TASK 1 simulating a neutron Velocity_selector:

Instrumentfile: Ex_selector.instr

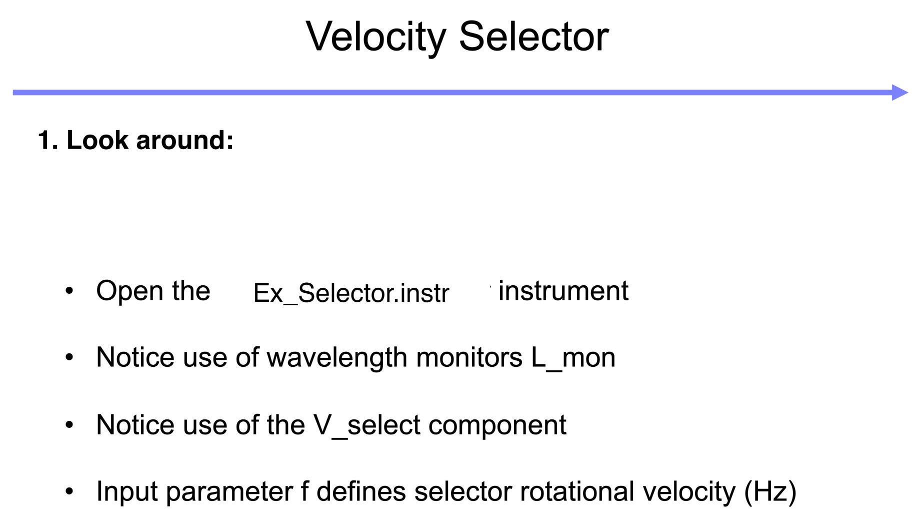


Figure 1: Look around

TASK 2 improve the instrument file:

Given that:

RESULT: Please form an expression of selector frequency as function of desired wavelength $f(\lambda)$ - may be used in Bonus task 3 below.

BONUS TASK 3 (or DEMO): Implement result from TASK 2 for futher work

- Exchange the instrument input-parameter `f` for a `lambda` (add a default of e.g. 6Å)
- In the `DECLARE` section, uncomment the `// double f;` -> `double f;`
- Add your equation in c-code under `INITIALIZE`

RESULT: Save this instrument file for further work on Tuesday

VELOCITY SELECTOR

2. Tasks

- Perform a TRACE at the default f=300 Hz
- Perform a SIMULATE of 1e7 neutrons at default f
- Estimate the relative bandwidth $\delta\lambda/\lambda$ of the transmitted beam
- Perform a series of simulations in the range
  - $150 < f < 800$ (5 steps)
- Compare the transmitted beam in the different cases
- Question: What is the ideal rotational speed to select neutrons of 10 Å with the selector from Ex 7.1?
  Hint: λ [Å] ≈ 3956/v [m/s]

Figure 2: Simulation tasks

VELOCITY SELECTOR

3. Analytical approach

Hints for analytical approach, use that:

1. Neutron velocity through selector $$v = \frac{\delta z}{\delta t}$$
   (and $\delta z = l$ in selector)

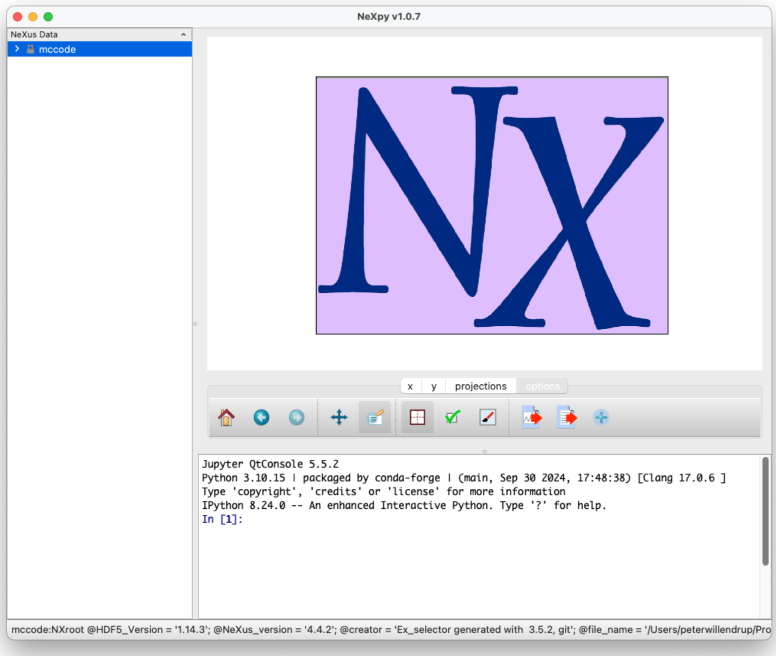
2. Relation between time of passage and selector parameters $$\delta t = \frac{\alpha}{2\pi}\frac{1}{\nu}$$
   Please note that in the instrument \alpha has a unit of degres, so use 360 deg in denominator :-)

3. Use that neutron $$v[\mathrm{m/s}] = \frac{3956}{\lambda[\mathrm{\AA}]}$$ (de Broglie)

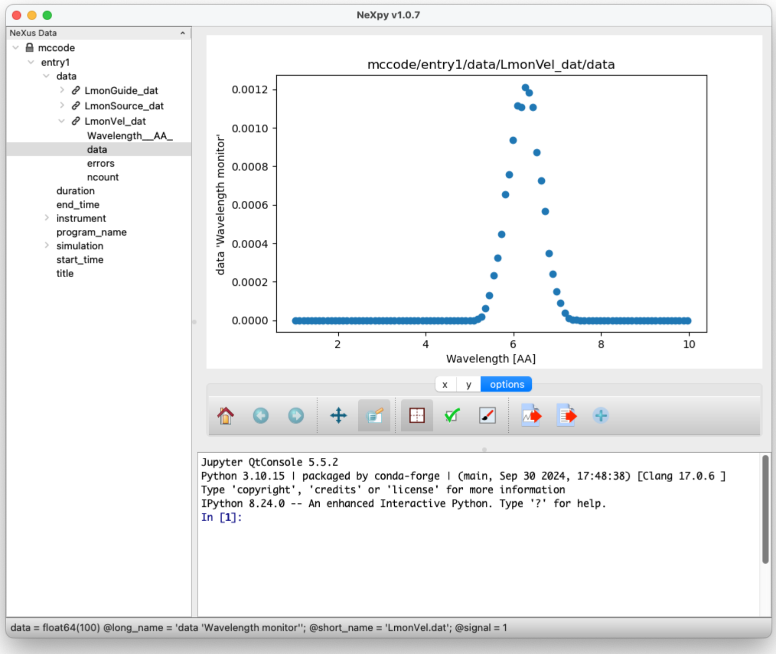
Figure 3: Analytical consideration

BONUS TASK 4 (or DEMO): Data analysis in NeXpy

- On the run dialogue in mcgui, select `Output format: NeXus -c` (-c means recompile to ensure format is supported) and run a simulation

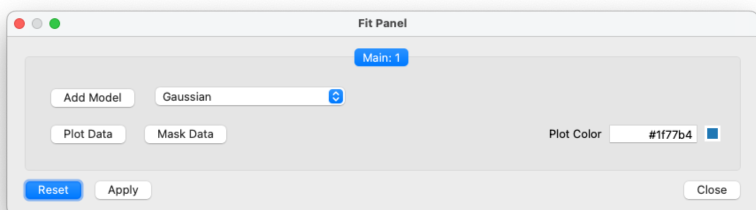


- Plot and NeXpy should appear:
- Expand the tree structure `mccode->data->LmonVel_dat`
- Double-click `data` inside and select `Axis 0` to be Wavelength in Å, a plot



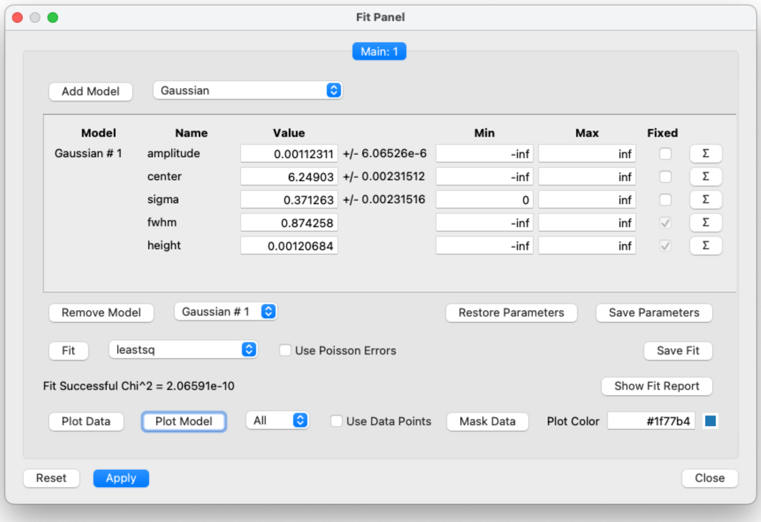
should appear:

- Clicking the `y` tab will give access to the `Fit` button, and pressing will bring

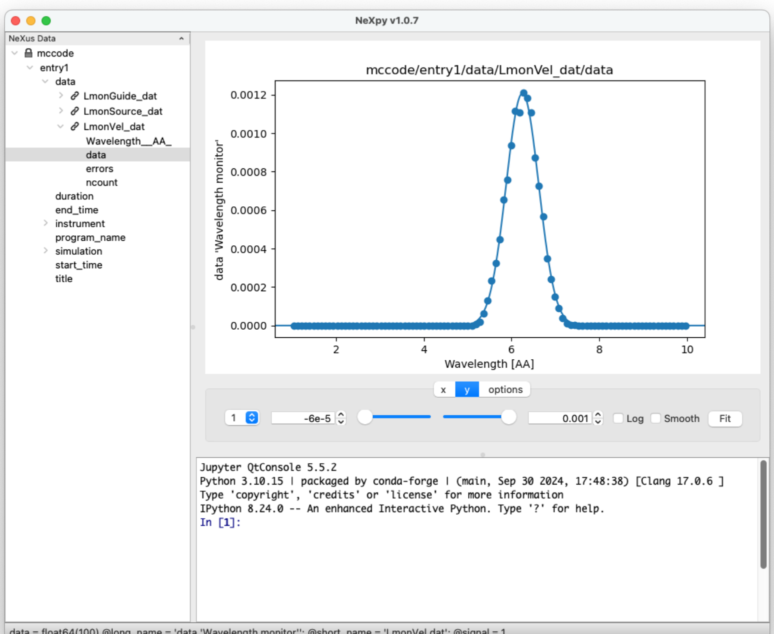


up a new small window:

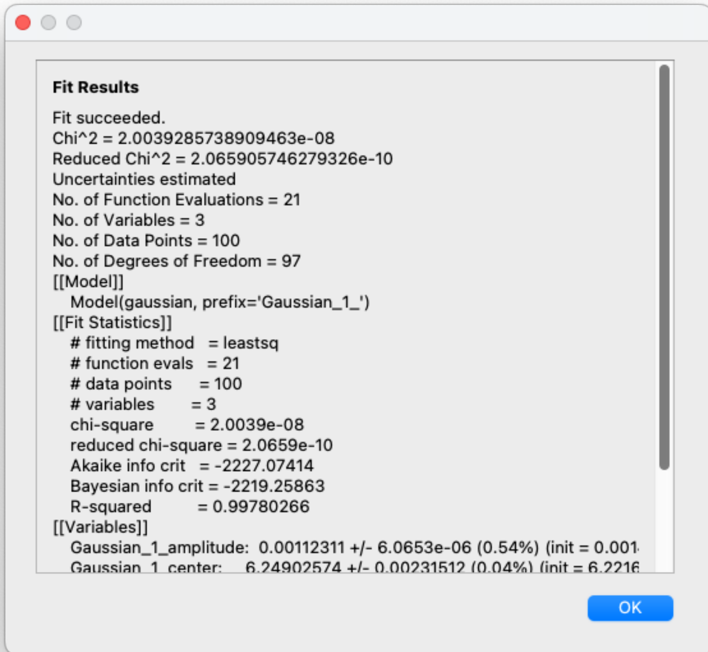
- Add a Gaussian Model, `Fit`, `Plot Model` and `Show Fit Report`



Fit Panel
Main: 1
Add Model
Gaussian
Model
Name
Value
Min
Max
Fixed
Gaussian # 1
amplitude
0.00112311
+/- 6.06526e-6
center
6.24903
+/- 0.00231512
sigma
0.371263
+/- 0.00231516
fwhm
0.874258
height
0.00120684
Remove Model
Restore Parameters
Save Parameters
Fit
leastsq
Use Poisson Errors
Save Fit
Fit Successful Chi^2 = 2.06591e-10
Show Fit Report
Plot Data
Plot Model
All
Use Data Points
Mask Data
Plot Color
#1f77b4
Reset
Apply
Close



NeXpy v1.0.7
mccode/entry1/data/LmonVel_dat/data
Wavelength [AA]
data 'Wavelength monitor'
Jupyter QtConsole 5.5.2
Python 3.10.15 | packaged by conda-forge | (main, Sep 30 2024, 17:48:38) [Clang 17.0.6]
Type 'copyright', 'credits' or 'license' for more information
IPython 8.24.0 -- An enhanced Interactive Python. Type '?' for help.
In [1]:
data = float64(100) @long_name = 'data 'Wavelength monitor''; @short_name = 'LmonVel.dat'; @signal = 1



(For those thinking a Gaussian is not a correct model in this case, this is correct. The v-selector transmission is in fact triangular ;-))

Hint: If you `save` your fits or parameters, these will become available in the left pane of NeXpy and can be saved by using a right-click menu point