PHEW

May 2024

1 Introduction

PHEW (PytHon Equivalent Widths) is a set of Python tools to calculate the equivalent width (EW), line depth, and full width at half maximum (FWHM) of a line for a given spectrum using the Python toolkit `PySpecKit`[1].

PHEW then estimates uncertainties via Monte Carlo (MC) simulations.

PHEW consists of three Python files:

1. EW.py: calculates EW and estimates uncertainty via MC method.
2. FWHM.py: calculates FWHM and estimates uncertainty via MC method.
3. linedepth.py: calculates line depth and estimates uncertainty via MC method.

2 Requirements

PHEW runs on Python 3, and it uses the following Python packages:

- `PySpecKit`
- `Astropy` (Warning: latest compatible version: 5.3.4)
- `Matplotlib` (Warning: latest compatible version: 3.8.4)
- `NumPy`
- `SciPy`

Please install these packages before running PHEW.

3 Known issues

The `PySpecKit` functionality to read FITS files does not always work as expected. We have found that the flux uncertainty dimension of a spectrum is never read or interpreted properly. Thus, when providing a filename as the input `spec`, be aware that the spectrum loaded will likely be missing flux uncertainties, even if they are available.

Furthermore, `PySpecKit` cannot recognize some FITS file formats at all. We therefore suggest to read the spectrum from the FITS file using some other tool and then use the spectrum array as the `spec` input. The folder `Example/` in the Gituhb repository contains a Python file called `readspec.py` that is capable of reading many different kinds of FITS file formats. Note, however, that `readspec.py` is far from foolproof. Please use with caution.

The annotation box of the Voigt fit in the middle panel (see Fig. 1) sometimes accidentally hides part of the fit. We are working to fix this issue so that the code can automatically find an optimal location of the annotation box.

[1] `https://pyspeckit.readthedocs.io/en/latest/`

4 EW.py

The python file EW.py contains the function called `equivalent_width()`, which the user invokes to perform the EW measurement.

4.1 Outputs

The function has two outputs:

- A PDF file with three panels (see Fig. 1). The top panel displays the full spectrum. The middle panel displays the spectral fit, with fitted parameters annotated. The bottom panel displays a histogram with the results of the MC routine, with the mean and standard deviation of the Gaussian distribution as the EW and its uncertainty. The 16th, 50th, and 84th percentiles are also indicated. This last panel is not drawn if the MC routine is skipped (see Fig. 2).
- A Python list containing the EW and its uncertainty. If the MC routine is skipped, then only a float number is returned with the preliminary EW measurement.

4.2 Inputs

Its input parameters are:

- `spec` - Numpy array, list or string. The user has the option to input either a FITS filename (as a string) or an array with the spectrum (as a Numpy array or a Python list). If a filename is given, it should contain its full path, unless the file is in the working directory. The code will attempt to read the spectrum from the FITS file using the `PySpecKit` functionality to read FITS files. We note that on most occasions, `PySpecKit` is unable to read or interpret the flux error dimension when available. This is a required input.

  If an array is given, it should contain the spectrum with wavelength, flux, and optional flux errors as columns.

- `bandloc` - Float or integer. It specifies the central location of the spectral line to be measured in Å. This is a required input.

- `xmin, xmax` - Integers. They specify the interval in wavelength space that defines the region of interest in Å. This is a required input.

- `exclude_min, exclude_max` - Integers. They specify the interval in wavelength space that binds the edges of the spectral feature itself, in Å. The pseudo-continuum will be defined using the spectral ranges [`xmin`, `exclude_min`] and [`exclude_max`, `xmax`]. This is a required input.

- `mc` - Boolean. This input determines whether to perform the MC iteration to estimate EW uncertainty. This parameter is relevant only in non-interactive mode (`interactive`=False). The default is True.

- `n` - Integer. It specifies the number of times the EW measurement is repeated in the MC iteration. The default is 1000.

- `fldr` - String. It specifies the location where the output figure is desired. The default is None, in which case the figure is saved in the working directory.

- `name` - String. It specifies the name of the object. This name will be used as the title of the figure, as well as the name of the output figure file, using the format `name`_EWfit.pdf. The default is None, in which case the object's name will be set to "Object".

- `speclims` - Numpy array or list. This array specifies the minimum and maximum wavelength values to be plotted in the top panel, which shows the full spectrum (see Fig. 1). Note that the fitting routine ignores this input. The default is None, in which case the full spectrum is plotted in the top panel.

- `outname` - String. This input gives the user the option of providing the code with a filename for the output figure besides `name`_EWfit.pdf. The default is None, in which case the output figure will be named using the input parameter `name` (see above). The full path can also be specified here, in which case `fldr` should not be set.
- `blorder` - Integer. It specifies the order of the polynomial used to fit the pseudo-continuum (baseline). The default is 1.
- `interactive` - Boolean. This input allows the user to run the code in interactive mode. In such a mode, the code pauses after performing the spectral fit and displays the fit in a Python window. The figure includes an annotation with a preliminary EW calculation. The code then asks the user if the fit is satisfactory. If the user replies yes ("y"), then the code continues running to perform the MC routine to calculate the EW and its uncertainty. If the user replies no ("n"), then the code stops and saves the figure without running the MC code.

  The default of this input parameter is True. Note that if `interactive` is set to True, then the input parameter `MC` (see above) is ignored.
- `clobber` - Boolean. It specifies whether to overwrite an existing figure file. The default is True.

5 FWHM.py

The Python file FWHM.py contains the function `measure_fwhm()`, which the user invokes to perform the FWHM measurement.

5.1 Outputs

The function has three outputs:

- The mean and standard deviation of the FWHM measured `n` times.
- The spectrum plotted with the the Voigt profile line fit (in blue), the pseudo-continuum (in yellow), and the FWHM (in blue).
- A histogram of the FWHM distribution.

5.2 Inputs

Its input parameters are:

- `filename` - String. It specifies the FITS filename, and it should contain its full path, unless the file is in the working directory. The code will attempt to read the spectrum from the FITS file using the `PySpecKit` functionality to read FITS files. We note that on most occasions, `PySpecKit` is unable to read or interpret the flux error dimension when available. This is a required input.
- `xmin, xmax` - Integers. They specify the interval in wavelength space that defines the region of interest. This is a required input.
- `exclude_min, exclude_max` - Integers. They specify the interval in wavelength space that binds the edges of the spectral feature itself. The pseudo-continuum will be defined using the spectral ranges [`xmin`, `exclude_min`] and [`exclude_max`, `xmax`]. This is a required input.
- `n` - Integer. It specifies the number of times the EW measurement is repeated in the MC iteration. This is a required input.

6 linedepth.py

The Python file linedepth.py contains the function `measure_line_depth()`, which the user invokes to perform the line depth measurement.

6.1 Outputs

The function has three outputs:

- The mean and standard deviation of the line depth measured `n` times.
- The spectrum plotted with the pseudo-continuum (in yellow).
- A histogram of the line depth distribution.

6.2 Inputs

Its input parameters are:

- `filename` - String. It specifies the FITS filename, and it should contain its full path, unless the file is in the working directory. The code will attempt to read the spectrum from the FITS file using the `PySpecKit` functionality to read FITS files. We note that on most occasions, `PySpecKit` is unable to read or interpret the flux error dimension when available. This is a required input.
- `xmin, xmax` - Integers. They specify the interval in wavelength space that defines the region of interest. This is a required input.
- `exclude_min, exclude_max` - Integers. They specify the interval in wavelength space that binds the edges of the spectral feature itself. The pseudo-continuum will be defined using the spectral ranges [`xmin`, `exclude_min`] and [`exclude_max`, `xmax`]. This is a required input.
- `n` - Integer. It specifies the number of times the EW measurement is repeated in the MC iteration. This is a required input.

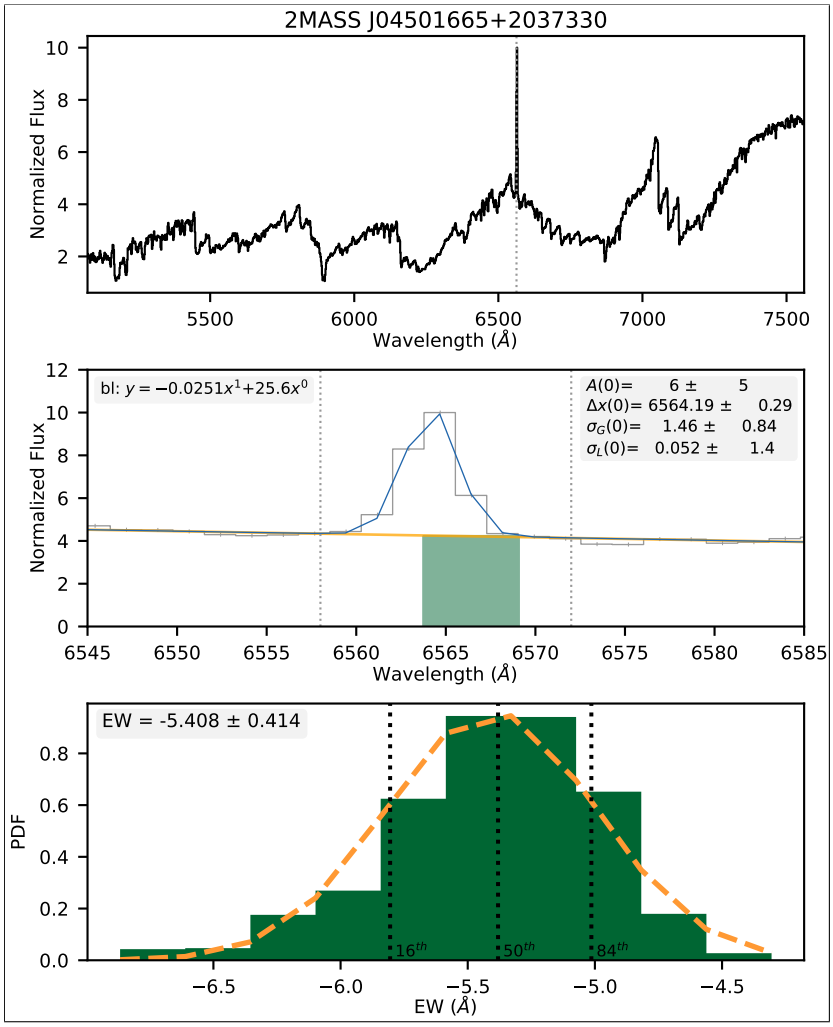

Figure 1: Output of the `EW.equivalent_width()` function. The title of the figure (here 2MASS J04501665 +2037330) can be specified with the input `name`. The top panel displays the full spectrum; the wavelength range of the top panel can be specified with the input `speclims`. The location of the line to be measured (with the input `bandloc`) is indicated with a vertical dotted line. The middle panel displays the fit of the spectral line. The upper left annotation shows the parameters of the pseudo-continuum fit (i.e. the baseline). The upper right annotation shows the parameters of the Voigt profile fit. In this example, `xmin`=6545, `xmax`=6585, `exclude_min`=6558, and `exclude_max`=6572. The latter two are indicated with vertical dotted lines. The bottom panel shows the histogram of the `n` measurements of the EW during the MC routine. The mean and standard deviation of the fitted Gaussian are adopted as the line's EW and EW uncertainty and are annotated in the upper left. The 16th, 50th, and 84th percentiles are indicated with vertical dotted lines.

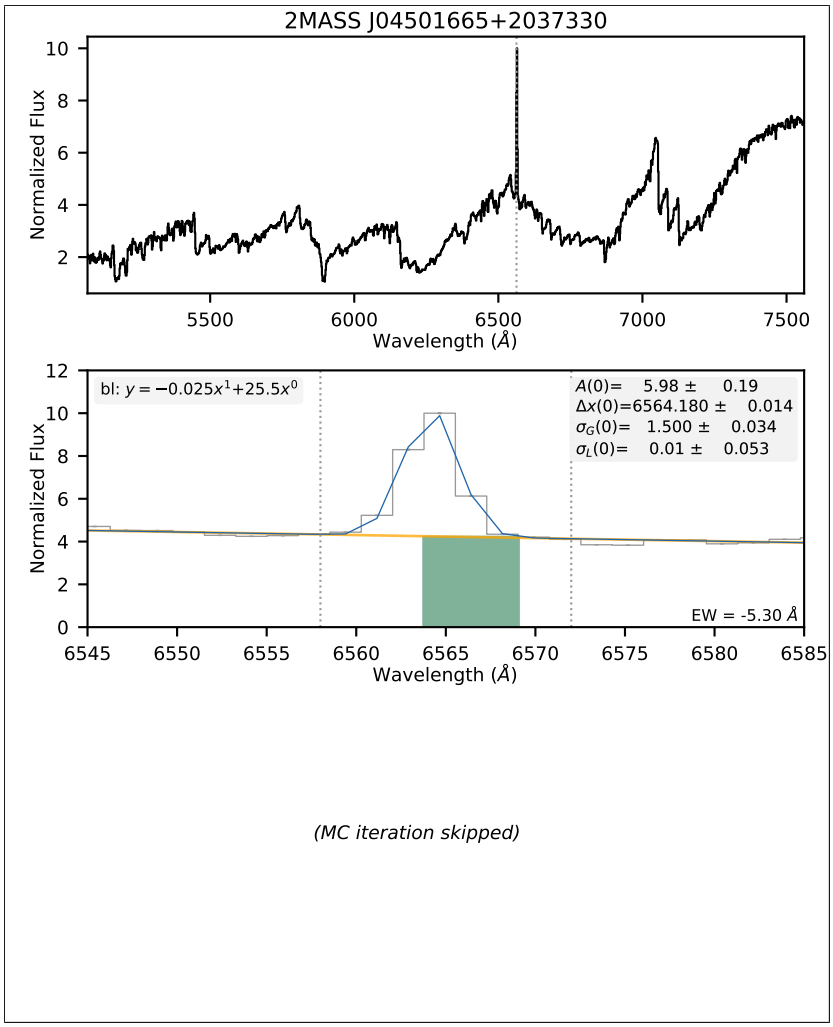

Figure 2: Same as Fig. 1, except that the MC routine was skipped. This can happen in interactive mode (`interactive`=True), when the user replies no to the question of whether to continue to the MC routine. It can also happen in non-interactive mode (`interactive`=False), when the input `MC` is set to False. In such cases, the bottom panel is not drawn, and the middle panel annotates at the bottom right the preliminary EW measurement from the fit.