Atelier : Python pour l'analyse des données haute pression

Alexis Forestier, Antoine Hilberer, Yiuri Garino, Silvia Boccato

cea

cnrs

The workshop

- **A short introduction about Python**

- **Demonstration on some data analysis examples (Hands-on)**
  - Ruby luminescence fit
  - Black body radiation fit and processing
  - hdf5 XRD image plate processing

Resources in:
https://alexisforestier.github.io/Atelier-Python-HP/

Atelier-Python-HP

cea cnrs

Atelier "Python pour l'analyse des données haute pression"

14e Forum de technologie des hautes pressions

Nouvelles frontières en haute pression : de l'instrumentation à l'analyse de données

3-7 juin 2024 - Argelès-sur-Mer

Cet atelier propose une initiation à l'analyse de données en utilisant le langage de programmation Python, adapté au contexte des expériences haute pression. L'objectif est de montrer un aperçu des vastes possibilités et de fournir des informations utiles à la fois pour les débutants et les utilisateurs confirmés, sans se focaliser sur la syntaxe du langage. Python bénéficie d'une grande popularité et est très utilisé dans la gestion et l'analyse de données, ainsi que pour le pilotage des expériences sur grands instruments (ESRF, Soleil…). Il figure désormais parmi les langages les plus enseignés aux étudiants en sciences.

Ressources

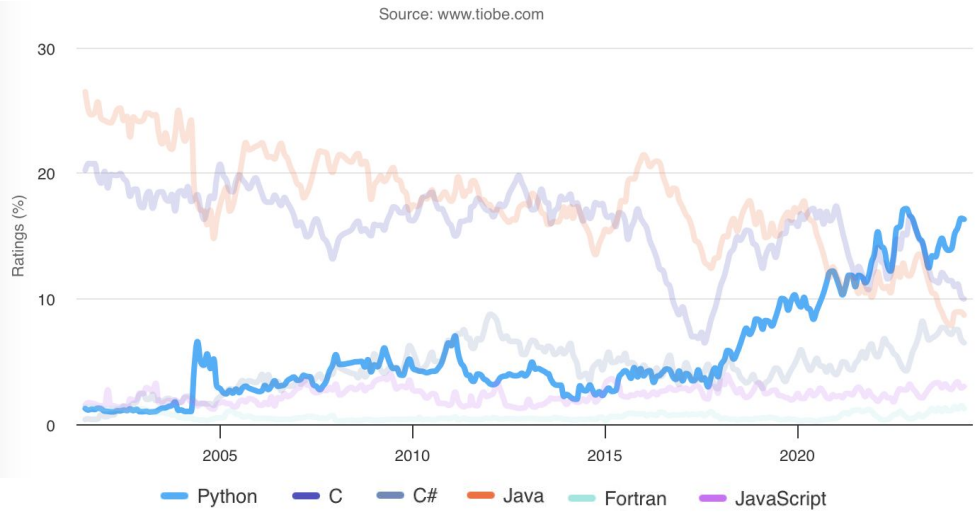
- Slides de présentation.
- Tutoriel : Installation de python, de packages additionnels, et utilisation des environnements virtuels
- Fit d'un spectre de luminescence du rubis. Visualiser Télécharger
- Analyse d'un spectre de rayonnement de corps noir. Visualiser Télécharger
- Ouverture et intégration azimuthale d'une plaque image de diffraction des rayons X au format HDF5 (.h5) Visualiser Télécharger

Autres ressources utiles

Bases

- Un très bon notebook pour apprendre les bases de python de manière interactive
- Les bases de python : boucles for/while, tests if else, fonctions et opérateurs, types natifs, listes, tuples et dictionnaires etc. : sur courspython.com
- Manipulation de tableaux numpy : doc. de numpy
- Indexation des tableaux numpy : dans la documentation de numpy, voir aussi : le slicing en python
- La librairie pandas, utilisation du type DataFrame : documentation de pandas
- Indexation des DataFrame de pandas : dans la documentation de pandas
- Les compréhension de listes
- Types mutables, non-mutables
- Gestion des exceptions, try…except : sur docs.python.org
- Quelques bases de programmation orientée objet : sur courspython.com

Introduction



Python is both the fastest growing and (one of) the most popular programming language right now.

- ➔ **Easy to learn** (high level, easy to debug)
- ➔ Massive, **open source** community
- ➔ Very **modular**, thus **versatile**

- ➔ Widely **taught to students in science**
- ➔ Used in **large scientific facilities**

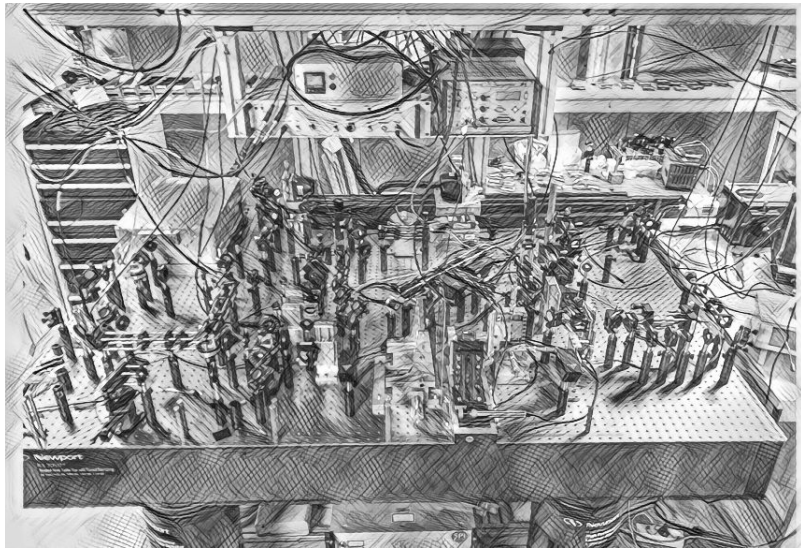
Good for :

- Web development
- Machine learning/AI
- Instrumentation
- Data visualization/analysis

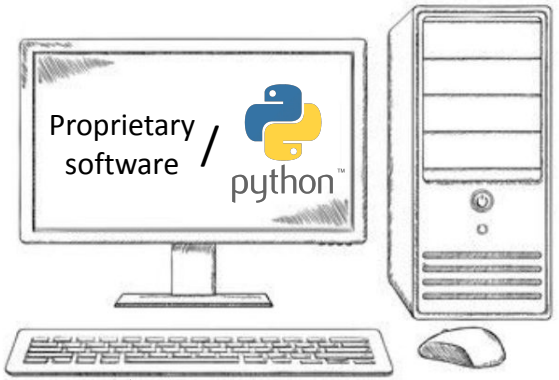
+ a lot more !

Python from hardware control to data analysis

Instrument



Data acquisition



Data analysis

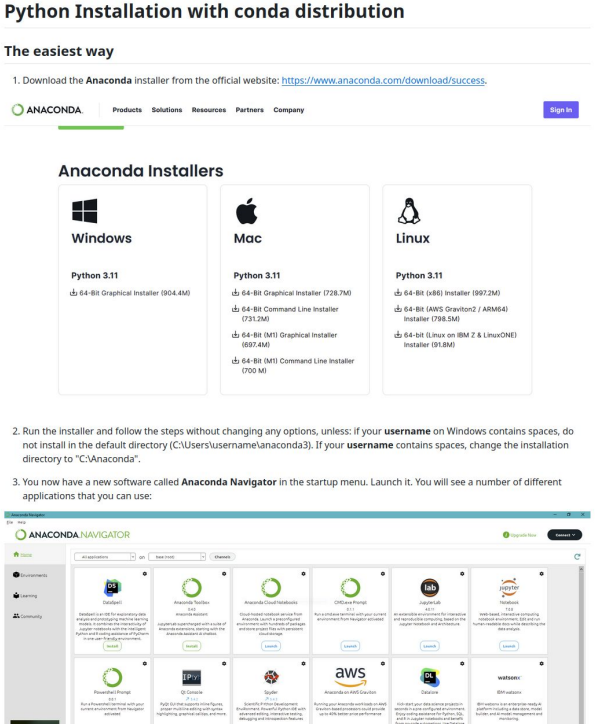
- Data reduction
- Fit
- Representation

- Drivers : to connect to the instrument
- SDK (software development kit) : proprietary libraries to control the instrument

How to python: Installation

How to install Conda: https://alexisforestier.github.io/Atelier-Python-HP/installation.html

Official documentation:
https://docs.conda.io/en/latest



Python Installation with conda distribution

The easiest way

1. Download the **Anaconda** installer from the official website: https://www.anaconda.com/download/success.

ANACONDA. Products Solutions Resources Partners Company Sign In

Anaconda Installers

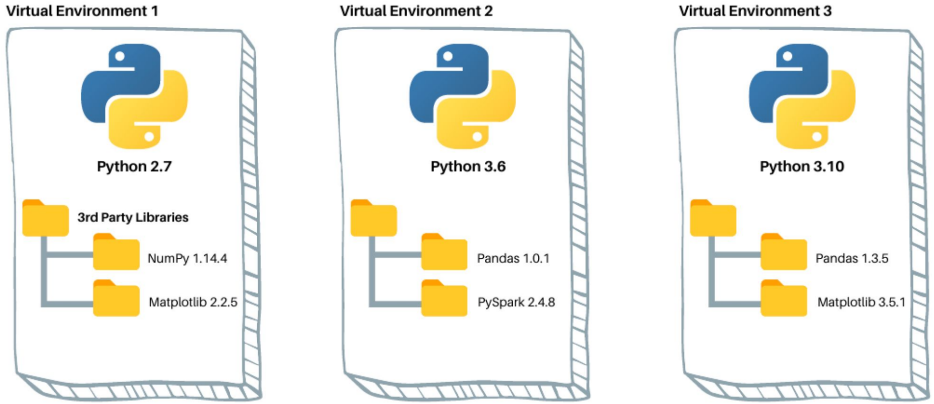
Windows	Mac	Linux
Python 3.11	Python 3.11	Python 3.11
64-Bit Graphical Installer (904.4M)	64-Bit Graphical Installer (728.7M)	64-Bit (x86) Installer (997.2M)
	64-Bit Command Line Installer (731.2M)	64-Bit (AWS Graviton2 / ARM64) Installer (798.5M)
	64-Bit (M1) Graphical Installer (697.4M)	64-bit (Linux on IBM Z & LinuxONE) Installer (91.8M)
	64-Bit (M1) Command Line Installer (700 M)	

2. Run the installer and follow the steps without changing any options, unless: if your **username** on Windows contains spaces, do not install in the default directory (C:\Users\username\anaconda3). If your **username** contains spaces, change the installation directory to "C:\Anaconda".

3. You now have a new software called **Anaconda Navigator** in the startup menu. Launch it. You will see a number of different applications that you can use:

ANACONDA.NAVIGATOR

Using virtual environments :



dataquest.io

Allow multiple Python installations and set of libraries to coexist

Write code: Text editor vs. Notebook

Python is a **interpreted** language : **no compilation required**

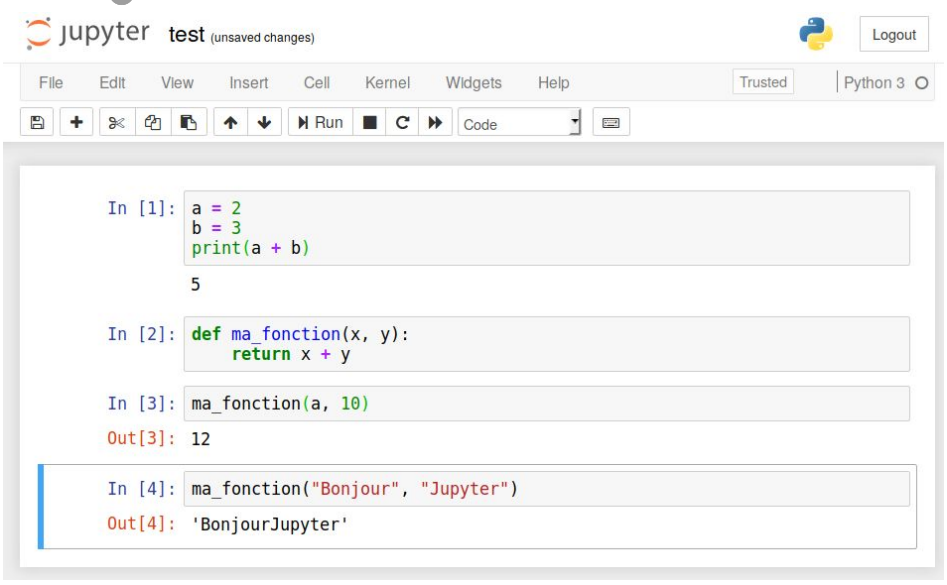
Basic Python interpreter:

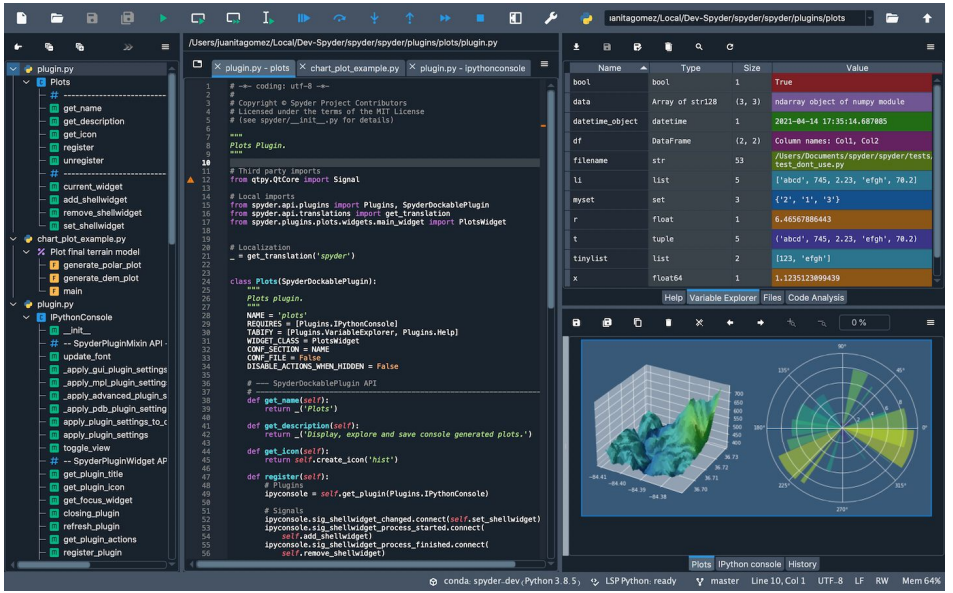
```
Python 3.6.9 (default, Mar 10 2023, 16:46:00)
[GCC 8.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> x = 3
>>> x*2
6
>>>
```

There are **two popular ways** to interact with Python code:

jupyter Notebook : notebook file .ipynb
use in the web browser

spyder, or any text editor
raw python file .py





Write code: **Text editor** vs. Notebook

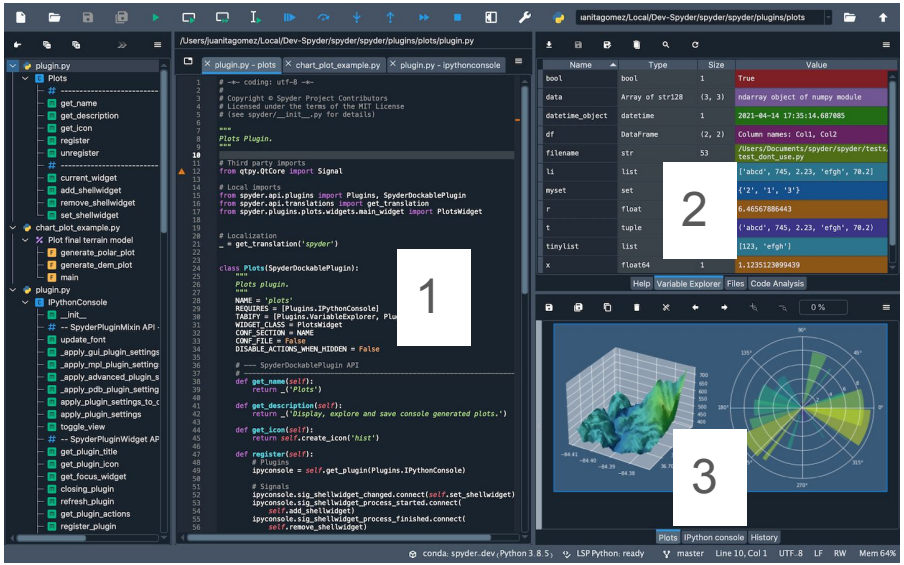
Use a text editor, or an
‘integrated development environment’ (IDE) :
the “developper” way

➔ Write a **.py file**
➔ Execute it using a terminal or console

Modern editors like **Spyder** or **VSCode** usually provide :

- Graphical interface to execute the .py file
- Interactive console
- Autocompletion
- Syntax highlighting
- Workspace splitting
- Variable explorer
- Easy data visualization
- Easy HTML report generation
- (Optionally executable by blocks of lines)

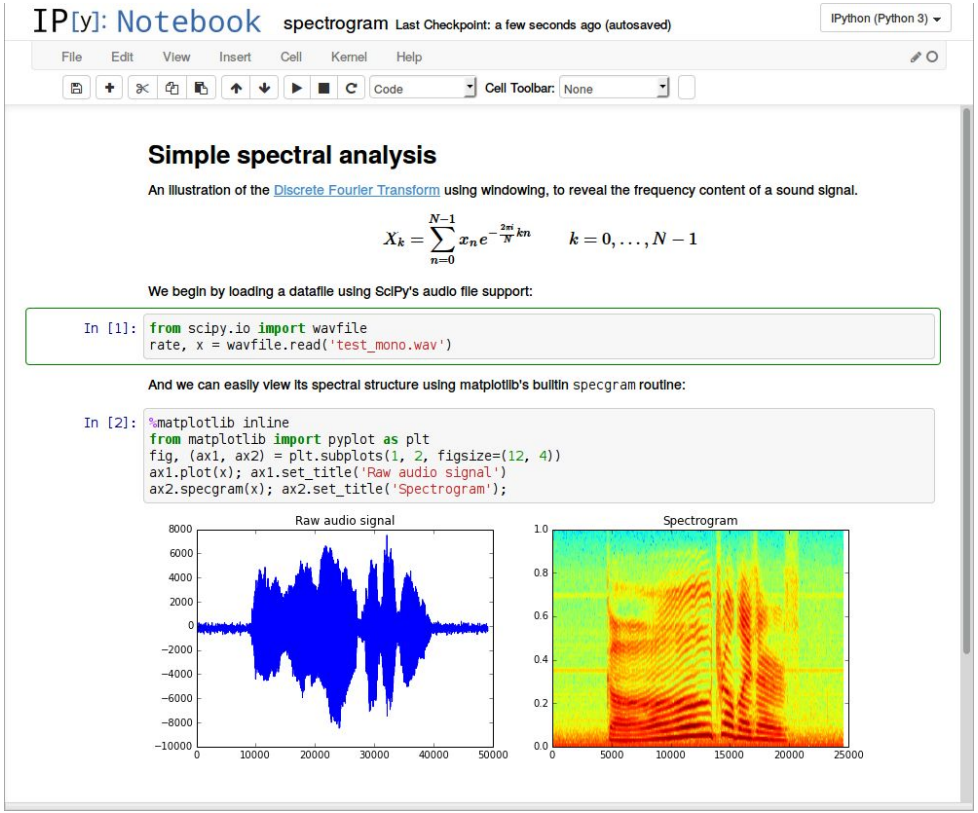
Ex: **Spyder**



1. Python code editor
2. Variable explorer
3. Console for output visualization

Official documentation:
https://docs.spyder-ide.org/5/index.html

Write code: Text editor vs. **Notebook**



Jupyter is a free open source software offering electronic notebooks mixing:

- ➔ Cell-based code execution
- ➔ ‘Markdown’-formatted text
- ➔ Latex formulas

The resulting file is a **.ipynb notebook**

- Interactive
- Execution by blocks for quick trial/error
- Take the form of a document (exportable)
- Easy readable documentation
- Facilitates collaboration
- Supports multiple programming languages (julia, R)
- Well suited for data exploration and analysis

\+ More in the hands-on part !

User interfaces

Python allows to easily build user interfaces, with different levels of friendliness and visual refinement.
This is great to share your work with the world !

Here are a few examples within high pressure science …

1. Command-line interface (CLI)
2. Tkinter
3. PyQt

User interfaces

Example:

Amorpheus

Command-line Interface (CLI)

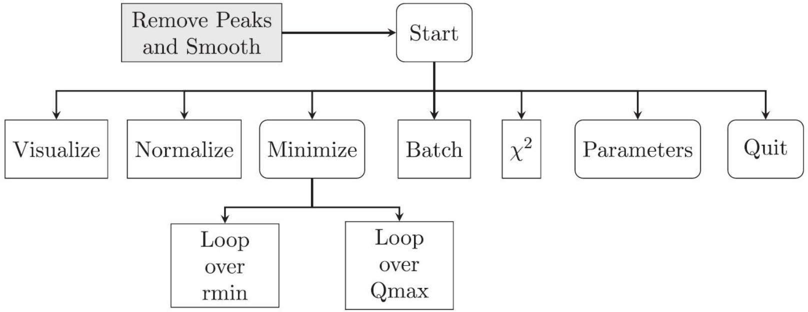
Keyboard only interaction

a. efficient navigation in the menu
b. quick change of input parameters
c. simple and user friendly

```
/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\

Welcome to AMORPHEUS!!

#################################################################
#
#  Visualize [v] - Normalize [n] - Loop rmin [r] - Loop qmax [x]
#  Batch [b] - Chi2 [c] - Parameters [p] - Quit [q]
#
#################################################################
```



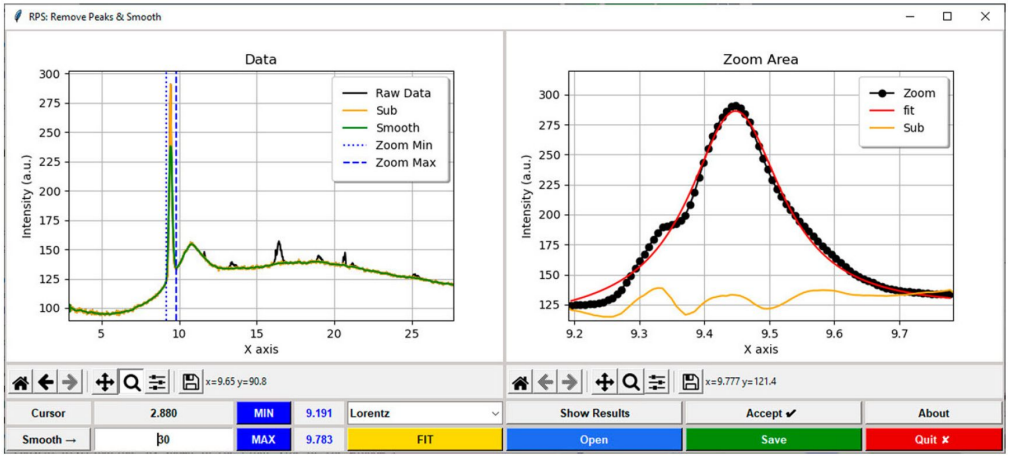
https://github.com/CelluleProjet/Amorpheus

e.g. Bliss at the ESRF

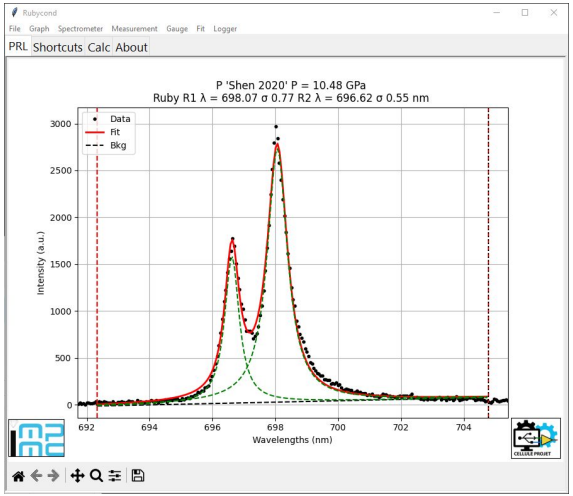
User interfaces

Examples:

RPS: Remove Peaks and Smooth



Rubycond



Tkinter : RPS, Rubycond

Basic graphical interface

a. part of the Python standard library, comes installed by default
b. extensive documentation and large community
c. quick to learn
d. same license as python

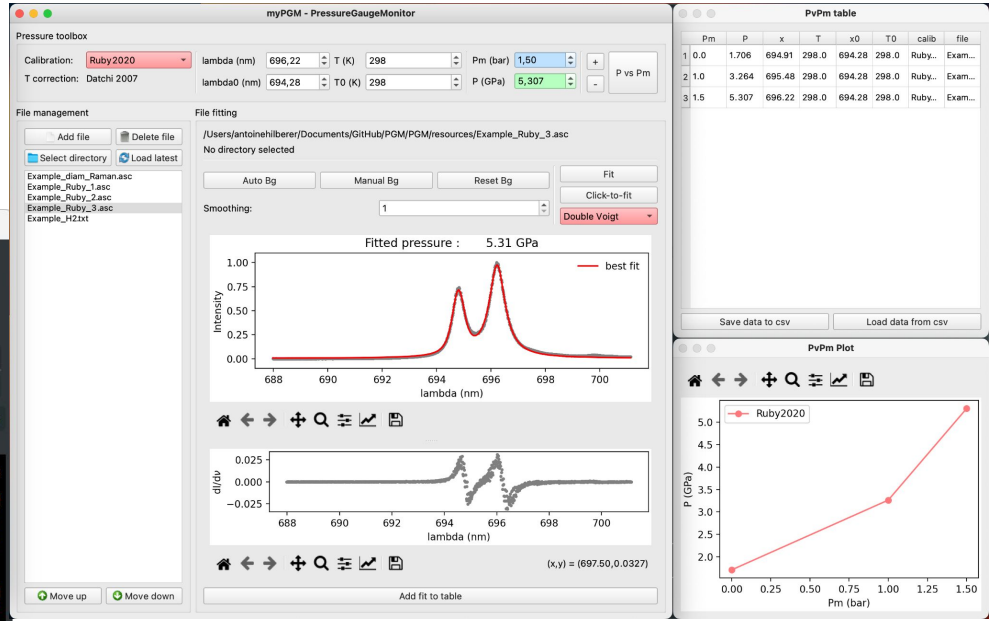
https://github.com/CelluleProjet/Rubycond

User interfaces

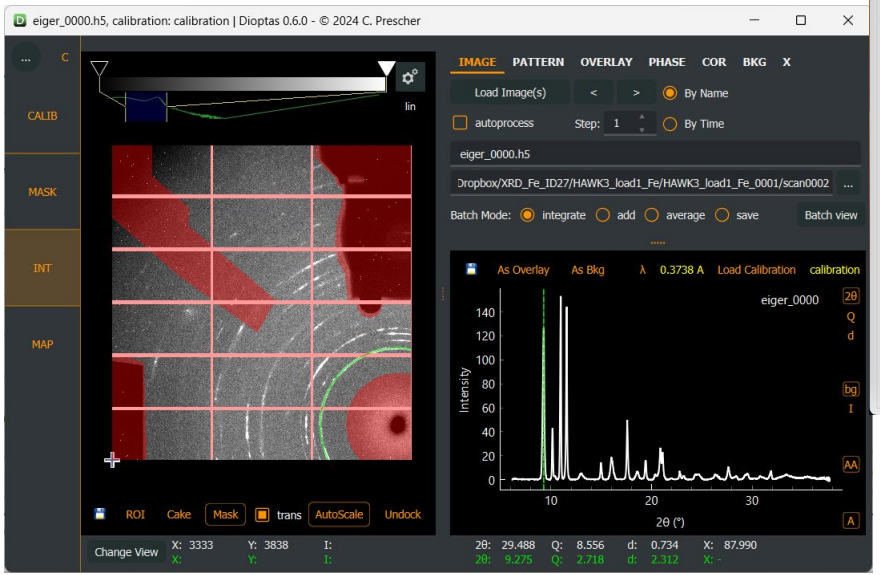
Examples:

PyQt : Python implementation of the industry-standard GUI engine Qt, for advanced graphical interface

myPGM https://github.com/AHilberer/myPGM



Dioptas



https://www.clemensprescher.com/programs/dioptas

Tkinter vs PyQt

Tkinter - beginners and small projects	**PyQt** - large projects and professional application
https://tkdocs.com/tutorial/index.html	https://www.pythonguis.com/pyqt5-tutorial/
part of the Python standard library (installed by default)	requires a separate installation
extensive documentation and a large community	poor and incomplete Python documentation
quick to learn	not easy to learn
it can be used freely for commercial software because its license is the same as Python's	it allows for a GPL or commercial license only
basic widgets	wide range of advanced features
old fashioned look	modern native look
has decent performance for small to medium-sized applications	better performance, more suitable for larger projects or app that requires fast refresh

More Info: https://www.pythonguis.com/faq/pyqt-vs-tkinter/

Tons of useful libraries for data analysis:

See also:
https://alexisforestier.github.io/Atelier-Python-HP/

matplotlib — Easy yet extremely customizable scientific plots

NumPy SciPy — Basically all the maths you might ever need

pandas — Large data structures and management (dataframe…)

LMFIT — Easy and modular curve-fitting

h5py — Opening HDF5 files from synchrotron facilities (ESRF, SOLEIL …)

fabio, PyFAI (Fast Azimuthal Integration) — Manipulate images produced by 2D X-Ray detectors

tkinter, Qt — Graphical interfaces

Supplementary

Prepare answer to AI help for python:

Github copilot

(https://www.boardinfinity.com/blog/10-best-ai-python-code-generators-in-2024/)