3 (e) Neutron Detectors in VITESS

1 Introduction and preparations

In this exercise we want to see how different kinds of neutron detectors can be modeled in VITESS. We concentrate on artificial setups showing the detector performance; realistic measurements including data evaluation are treated in exercise 3(f).

In order to see the detector performance easily, we build an artificial source emitting a flat spectrum: choose `source_const_wave`, create a *moderator description file* for a 10x10 cm^2 rectangular source at (0,0,0) with a temperature of 50 K. Further create a *user wavelength dist. file* looking like this:

```
00.01 1E6
10.01 1E6
```

Set the simulated waveband to [0.1,10] Å. Put a wavelength monitor directly behind the source to check that the above input file gives you a flat wavelength distribution.

Now we are set to test the wavelength dependent detection efficiency!

2 Detection material and efficiency

Add a detector the same size as the source and direct the neutrons from the source onto the detector surface (*direction defined* by virtual window, 10x10 cm^2 in a distance of 100 cm). The default values in the detector should give you a flat area/volume detector in normal usage, with 10 repetitions, centered at $(\theta,\phi)=(0,0)$ in a distance of 100 cm. Set height and width to 10 cm.

The new `detector` module offeres three different ways to set a detection efficiency: calculated from a chosen detection material, read in from an input file, or by using a constant mean efficiency.

1. We start with the simplest option: a **constant efficiency**. To obtain this, set *absorber/converter type* to "other" and set a mean efficiency of 0.9 in *efficiency modifyer*. Add a wavelength monitor with the same settings as the one at the source (but different output filename!) after the detector and start the simulation. Since all neutrons hit the detector, a comparison of the intensity in the two monitors directly shows the detection efficiency. It should be wavelength independent, and 90%.

2. Set the detector thickness to 10 cm and repeat. The efficiency is now a bit lower than 90%: the value you give is the mean efficiency of neutrons hitting the detector surface perpendicular, i.e. with a maximal length through the detector of "thickness" cm. With a 10 cm thick detector, neutrons of larger divergence can leave the detector through the side/top/bottom surface and thus have a shorter way through the detection material. Therefore, set the neutron direction in the source to "by divergence" and use a small divergence of 0.01°. Now the efficiency is back to 90%!

3. The second possibility to model a detection efficiency is by a **user-defined input file**: create an efficiency file with two columns, the wavelength and according efficiency, by giving a filename in *lambda efficiency* and clicking on "Edit". Write
   4.0 0.5
   5.0 0.6
   6.0 0.8
   Note that this is not a realistic detector efficiency description, but serves for illustration purposes only! Run the simulation and look at the wavelength dependence of the efficiency by comparing the wavelength monitors: we have only defined the efficiency between 4 Å and 6 Å; the efficiency in between is extrapolated from these values. The efficiency for wavelengths outside the given range are set to 0 in case of long wavelengths, and to the one of the shortest given wavelength in case of smaller λ values[1]. Note that this efficiency is now independent of the detector thickness even for a larger divergence range since it is independent of the pathway through the detector. It is also independent on the selection in *absorber/converter type*, which is ignored in case a *lambda efficiency* file is given.

4. Let's remove the efficiency input file and choose a **detector material**: use a helium or BF_3 gas first and give a sensible *gas pressure* and *gas temperature*. Make sure the *efficiency modifyer* is set back to 1. Now the detector thickness matters: compare the efficiencies for a 10 cm and a 1 cm thick gas detector.

5. Change to a solid material and give reasonable numbers for the layer thickness and the atom density of the converter material. Of course, the layer thickness should be smaller than the total thickness divided by the number of layers! The possible path length of neutrons through detection material is determined by the number of layers times the layer thickness now[2]! Since the efficiency is calculated from the total cross-section of neutrons with the detection material, in case of a solid material this usually corresponds to the probability of producing secondary particles to be detected. You can include wavelength independent efficiency reducing effects, like the efficiency of secondary particle detection, by the *efficiency modifyer*, which now that a material is chosen acts like its name implies: it modifies the calculated efficiency.
   Look at the detected spectrum for different detector materials in combination with different values of the relevant parameters.

[1]In Vitess 3.2, the efficiency for $\lambda > \lambda_{max}$ will be set to the one of λ_{max} like it is done for $\lambda < \lambda_{min}$. You should however make sure you cover the whole wavelength range in your file when you use this option!

[2]In a gas detector, the number of layers just determines the pixel size in x direction.

3 Detector grid and resolution effects

Now we want to learn about the detector resolution:

1. Put identical 2D position monitors before and after the detector (same size as the detector). Change the source size to 8x8 cm^2 and choose the direction in the source by divergence (0.01°) such that only the central 8x8 cm^2 of the detector surface are illuminated.

2. Set 1x1 cm^2 pixels by setting 10 rows and columns; we don't see the pixelization in the x direction in our monitors, but the effect is the same as in y and z (you can see it by using a `writeout` module after the detector and plot the x position values). Start the simulation and compare the position monitors: the detector only gives the pixel centers.

3. Now set the *hor.* and *vert. resolution* to 0.3 cm and run again. You shouldn't see a difference (why?).

4. Set the number of rows and columns to 500 and run again with 0 or 0.3 cm resolution. What do you expect?

4 Tube detector

In an area/volume detector which we used so far, the detection efficiency is independent of the position the neutrons hits the detector surface (although it can be dependent on the neutron divergence, as we have seen above). In a tube detector, the detection efficiency is higher at the center of a (circular) tube than at the borders since more detection material is crossed.

1. Change the detector *type* from are/volume to tubes. For horizontal (vertical) tubes, thickness/N_{layers} is the tube diameter and has to be equal to height/N_{rows} (width/$N_{columns}$). Build a detector out of horizontal tubes and run the simulation. When the number of columns (readout segmentation) is high enough, you should see stripes in the 2D position monitor: the detected position in y and x is just the tube center.
   In order to see the different detection efficiency across the tube diameter, set the detector *usage* to "grid off". Add a 1D position monitor to confirm that the intensity between tubes does not reach 0.

2. Set the tube walls $\neq$0 (e.g. 1 mm) and run the simulation again. The tube walls are insensitive regions with zero intensity measured.

3. Shift tube layers against each other (tick *layers shifted*): now the effect of the tube walls and rim area are reduced.

A tube detector can only be build in combination with a flate geometry.

5 Overall detector geometry: flat, cylindrical and detector array

The detector doesn't have to be a flat, rectangular surface centered at (distance,0,0) as we used so far. You can build various detector shapes out of flat and cylindrical pieces.

1. Put a powder sample at 1 m from the source (`sample_powder`, use the NAC input file from the first day exercise, which is a cylinder of 3 cm height and 6 mm diameter). Focus the neutron beam in the source onto the sample.

2. First we use the same detector as in the previous exercise. Place it 1 m from the sample such that it cover a (θ,ϕ) region of $\pm 10°$ in ϕ and $\pm 20°$ in θ directions, centered at (60°,0°). In the sample module, set the scattering direction accordingly to save simulation time. Use the visualization to check that you put the detector in the right position.

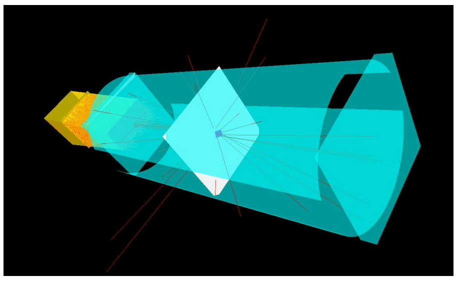
3. Change the detector *geometry* to cylindrical. You can't use cylindrical tubes, so change the detector *type* back. Try different *axis orientation*s.

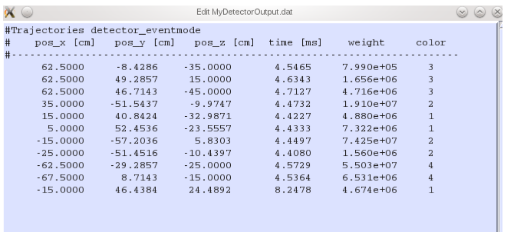
4. Now build a detector shape as shown in figure 1(a) out of several detector modules: you need two cylindrical detectors with axis along the x direction and two flat detectors to cover the forward and backward scattering regions. Tick *array (first or intermediate part)* in all but the last detector module. Let the sample scatter in all directions. Look at the logfile: it tells how many trajectories are detected in the different detector parts. (Ignore the error message about neutrons being already inside the detector: this is not a real error in this case, the logfile output will be fixed in release 3.2.). The order of the detector parts does not matter, apart from overlapping detectors (which shouldn't be build anyway): if detector parts overlap, the detector module first in the pipe will detect the neutrons.

5. You can distinguish neutrons detected in different parts of the the detector by setting *add to color* to different values. Set the color in the source to 0 and look at the different colors after the detector in the logfile or a monitor. This distinction is e.g. important for a time of flight analysis with detector parts at different distances from the sample: you can use several `eval_elast` modules to analyse neutrons from the different detector parts by the different color values!

6 Output file

The detected neutron properties can be viewed by monitors placed after the detector, and also the resulting d-spacing or Q value can be calculated and viewed directly with the `eval_elast` module (see exercise 3(f)). If the event rate is low compared to the detector resolution, an event mode detector output can be simulated by giving a filename in *Outpu filename*. In case of a detector array, giving a filename in the last detector is sufficient. The event mode output file contains a list of detected neutrons with the detected position, time of flight, neutron weight and color. An example is shown in figure 1(b).


(a) Example detector build as array of 4 subdetectors.



```
#Trajectories detector_eventmode
#    pos_x [cm]   pos_y [cm]   pos_z [cm]  time [ms]    weight     color
#-----------------------------------------------------------------------
     62.5000      -8.4286     -35.0000       4.5465    7.990e+05     3
     62.5000      49.2857      15.0000       4.6343    1.656e+06     3
     62.5000      46.7143     -45.0000       4.7127    4.716e+06     3
     35.0000     -51.5437      -9.9747       4.4732    1.910e+07     2
     15.0000      40.8424     -32.9871       4.4227    4.880e+06     1
      5.0000      52.4536     -23.5557       4.4333    7.322e+06     1
    -15.0000     -57.2036       5.8303       4.4497    7.425e+07     2
    -25.0000     -51.4516     -10.4397       4.4080    1.560e+06     2
    -62.5000     -29.2857     -25.0000       4.5729    5.503e+07     4
    -67.5000       8.7143     -15.0000       4.5364    6.531e+06     4
    -15.0000      46.4384      24.4892       8.2478    4.674e+06     1
```

(b) Example detector output file.