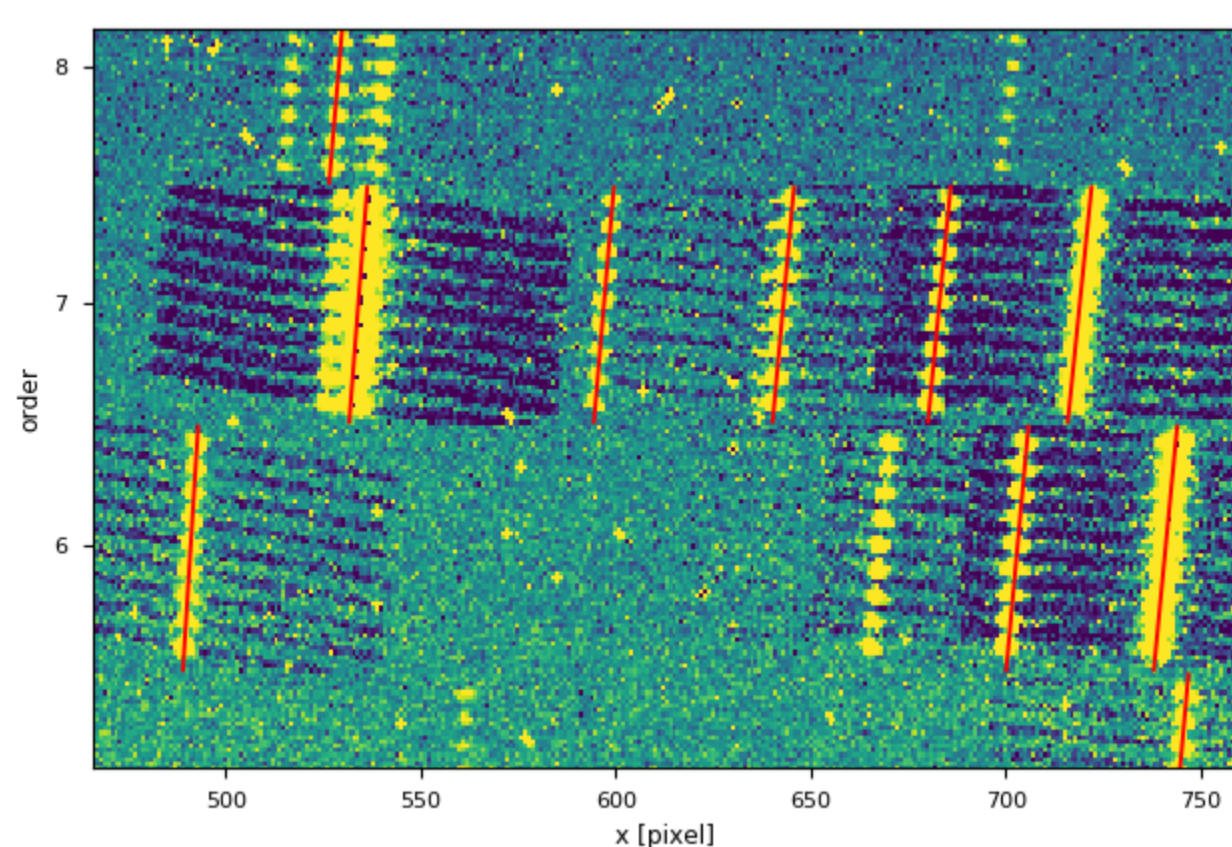
Parameters



PySME



Spectrum



A section of the input image from the X-shooter instrument at the VLT. Red lines mark the position of spectral lines. Credit: Data from VLT/X-Shooter

PyReduce - A Data Reduction Pipeline for Echelle Spectrographs

PyReduce is the new and improved version of REDUCE, now rewritten in Python. It creates wavelength calibrated and continuum normalized 1D spectra, based on raw observation images from various instruments. From high resolution instruments like HARPS, to low resolution observations with the JWST. This makes it the perfect companion tool to SME. Reduce your observations with PyREDUCE then analyse them with PySME.

References

[1] Asplund, Martin, Nicolas Grevesse, A. Jacques Sauval, and Pat Scott. 2009. "The Chemical Composition of the Sun." *Annual Review of Astronomy and Astrophysics* 47 (1): 481–522. <https://doi.org/10.1146/annurev.astro.46.060407.145222>.

[2] Buder, Sven, Martin Asplund, Ly Duong, Janez Kos, Karin Lind, Melissa K. Ness, Sanjib Sharma, et al. 2018. "The GALAH Survey: second data release." *Monthly Notices of the RAS* 478 (4): 4513–52. <https://doi.org/10.1093/mnras/sty1281>.

[3] Gustafsson, B., B. Edvardsson, K. Eriksson, U. G. Jørgensen, Nordlund, and B. Plez. 2008. "A grid of MARCS model atmospheres for late-type stars. I. Methods and general properties." *Astronomy and Astrophysics* 486 (3): 951–70. <https://doi.org/10.1051/0004-6361/200809724>.

[4] Kupka, F. G., T. A. Ryabchikova, N. E. Piskunov, H. C. Stempels, and W. W. Weiss. 2000. "VALD-2 – the New Vienna Atomic Line Database." *Baltic Astronomy* 9 (January): 590–94. <https://doi.org/10.1515/astro-2000-0420>.

[5] Kupka, F., N. Piskunov, T. A. Ryabchikova, H. C. Stempels, and W. W. Weiss. 1999. "VALD-2: Progress of the Vienna Atomic Line Database." *Astronomy and Astrophysics, Supplement* 138 (July): 119–33. <https://doi.org/10.1051/aas:1999267>.

[6] Loss, RD. 2003. "Atomic Weights of the Elements 2001 (Iupac Technical Report)." *Pure and Applied Chemistry* 75 (8). De Gruyter: 1107–22.

[7] Piskunov, N. E., F. Kupka, T. A. Ryabchikova, W. W. Weiss, and C. S. Jeffery. 1995. "VALD: The Vienna Atomic Line Data Base." *Astronomy and Astrophysics, Supplement* 112 (September): 525.

[8] Piskunov, Nikolai, and Jeff A. Valenti. 2017. "Spectroscopy Made Easy: Evolution." *Astronomy and Astrophysics* 597 (January): A16. <https://doi.org/10.1051/0004-6361/201629124>.

[9] Ryabchikova, T. A., N. E. Piskunov, F. Kupka, and W. W. Weiss. 1997. "The Vienna Atomic Line Database : Present State and Future Development." *Baltic Astronomy* 6 (March): 244–47. <https://doi.org/10.1515/astro-1997-0216>.

[10] Ryabchikova, T., N. Piskunov, R. L. Kurucz, H. C. Stempels, U. Heiter, Yu Pakhomov, and P. S. Barklem. 2015. "A major upgrade of the VALD database." *Physica Scripta* 90 (5): 054005. <https://doi.org/10.1088/0031-8949/90/5/054005>.

[11] Valenti, J. A., and N. Piskunov. 1996. "Spectroscopy made easy: A new tool for fitting observations with synthetic spectra." *Astronomy and Astrophysics, Supplement* 118 (September): 595–603.

