

Article

Not peer-reviewed version

---

# PyMossFit: A Google Colab Option for Mössbauer Spectra Fitting

---

[Fabio Daniel Saccone](#) \*

Posted Date: 27 August 2025

doi: 10.20944/preprints202506.2153.v3

Keywords: Cloud Computing; Mössbauer Effect Spectroscopy; Condensed Matter Physics; Python



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

# PyMossFit: A Google Colab option for Mössbauer Spectra Fitting

Fabio D. Saccone <sup>1,2,3</sup>

<sup>1</sup> Departamento de Física, Facultad de Ingeniería, Universidad de Buenos Aires, Av. Paseo Colón 850, Buenos Aires. ARGENTINA

<sup>2</sup> YPF Tecnología S. A., Av. del Petróleo S/N, Berisso, ARGENTINA

<sup>3</sup> CONICET, Godoy Cruz 2290, Buenos Aires. ARGENTINA

\* Correspondence: fsaccone@fi.uba.ar

## Abstract

This article introduces the main characteristics of PyMossFit, a software for Mössbauer spectra fit. It is explained how each utility of their code sections works. Based on the `Lmf it` python package, it is a robust data fitting tool. Designed to run as a Jupyter notebook at the Google Colab cloud, it also allows us to work from multiple devices and operating systems. In addition, it facilitates the fact that the fitting procedure can be performed collaboratively by researchers. The software performs the folding of raw data with a discrete Fourier transform. Data smoothing is available with the use of a Savitzky-Golay algorithm. Likewise, a K-nearest neighbor algorithm helps us to determine the present phases by matching the correlations of hyperfine parameters from a local database.

**Keywords:** cloud computing; Mössbauer Spectroscopy; hyperfine interactions; condensed matter physics; chemical physics

## 1. Introduction

Mössbauer effect spectroscopy [1,2] is a highly specialized technique that investigates the resonant absorption of  $\gamma$  rays by atomic nuclei. The Mössbauer effect, observed primarily in isotopes such as iron-57 ( $^{57}\text{Fe}$ ), provides information on hyperfine interactions, including isomer shifts, quadrupole splittings, and magnetic hyperfine fields. These parameters offer detailed information on the electronic, magnetic and structural environment of the sample, making the technique invaluable in materials science, condensed matter physics, chemistry, biology [3], planetary science [4], and also in cosmology [5]. The gamma ray of resonant absorption in  $^{57}\text{Fe}$  nuclei is 14.4 keV and hyperfine interactions promote the break of degeneracy at ground and excited nuclear levels. The above cited parameters are described in the following subsections:

### 1.1. Isomer Shift, IS or $\delta$

This phenomenon corresponds to a certain probability that electrons have to be inside the nucleus [6], an exclusive quantum event. The next equation describes this parameter.

$$IS = C \cdot \frac{\delta R}{R} \cdot (|\psi_A(0)|^2 - |\psi_S(0)|^2) \quad (1)$$

where  $\psi_A(0)$ ,  $\psi_S(0)$  are the s-wavefunctions at the origin of the absorbent (A) and the  $\gamma$ -ray source (S) nucleus, while  $\frac{\delta R}{R}$  is the relative nuclear radius R change during recoilless  $\gamma$  resonant absorption.

### 1.2. Quadrupole Splitting, QS or $\Delta$

This quantum effect gives account of the interaction of nuclear charge with an electric field gradient (EFG) originated in an asymmetric electronic orbital distribution. The analytical expression is

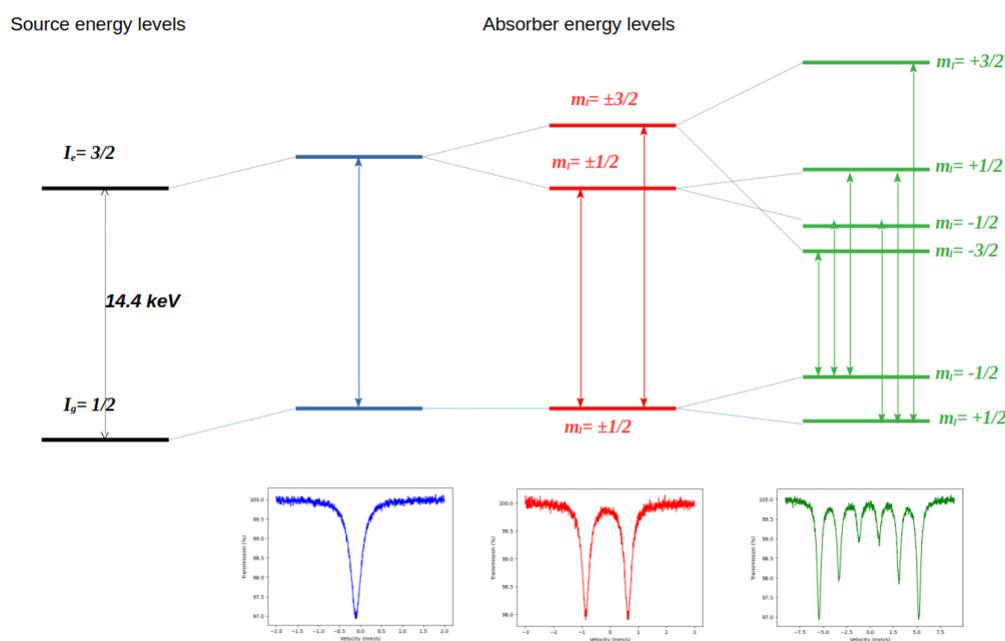
$$QS = \frac{1}{2}e \cdot Q \cdot V_{ZZ} \sqrt{\left(1 + \frac{1}{3} \cdot \eta\right)} \quad (2)$$

being  $Q$  the quadrupolar moment,  $V_{ZZ}$  the principal component of the EFG, and  $\eta$  the asymmetric factor.

### 1.3. Magnetic Hyperfine Field, $B_{HF}$

Nucleons have an intrinsic quantum property, the spin. When the spin is coupled with the quantum orbital moment, a total moment can interact with the local magnetic field induced by the electronic orbitals, giving rise to a more complex splitting (Zeeman effect) and, according to the nuclear selection rules with some forbidden transitions, and consequently, certain decays by  $\gamma$ -ray emission are not physically possible.

The Figure 1 shows the energy levels schemes of all of these hyperfine interactions for the  $^{57}\text{Fe}$  isotope. As can be observed, IS shifts the excited and ground energy levels between the source nucleus and the sample one; meanwhile, QS promotes the splitting of the excited level and  $B_{HF}$  the same in both the ground and excited levels, with the  $I_e = +3/2 \rightarrow I_g = -1/2$  and  $I_e = -3/2 \rightarrow I_g = +1/2$  forbidden transitions by the selection rules.



**Figure 1.** Energy levels schemes corresponding to Isomer Shift (IS, in blue), ii) Quadrupole Splitting (QS, in red) and iii) Hyperfine Field Splitting with all the allowed transitions (Magnetic Zeeman Effect Field,  $B_{HF}$ , in green).

In a typical Mössbauer experiment, a radioactive source emits gamma rays, which are absorbed by the nuclei in the sample (see Figure 2). Resonant absorption occurs, in a small fraction, when  $\gamma$  rays hit the probe with recoilless. The detector measures the intensity of the transmitted radiation as a function of the velocity of the gamma-ray source (with an additional energy added by Doppler effect). This results in a Mössbauer spectrum, typically characterized by sharp peaks or dips at resonance frequencies corresponding to different hyperfine interactions within the sample. The most common experimental setup corresponds to Transmission Geometry (in this case, the observed lines come from a reduction of 14.4 keV gamma counting in the detector, as compared to the background signal). Another possible option is the Backscattering Mössbauer Spectroscopy (BMS) which corresponds

to detection of back-scattered electrons after gamma absorption. In recent years, the use of synchrotron radiation was extended, helping to collect Mössbauer spectra with deeper details and in a short time [7].

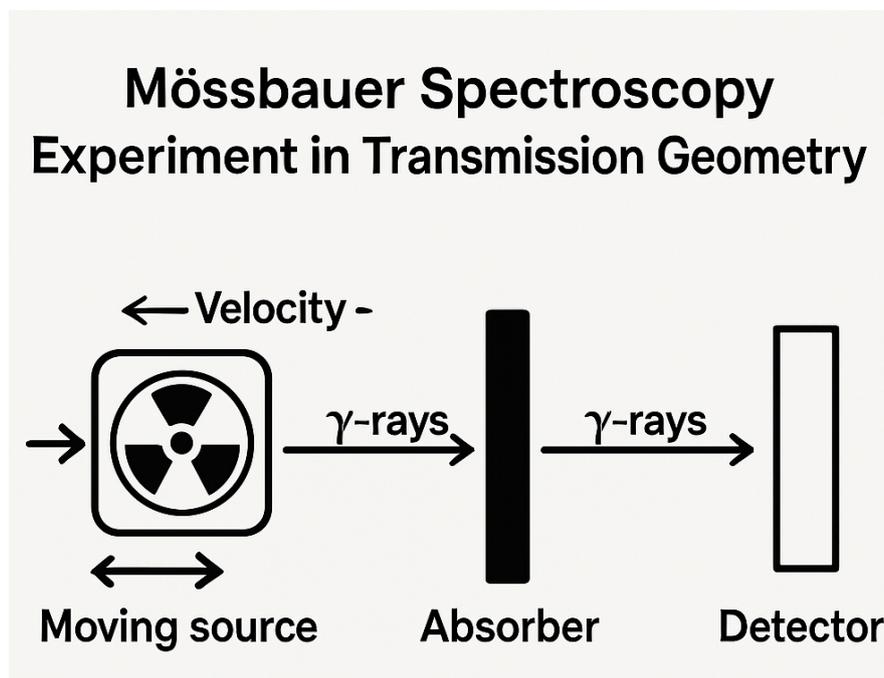


Figure 2. Scheme of a typical Mössbauer experiment in transmission geometry

#### 1.4. Statement of Need

Recently, in a review by Grandjean et al. [8], the authors made a series of suggestions about how good measurements should be taken, which could be a good practice for Mössbauer data treatment and its corresponding fitting presentation. Usually, scientists working with Mössbauer spectroscopy manage their data and fit them with non-open source software that runs on a Windows OS with poor upgrade services [9]. The needs of an open source option with less OS or package dependency has motivated this work.

In this sense, Google Colab is a useful tool for a collaborative team job in data analysis with the advantage of no additional packages locally installed, also compatible with the fact that the user can run their codes from multiple devices.

## 2. Mössbauer Spectra and Curve Shapes

The shape of a Mössbauer spectrum varies depending on the nature of the hyperfine interactions in the sample. For example, a simple paramagnetic material might produce a single absorption peak (Lorentzian or Gaussian) due to the isomer shift. Materials with quadrupole splitting generate a doublet (two peaks), while materials experiencing magnetic hyperfine interactions often exhibit more complex sextet patterns. These spectral features often overlap, making it difficult to isolate and quantify each individual component.

### 2.1. Curve Fitting and Least Squares Method

Curve fitting plays a crucial role in Mössbauer spectral analysis, as it allows for the decomposition of these complex spectra into individual contributions, each associated with specific hyperfine parameters. The challenge lies in accurately reproducing the spectral shapes using mathematical models and adjusting the parameters until a satisfactory fit is achieved.

To accurately extract physical parameters from Mössbauer spectra, fitting procedures are applied to the experimental data. One of the most common approaches is the least squares method, which minimizes the difference between the experimental data points and the theoretical model curve. The

objective is to adjust the parameters of the theoretical model (such as isomer shift, quadrupole splitting, and line broadening) so that the calculated spectrum fits the experimental data as closely as possible. Then, a minimization of the  $\chi^2$  is defined as:

$$\chi^2 = \sum_{i=1}^N \frac{(y_i^{exp} - y_i^{model}(\vec{p}))^2}{\epsilon_i^2} \quad (3)$$

being  $\vec{p}$  a set of selected parameters for fitting, which minimizes the distance between the experimental data,  $y_i^{exp}$  and the modeled data points  $y_i^{model}$ .

In Python, this process can be implemented using libraries like Lmfit [10], SciPy [11], or NumPy [12], which provide robust tools for processing data, graphics, and least-squares curve fit. The general approach involves defining a model function that represents the expected shape of the Mössbauer spectrum, which could be a sum of multiple Lorentzian or Gaussian functions, depending on the number and type of spectral components. Lorentzian functions are preferred with crystalline samples, whereas PseudoVoigts (sum of Lorentzian and Gaussian functions) are appropriate for Fe sites in disordered materials.

The least-squares fitting algorithm iteratively adjusts the model parameters until the sum of squared residuals between the experimental and calculated spectra is minimized. An example of a Lorentzian function definition for Python, can be seen next,

```
from scipy.optimize import curve_fit
import numpy as np

# Define the model function (e.g., a sum of Lorentzian components)
def lorentzian(x, amplitude, center, width):
    return amplitude * (width**2 / ((x - center)**2 + width**2))

# Generate synthetic data for fitting
x_data = np.linspace(-10, 10, 500) # Example velocity range
y_data = lorentzian(x_data, 1, 0, 1) + np.random.normal(0, 0.02,
    size=len(x_data))

# Perform least-squares curve fitting
initial_guess = [1, 0, 1] # Initial guess for parameters
params, covariance = curve_fit(lorentzian, x_data, y_data,
    p0=initial_guess)

# params contains the optimized parameters: amplitude, center,
    and width
```

This example demonstrates fitting a single Lorentzian function to synthetic data, though more complex models involving sums of Lorentzians or other spectral shapes can be implemented to represent real Mössbauer spectra.

### 3. Description of the Python code for Google Colab (PyMossFit)

The Python code is available in several Jupyter notebooks as typical examples found in practice [13]. In their names, some text is added that indicates the number and type of interactions (i. e. 1Q means one quadrupolar interaction, 2X for two magnetic sites, and so on). Some selected parts of it are described throughout this page.

The code is structured in three cells. The first includes the installation in the Google Colab environment of Lmfit, the core of data fitting. Likewise, it imports some packages of Scipy, Pandas,

Matplotlib and Numpy, among others. The next step includes a Drive connection that asks the user for permission.

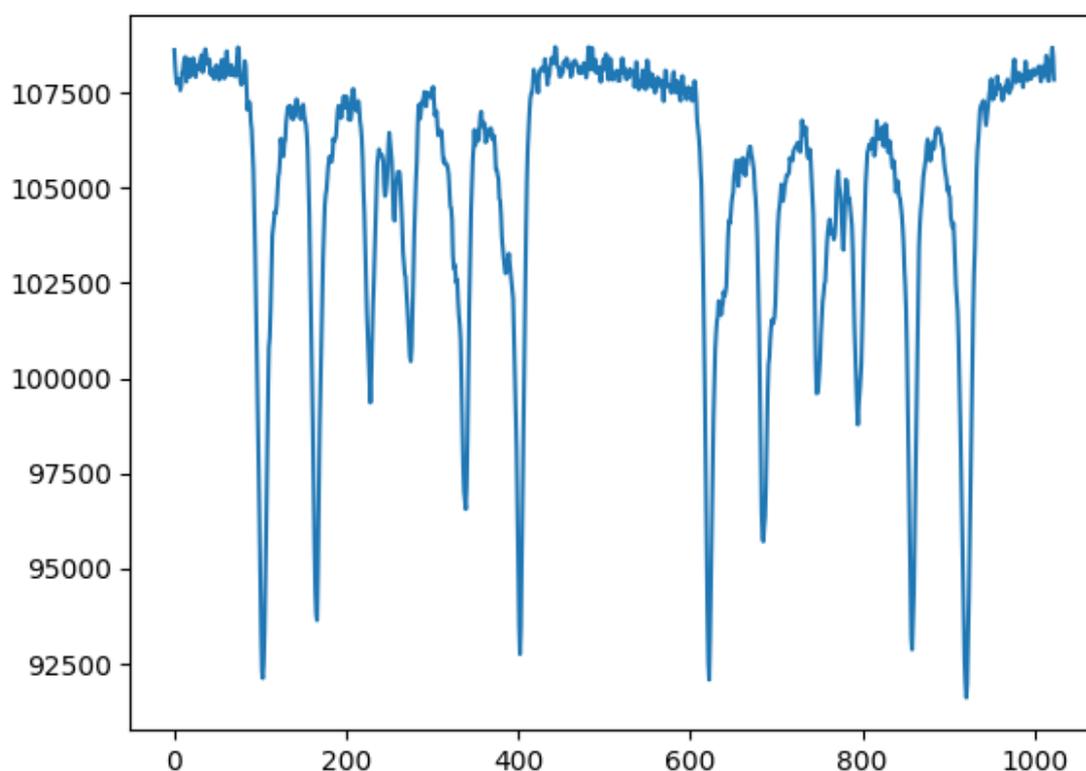
The second cell reads the datafile (format should be inspected previously to define "delimiter", "columns" and "skiprows" parameters). The required inputs are date (in a YYYYMMDD format) and maximum velocity associated to the extreme channels, in order to define the velocity scale.

In this cell, spectrum folding is performed with a Discrete Fourier Transforming (DFT) routine, the Numpy `fft`. The theory of this procedure corresponds to the Nyquist-Shannon Sampling Theorem that helps to determine a folding channel from the symmetry of discrete Fourier spectra [14].

Data can also be smoothed using a Savitsky-Golay package (I use `savgol` from `Scipy`). After folding, a new datafile is saved for fit with the use of the next cell.

The next figures illustrate the partial plots of data management with this procedure. This spectrum corresponds to zinc-ferrite nanoparticles obtained by a wet chemical method [15], which present a spinel structure with a tetrahedral site for  $Fe^{3+}$  and a octahedral site for both,  $Fe^{2+}$  and  $Fe^{3+}$ , cations.

When raw data are plotted, they look as shown in Figure 3 the following because of the collection of absorption resonances converted to signals in a single-channel analyzer output, while the source moves back and forth. Figures 4 and 5 show the DFT plot and the final folded spectrum, respectively. In the final folded spectrum, the horizontal sequence of channels was transformed to a  $mm/s$  scale based on a calibration with a metallic  $^{57}Fe$  thin foil sample.



**Figure 3.** Unfolded plot of a raw Mossbauer experiment data

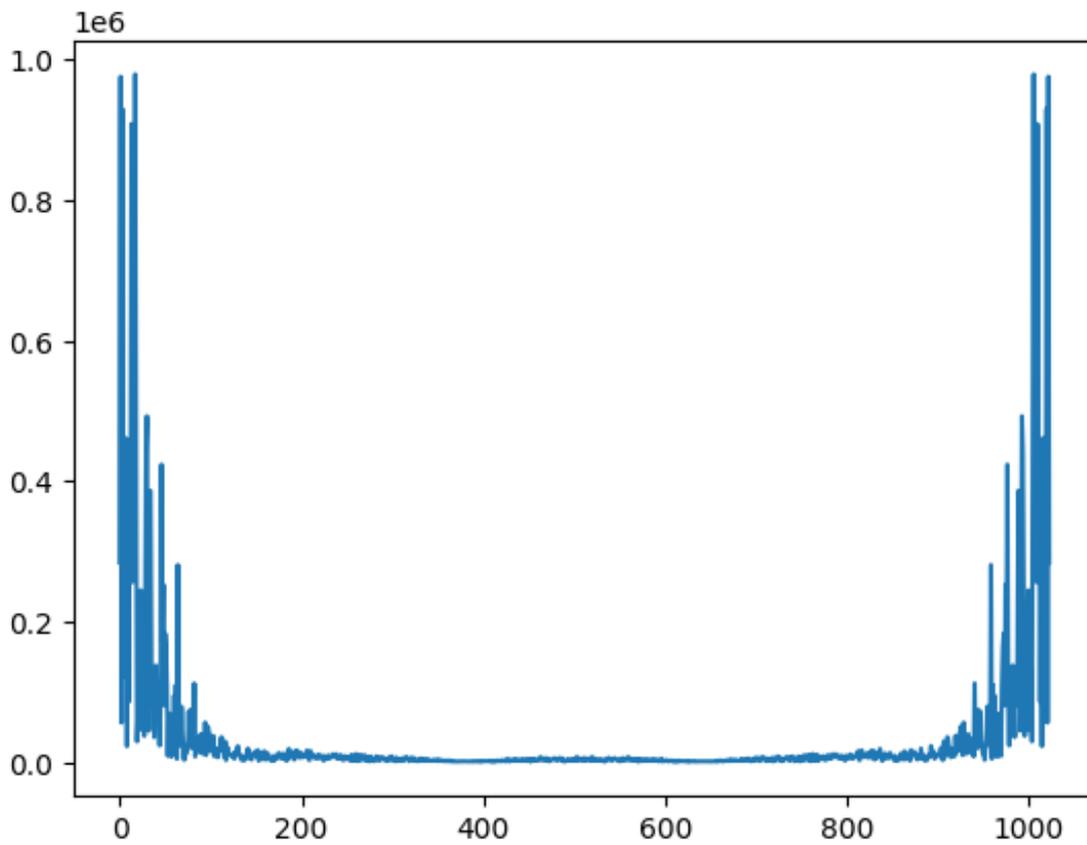


Figure 4. Discrete Fourier Transform of the raw Mossbauer experiment data plotted in the previous figure

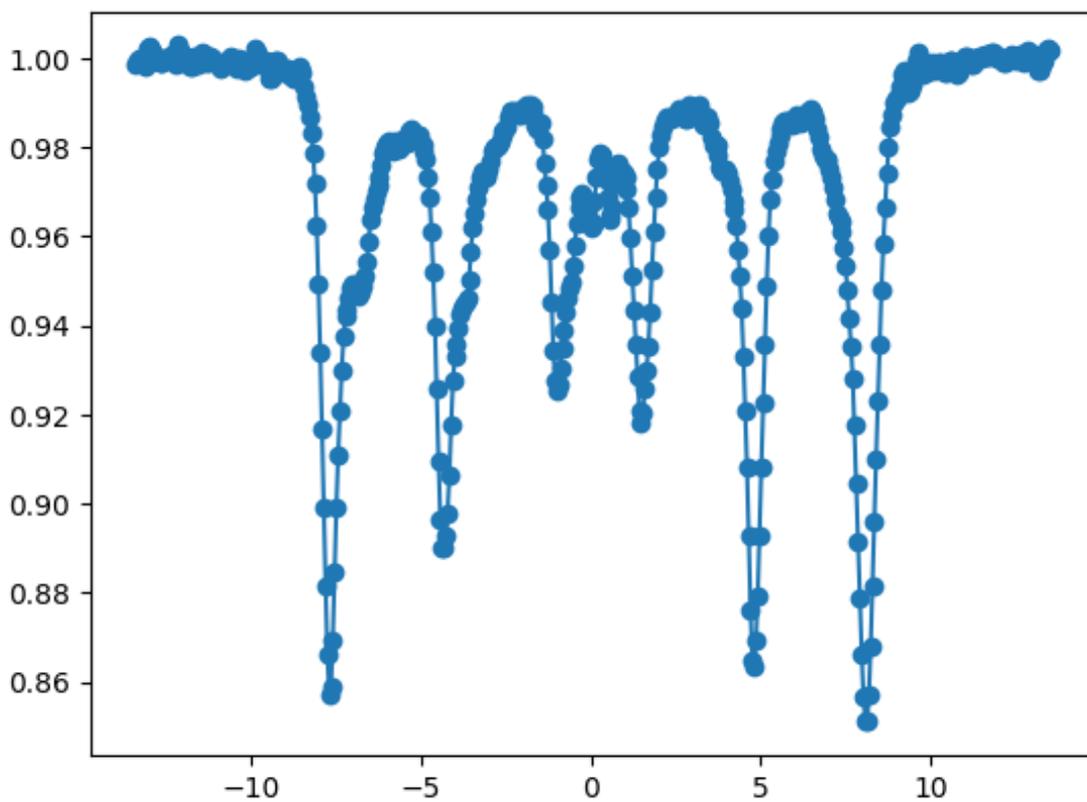
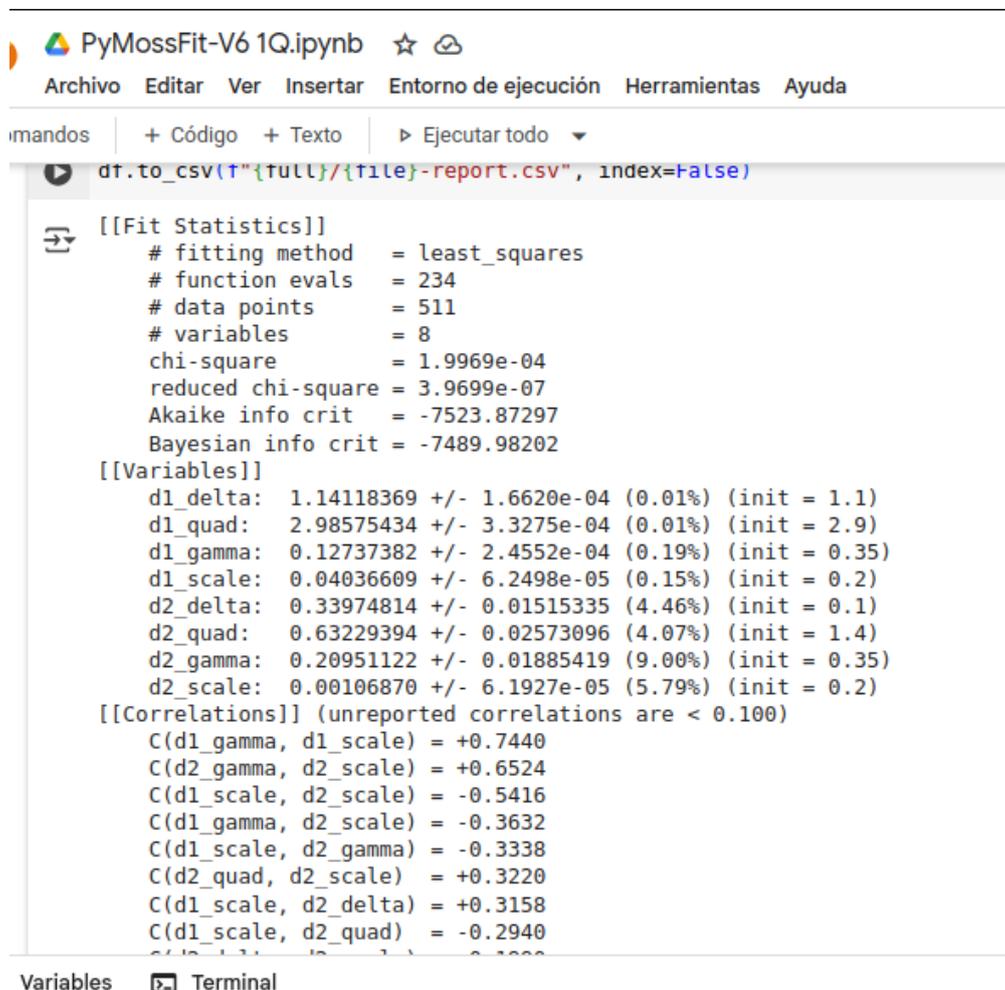


Figure 5. Final folded spectrum

The set of physical parameters such as linewidth ( $\Gamma$ ), isomer shifts (IS), quadrupole splitting (QS) and magnetic hyperfine field ( $B_{HF}$ ) are calculated after adjusting the amplitude (a), full width at half maximum (b), centroid (m), line shift (d) and line separation (q). The initial set of these parameters can be fitted or fixed by selecting the "True" or "False" options, respectively.

A typical fit report of the output of this cell looks as follows:



```
df.to_csv(f"{full}/{file}-report.csv", index=False)

[[Fit Statistics]]
# fitting method      = least_squares
# function evals      = 234
# data points         = 511
# variables            = 8
chi-square            = 1.9969e-04
reduced chi-square    = 3.9699e-07
Akaike info crit     = -7523.87297
Bayesian info crit   = -7489.98202

[[Variables]]
d1_delta:  1.14118369 +/- 1.6620e-04 (0.01%) (init = 1.1)
d1_quad:   2.98575434 +/- 3.3275e-04 (0.01%) (init = 2.9)
d1_gamma:  0.12737382 +/- 2.4552e-04 (0.19%) (init = 0.35)
d1_scale:  0.04036609 +/- 6.2498e-05 (0.15%) (init = 0.2)
d2_delta:  0.33974814 +/- 0.01515335 (4.46%) (init = 0.1)
d2_quad:   0.63229394 +/- 0.02573096 (4.07%) (init = 1.4)
d2_gamma:  0.20951122 +/- 0.01885419 (9.00%) (init = 0.35)
d2_scale:  0.00106870 +/- 6.1927e-05 (5.79%) (init = 0.2)

[[Correlations]] (unreported correlations are < 0.100)
C(d1_gamma, d1_scale) = +0.7440
C(d2_gamma, d2_scale) = +0.6524
C(d1_scale, d2_scale) = -0.5416
C(d1_gamma, d2_scale) = -0.3632
C(d1_scale, d2_gamma) = -0.3338
C(d2_quad, d2_scale) = +0.3220
C(d1_scale, d2_delta) = +0.3158
C(d1_scale, d2_quad) = -0.2940
```

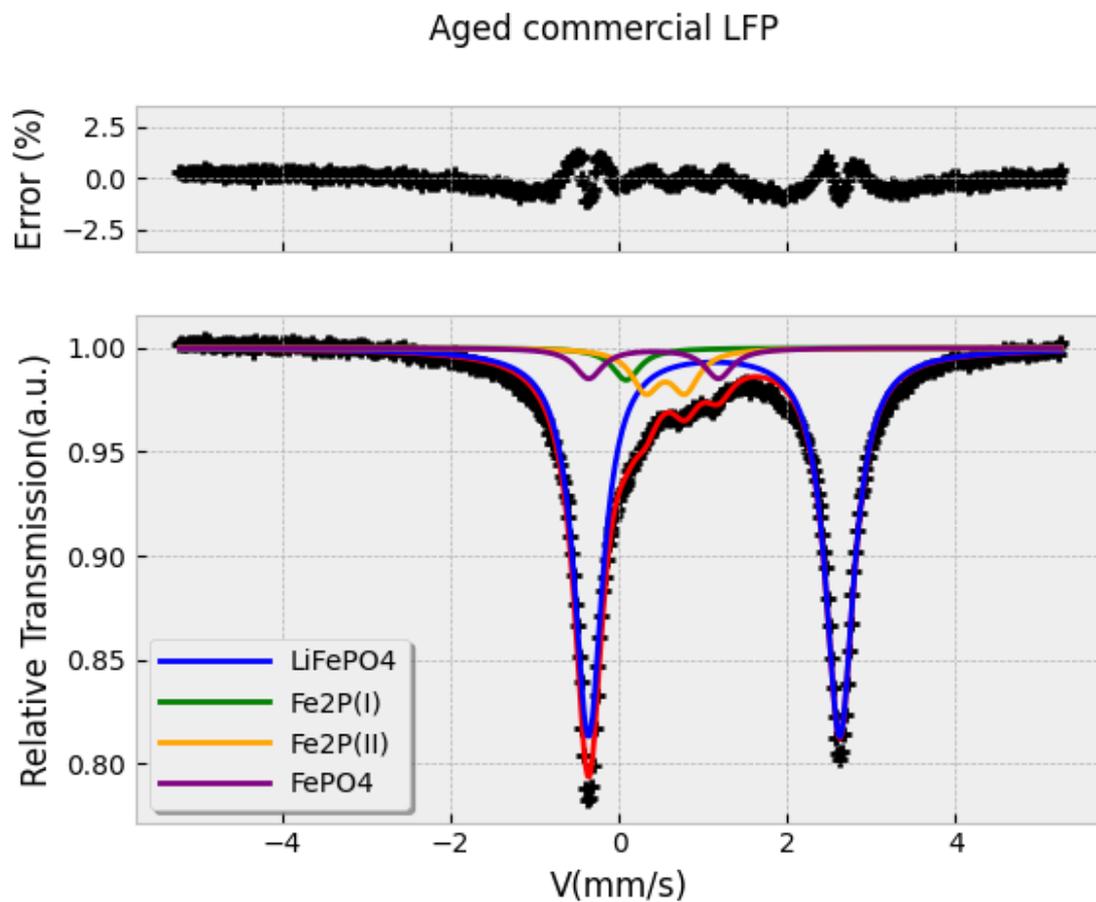
Figure 6. An example of a fit report output of a Jupyter notebook in Google Colab

Finally, the code saves fitted parameters, experimental and modeled sub-spectra in CSV formatted output files.

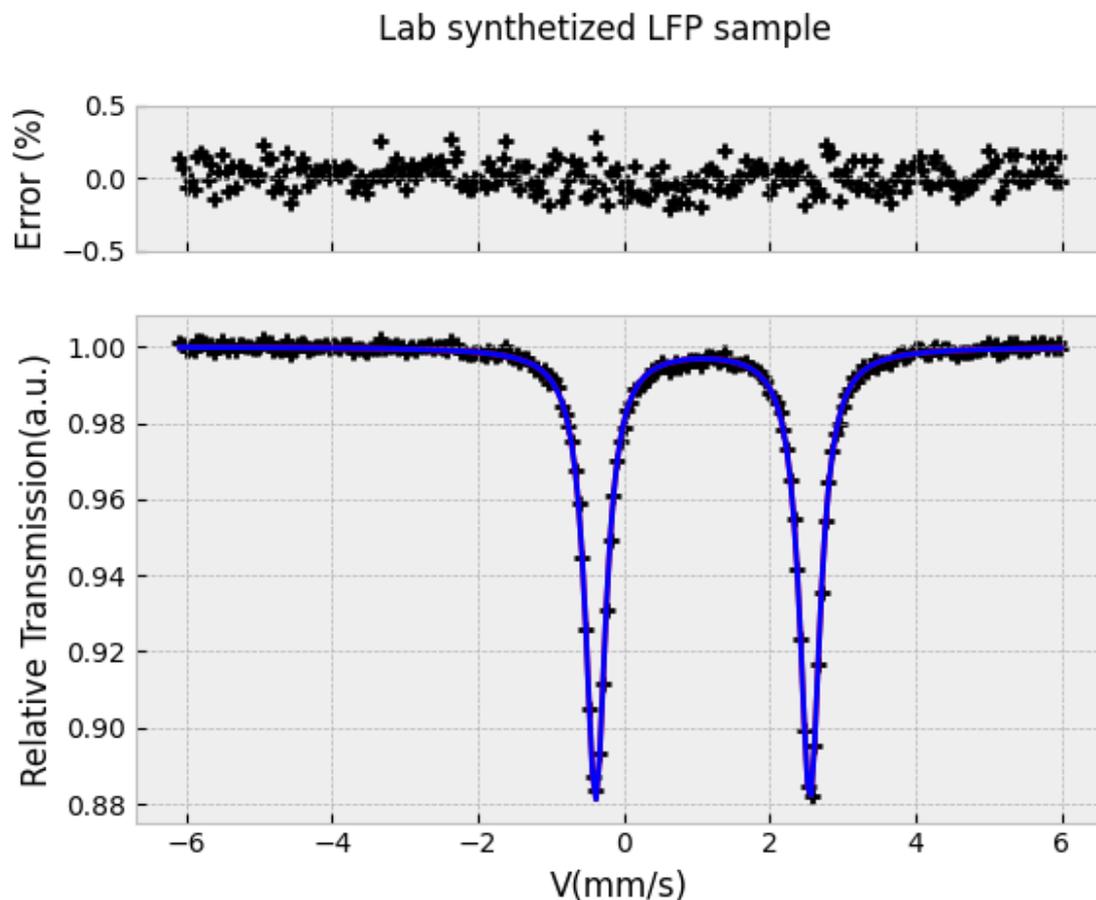
### 3.1. Some Examples

The following figures illustrate the fit results obtained from the Mössbauer spectra of different samples. Each exhibits several characteristics corresponding to the short-range-order neighborhood of  $^{57}\text{Fe}$  probes. Figure 7 corresponds to the fitted spectrum of an aged cathodic material  $\text{LiFePO}_4$  (LFP) where iron-rich secondary phases can clearly be identified. These undesired phases, which are responsible for the lowering of storage capacity, precipitated after extended thermal and environmental degradation. The fitting procedure took less than 40 secs after an appropriate selection of range parameters.

In contrast, Figure 8 shows a spectrum of a pure LFP sample which was synthesized by a solid-state reaction from analytical grade precursors.

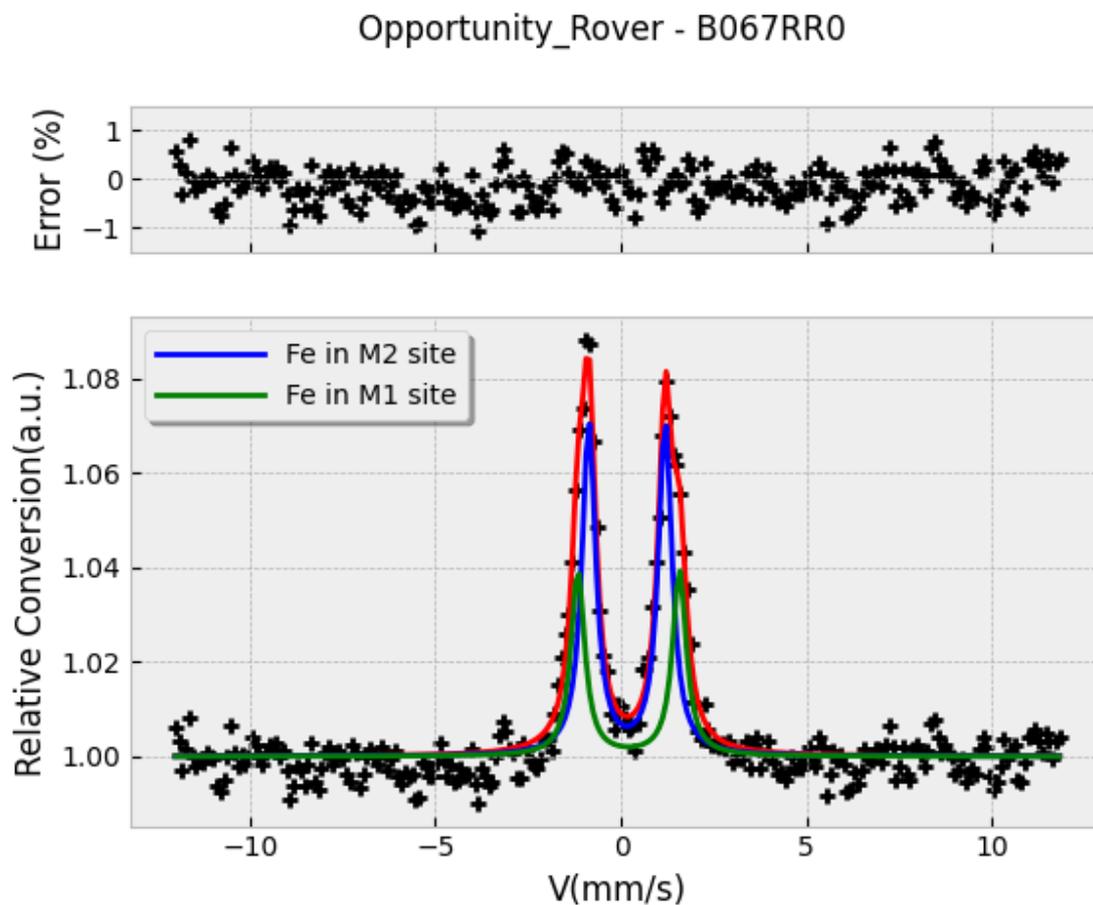


**Figure 7.** Mössbauer spectrum of a degraded commercial  $\text{LiFePO}_4$  active material for batteries. Residual phases were detected, such as  $\text{FePO}_4$  and  $\text{Fe}_2\text{P}$ . This last phase, shows the presence of  $^{57}\text{Fe}$  in two different oxidation states,  $\text{Fe(I)}$  and  $\text{Fe(II)}$ , respectively



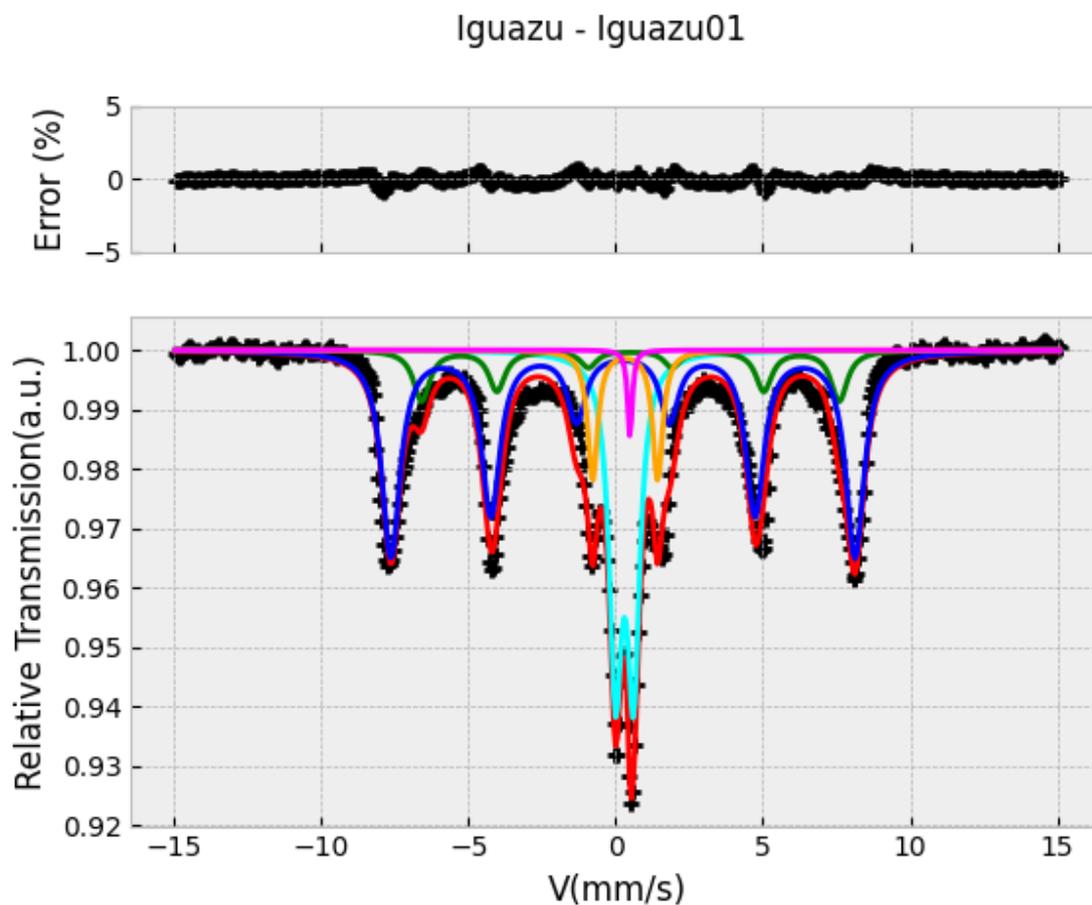
**Figure 8.** Mössbauer spectrum of the  $\text{LiFePO}_4$  active material for batteries, after synthesis at lab scale. In this case, just  $\text{LiFePO}_4$  was detected, besides the extraordinary sensitivity of the characterization technique

The Figure 9 corresponds to the fitted spectrum with PyMossFit of a soil sample taken by the Mössbauer spectrometer on board the Mars Opportunity Rover. The data is available from [16] and this set is named the "Bounce Rock Case" (B0RR067). The iron-rich detected phase was identified with the code and corresponds to the pyroxene phase  $((\text{Ca}, \text{Mg}, \text{Fe})\text{SiO}_3)$ , with two different octahedral Fe sites, M1 and M2 respectively [17] [18].



**Figure 9.** BMS spectrum of pyroxene in Mars soil (Opportunity Mission, Sept. 2004,  $T= 240\text{-}260\text{K}$ ). The subspectra correspond to two different  $^{57}\text{Fe}$  sites, M1 and M2, typically found in pyroxene.

The final example of the use of the PyMossFit notebook is the observed in Figure 10 and is the spectrum of an Iguazú fall soil sample. It corresponds to a complex spectrum where the use of a combination of sextets, doublets, and a singlet was required. All these subspectra correspond to different characteristics of iron oxides and hydroxides present in this kind of soils, originating from the erosion of basalt rocks.



**Figure 10.**  $^{57}\text{Fe}$  Mössbauer spectra of an Iguazú falls soil sample (Argentina-Brazil border)

### 3.2. Match of Fitted Parameters with a Local Database

A final cell allows us to identify the phases from an user's own database or a public ones [19]. This procedure is based on the minimization of Eulerian distances of the set of parameters with the bibliographic collection, which can be updated or extended by the user. Some lines that interpret ranges of hyperfine parameters from the database are also included. The phases cited in the examples above were identified or verified using this code section.

The machine learning algorithm used to guess the present phases is based on a K-Nearest Neighbors (`sklearn.neighbors`) [20]. The output is a recommendation with a ranking of suggested phases that match the best.

```

is_ref = df_ref.iloc[pos]['IS (mm/s)']
qs_ref = df_ref.iloc[pos]['Quad Splitting (mm/s)']
bhf_ref = df_ref.iloc[pos]['Bhf (T)']
print(f" Match {j+1}: {compound} ({{formula}})")
print(f"      IS: {is_ref:.2f} mm/s | QS: {qs_ref:.2f} mm/s")
print(f"      Distancia euclidiana: {d:.2f}\n")

print("## Usar este resultado a modo orientativo. Se recomienda co

Drive already mounted at /content/drive; to attempt to forcibly re

Muestra experimental 1:
Match 1: Fe(CO)5 (Fe(CO)5)
      IS: 0.25 mm/s | QS: 0.05 mm/s | Bhf: 0.0 T
      Distancia euclidiana: 2.11

Match 2: Fe(phen)2(NCS)2 ([Fe(phen)2(NCS)2])
      IS: 0.05 mm/s | QS: 0.05 mm/s | Bhf: 0.0 T
      Distancia euclidiana: 2.11

Match 3: Fe4[Fe(CN)6]3·xH2O (Fe4[Fe(CN)6]3·xH2O)
      IS: 0.05 mm/s | QS: 0.10 mm/s | Bhf: 0.0 T
      Distancia euclidiana: 2.16

Match 4: Prussian Blue (Fe4[Fe(CN)6]3)
      IS: 0.05 mm/s | QS: 0.10 mm/s | Bhf: 0.0 T
      Distancia euclidiana: 2.16

Match 5: Fe3[Fe(CN)6]2 (Fe3[Fe(CN)6]2)

```

Figure 11. A match report obtained with a KNN algorithm for phase identification

## 4. Conclusions

PyMossFit is a tool for Mössbauer spectroscopy analysis that explodes cloud computing. It was coded in Python and run in Jupyter notebook. The code requires the installation of the Lmfitt package as the core of model fitting. PyMossFit was designed to work in Python environment at cloud computing services, such as Google Colab, being a good choice for user update, free OS and device dependencies. Likewise, it simplifies the collaborative work during spectra analysis with the addition of a code section for phase identification based on a KNN algorithm.

**Acknowledgments:** The author thanks C. P. Ramos from Mössbauer group of CNEA and the INTECIN(UBA-CONICET) team for sample measurement.

## References

1. Wikipedia contributors. Mössbauer spectroscopy — Wikipedia, The Free Encyclopedia, 2025. [Online; accessed 19-June-2025].
2. Tao, W.; Junhu, Z. The Mössbauer Effect Data Center.
3. Martinho, M.; Münck, E., 57Fe Mössbauer Spectroscopy in Chemistry and Biology. In *Physical Inorganic Chemistry*; John Wiley and Sons, Ltd, 2010; chapter 2, pp. 39–67, [https://onlinelibrary.wiley.com/doi/pdf/10.1002/9780470602539.ch2]. https://doi.org/https://doi.org/10.1002/9780470602539.ch2.
4. Dyar, M.D.; Agresti, D.G.; Schaefer, M.W.; Grant, C.A.; Sklute, E.C. MÖSSBAUER SPECTROSCOPY OF EARTH AND PLANETARY MATERIALS. *Annual Review of Earth and Planetary Sciences* **2006**, *34*, 83–125. https://doi.org/https://doi.org/10.1146/annurev.earth.34.031405.125049.

5. Gao, Y.; Xu, W.; Zhang, H. A Mössbauer scheme to probe gravitational waves. *Science Bulletin* **2024**, *69*, 2795–2798. <https://doi.org/https://doi.org/10.1016/j.scib.2024.07.038>.
6. Walker, L.R.; Wertheim, G.K.; Jaccarino, V. Interpretation of the Fe<sup>57</sup> Isomer Shift. *Phys. Rev. Lett.* **1961**, *6*, 98–101. <https://doi.org/10.1103/PhysRevLett.6.98>.
7. Yi-Long, C.; De-Ping, Y., Synchrotron Mössbauer Spectroscopy. In *Mössbauer Effect in Lattice Dynamics*; John Wiley and Sons, Ltd, 2007; chapter 7, pp. 253–304, [<https://onlinelibrary.wiley.com/doi/pdf/10.1002/9783527611423.ch7>]. <https://doi.org/https://doi.org/10.1002/9783527611423.ch7>.
8. Grandjean, F.; Long, G.J. Best Practices and Protocols in Mössbauer Spectroscopy. *Chemistry of Materials* **2021**, *33*.
9. Lagarec, K.; Rancourt, D.G. RECOIL, Mössbauer Spectral Analysis Software for Windows (Version 1.0), 1998.
10. Newville, M.e.a. LMFIT: Non-Linear Least-Squares Minimization and Curve-Fitting for Python (version 1.3.4).
11. Virtanen, P.; Gommers, R.; Oliphant, T.E.; Haberland, M.; Reddy, T.; Cournapeau, D.; Burovski, E.; Peterson, P.; Weckesser, W.; Bright, J.; et al. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods* **2020**, *17*, 261–272. <https://doi.org/10.1038/s41592-019-0686-2>.
12. Harris, C.R.; Millman, K.J.; van der Walt, S.J.; Gommers, R.; Virtanen, P.; Cournapeau, D.; Wieser, E.; Taylor, J.; Berg, S.; Smith, N.J.; et al. Array programming with NumPy. *Nature* **2020**, *585*, 357–362. <https://doi.org/10.1038/s41586-020-2649-2>.
13. Saccone, F.D. Fitting code for Mössbauer Spectroscopy running in Jupyter Colab, 2025.
14. Kong, Q.; Siau, T.; Bayen, A. Discrete Fourier Transform (DFT), 2020.
15. Ferrari, S.; Apehesteguy, J.C.; Saccone, F.D. Structural and magnetic properties of Zn doped magnetite nanoparticles obtained by wet chemical method. *IEEE Transactions on Magnetics* **2015**, *51*.
16. Opportunity (MERB) Analyst's Notebook, 2025.
17. Corke, P.; Haviland, J. Mössbauer mineralogy of rock, soil, and dust at Meridiani Planum, Mars: Opportunity's journey across sulfate-rich outcrop, basaltic sand and dust, and hematite lag deposits. *Journal of Geophysical Research: Planets* **2006**, *111*.
18. Oshtrakh, M.I.; Petrova, E.; Grokhovsky, V. Determination of quadrupole splitting for <sup>57</sup>Fe in M1 and M2 sites of both olivine and pyroxene in ordinary chondrites using Mössbauer spectroscopy with high velocity resolution. *Hyperfine Interactions* **2007**, *177*.
19. Tao, W.; Junhu, Z. Mössbauer Effect Data Reference Journal.
20. Cover, T.; Hart, P. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory* **1967**, *13*, 21–27. <https://doi.org/10.1109/TIT.1967.1053964>.

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.