HyProC - Manual

24th September 2025

L. Dupont

Starting HyProC

Run the UI.py file. Make sure you are working from the HyProC folder so the copy of SRIM that's included can be accessed.

Button behaviors

Importing experimental excitation curve

Loads an excitation curve.

File format: Excel (.xlsx, .xls, .csv).

If there are multiple sheets in the loaded file, the first sheet will be read. Headers must be on the second row. The code will read the columns whose headers are: "Energie (keV)", "Ng/µC" and "Incertitude". Those parameters (sheet number, header row and headers' names) can be changed in the import_file_settings.json file. When modifying them, remember Python uses zero-based numbering!

Data don't need to be sorted by ascending energy, automatic sorting of the data is implemented.

The Excel file cannot be loaded if opened in Excel, an error will appear otherwise.

Isolate & Normalise

Keeps the atomic concentration of the selected element constant whilst changing the concentration of all other elements.

Run Calculation

Calculate the yield for the energy values taken from the experimental excitation loaded. As a result, an experimental curve must be loaded beforehand.

The sum of atomic concentrations in each layer doesn't have to be 100%. Automatic normalisation is made if needed. This means stoichiometric coefficients can be used.

Broadening data

Broadening data can be saved by ticking the "Save broadening data" box. When calculations are launched with this option on, folders will be created using this architecture:

```
C:/HyProC/
└── Session/
    └── Run/
        └── Datapoint/
            ├── Beam.txt
            ├── Stragg.txt
            ├── Doppler.txt
            ├── Total_Gauss.txt
            ├── xsec.txt
            ├── Total_Broadening.txt
            └── Total_Broadening_TFU.txt
```

Total_Gauss represents the convolution of all gaussian broadening (Beam, Straggling and Doppler). Convoluting this curve with the cross section of the reaction (xsec, modelled as a lorentz curve) give the Total_Broadening. Total_Broadening_TFU represents the same data with a different x-axis (x [TFU] instead of Energy [keV]).

A Session folder is created when broadening data are saved for the first time since the interface was opened. A run folder is created whenever a calculation is made (with the save broadening data on). For each incident energy, a datapoint is created.

The broadening data can be visualised using the broadplotter.py script or through a third-party software (OriginLab, ...).