Exercise in Polarized Neutron simulation using McStas

September 19, 2013

We will be using the following components: Set_pol, Pol_mirror, Pol_simpleBfield, Pol_monitor, PolLambda_monitor

1 Introduction.

In McStas each neutron ray is assigned a polarization vector, $\boldsymbol{P_i}$, which is the statistical mean of the magnetic moments taken over the ensemble of neutrons represented by the i:th ray. The scalar polarization P is the weighted mean of a set of rays:

$$P = \frac{\sum_i w_i(\mathbf{P} \cdot \mathbf{S})}{\sum_i w_i}$$

where the weights w_i are the Monte Carlo weights of the rays and the vector $\mathbf{S}$ is a reference direction, often this is just the vertical (y) direction.

1.1 Front matter

Set up an instrument with a simple source and an ideal polarizer (Set_pol) and verify that you do get the polarization you expect.

1. Start out with an empty template instrument
2. Insert a simple neutron source with a flat wavelength spectrum centered around 5 Å.
3. Insert an ideal polarizer `Set_pol` with the parameters: `px=0, py=0, pz=1`. This sets the polarization of all rays passing through the device exactly to 1, i.e. a perfectly polarized beam.
4. Verify that this works by adding a polarization monitor `Pol_monitor` after the polarizer (use parameters `mx=0,my=0,mz=1`).

If you ask for a polarization-vector |P|<1 – what does that mean?

1.2 Polarization monitors

Set up a bank of PolLambda_monitors to monitor the X Y Z-components of the polarization vs lambda, to get a full handle on what the polarization is.

1.3 Constant magnetic field

Insert a constant magnetic field of 1 T perpendicular to the polarization direction, and monitor the wavelength dependence of the polarization downstream. First insert a field region using the Pol_simpleBfield component. For instance with parameters as follows:

```
COMPONENT field = Pol_simpleBfield(
        xwidth=0.1, yheight=0.1, zdepth=1, Bx=0, By=1, Bz=0,
        fieldFunction=const_magnetic_field)
AT(0,0,1) RELATIVE PREVIOUS
```

They key parameter is the fieldFunction which points to a C-function that returns the magnetic field as a function of of time and spatial coordinates: $\mathbf{B} = f(x, y, z, t)$. Pol_simpleBfield defines an entry "window" beyond which the magnetic field exists. To define an exit window we use Pol_simpleBfield_stop.

```
COMPONENT field_stop = Pol_simpleBfield_stop(
        magnet_comp_stop=field)
AT(0,0,1) RELATIVE field
```

This defines a magnetic field volume $0.1 \times 0.1 \times 1\text{m}^3$ where the magnetic field is constant and point in the positive y-direction. Note that by rotating either or both the start and stop components oblique field regions may be targeted.

1.4 "Real" Polarizer

Exchange the ideal polarizer with a more realistic one which uses a difference between spin-up and spin-down refelctivities in a supermirror to create a polarized beam. In McStas we use the Pol_mirror component for this. To transmit a beam polarized a beam at 5 Å we can use the following:

```
COMPONENT polarizer = Pol_mirror(
        rDownPar = {0.99,0.11,0.9,2,0.0185},
        rUpPar = {0.99,0.0485,0.9,1,0.0002},
        zw = 0.3, yh = 0.3, option = 3)
AT (0, 0, 0) RELATIVE something
ROTATED (90-POL_ANGLE,90,0) RELATIVE something
```

Also change the samDid that work

2 Spin flipper

The Mezei version of a spin-flipper is, generally speaking, nothing else than a magnet with a field of a particular direction and magnitude. Consider the gyromagnetic ratio of and a field in the direction of b=(0 1 1).

The rotation angle:

$\alpha = \omega t = \gamma BL/\nu_n \frac{l}{t} = \gamma BL\lambda/A_n$; $A_n = 3956$ mÅ/s; $\gamma = 183 \cdot 10^6$ Hz/T

Can you find a combination of field region length and strength that would would flip the spin of a 5 Å neutron from the Z to the Y direction?

What happens with neutrons that are not 5 Å?

Why not turn the spin around b=(1 0 0) which could accomplish the same flip? Try to find out by changing the direction and strength of the magnetic field.

3 Tabled Magnetic field

Now change the constant magnetic field in your instrument file to one which reads the field description from a file "flipfield.dat". (please find the data file on the workshop web).

Insert a tabled magnetic field in your instrument instead of the constant one, using something like the following:

```
COMPONENT field = Pol_simpleBfield(
    xwidth=0.1, yheight=0.1, zdepth=1,
        filename="flipfield.dat",Bx=0,By=1,Bz=0,
        fieldFunction=table_magnetic_field)
AT(0,0,1) RELATIVE PREVIOUS
```

and later in the instrument file (just as before)

```
COMPONENT field_stop = Pol_simpleBfield_stop(
        magnet_comp_stop=field)
AT(0,0,1) RELATIVE field
```

What happens? Take a look inside the flipfield data file to explain what is going on.