TSFitPy docs

N. Storm

January 25, 2024

Abstract

1 χ^2 calculation

1.1 Finding best fit

First we take the input spectra, and try to calculate χ^2 within the linemask. Hence:

$$lmin \leq \lambda_{i,O} \leq lmax,$$

where $lmin$ and $lmax$ are our input linemask values. $\lambda_{i,O}$ here represents the RV corrected wavelength values of the observed spectra with corresponding normalised flux O_i. By trying different inputs of our desired parameters $params$ (e.g. abundance), we calculate different S_i values (synthetic normalised flux values):

$$S_i = f(params),$$

where our $f()$ function is basically the code that generates synthetic code and $params$ are our input parameters. We interpolate our S_i values using linear interpolation at each wavelength point of the observed spectra $\lambda_{i,O}$, to be able to calculate our χ^2. Hence, it is important to have sufficiently small $\Delta\lambda_S$ for the synthetic spectra. Thus we calculate χ^2 using:

$$\chi^2 = \sum_{i=0}^{N} \left(\frac{O_i - S_i}{\sigma_i} \right)^2,$$

where O_i is observed spectra normalised flux values, S_i is synthetic spectra normalised flux values and σ_i is the observed spectra normalised flux error (sigma variation). Thus this χ^2 is calculated by summing it over all pixels. This is converted to χ^2_{red}:

$$\chi^2_{\text{red}} = \frac{\chi^2}{dof}$$
$$dof = i - \text{parameters count},$$

thus i is the amount of pixels in the linemask, where parameters is number of fitted parameters (usually assumed as 1, because we mainly change abundance for one element only). Individual σ_i for each pixel is provided in the spectra, assumed to be 0.01 if none given, or calculated from input SNR:

$$\sigma_i = \frac{1}{\text{SNR}}.$$

During the fitting process, we try to minimise χ^2_{red} until we reach desired $\chi^2_{\text{min, red}}$, depending on the requested tolerance of the minimisation method (usually Nelder-Mead).

1.2 Error of the fit

The error of the input *params* is found by calculating the offset of χ^2. In this case the user inputs the desired σ offset (different to σ_i, which is error for each pixel; here σ is the confidence interval). We use approximation:

$$\Delta\chi^2 \approx \sigma^2,$$

so for example if the desired confidence interval is 3 sigmas, then we try to find $\Delta\chi^2 = 3^2 = 9$. In the code, we actually use χ^2_{red}. Hence we are trying to find for which *params* we get the following χ^2:

$$\chi^2_{\text{error, red}} = \chi^2_{\text{min, red}} + \frac{\Delta\chi^2}{dof}.$$

I.e. we try to change *params* until we solve the root-problem:

$$\chi^2_{\text{new fit, red}} - \chi^2_{\text{error, red}} = 0,$$

by calculating $\chi^2_{\text{new fit, red}}$ at different input *params*.

References