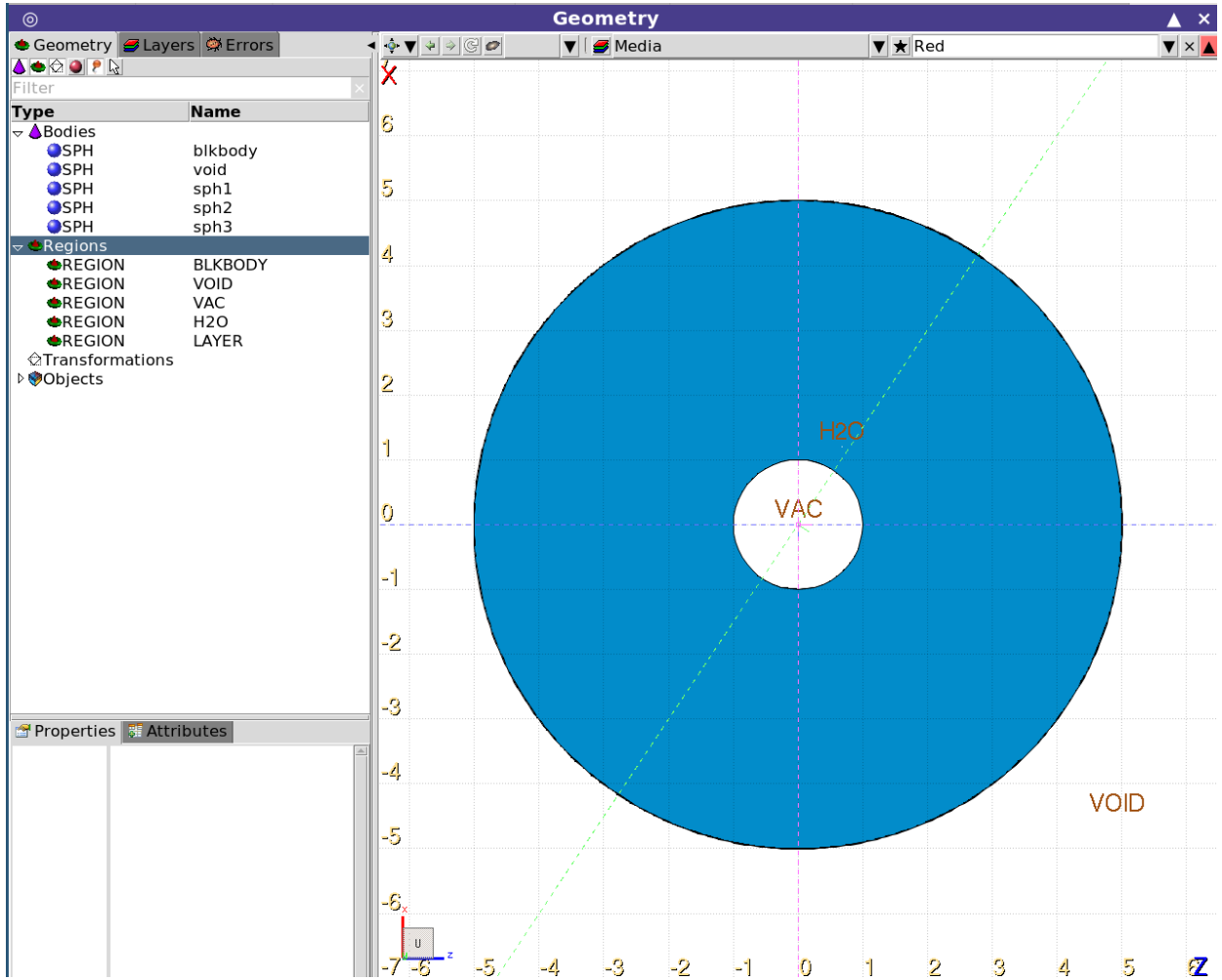
Exercise: low-energy neutronics

Exercise objectives

- Get familiar with FLUKA's pointwise treatment of low-energy neutrons and its advantages over a group-wise approach
- Witness how various neutron cross section features manifest in neutron fluences
- Master the plotting of histograms in logarithmic abscissas (lethargy units)
- Further practice with pre-processor directives
- Let's try to complete tasks 1-4, task 5 left as optional (examining crystal binding effects on the neutron fluence)

01 – Geometry (provided)



- Consists of three sphere bodies:
  - `sph1`, R = 1 cm
  - `sph2`, R = 5 cm
  - `sph3`, R = 5 cm + 100 μm
- …and corresponding regions:
  - `VAC`: the inside of `sph1`, material: `VACUUM`
  - `H2O`: outside of `sph1`, inside `sph2`, material: `WATER`
  - `LAYER`: outside of `sph2`, inside `sph3`, material: `VACUUM`

01 – Source, preprocessor, LOW-PWXS, scoring (provided)

- **Source** (isotropic 1 MeV neutron source):

```
BEAM                Beam: Energy ▾      E: =-1*MeV          Part: NEUTRON ▾
      Δp: Flat ▾          Δp:           Δϕ: Isotropic ▾
  Shape(X): Rectangular ▾  Δx:          Shape(Y): Rectangular ▾   Δy:
Define the beam position
BEAMPOS                   x:            y:                  z:
                          cosx:         cosy:               Type: POSITIVE ▾
```

- **Preprocessor** directives:

```
# #define pw
# #define 10B
# #define Cd
# #define graphite
# #define binding
```

- **LOW-PWXS** conditional on `pw`:

```
#if  pw ▾
LOW-PWXS          Mat: ▾      to Mat: ▾      Step:
db: ▾             IAZ:        S(α,β): ▾      T:
```

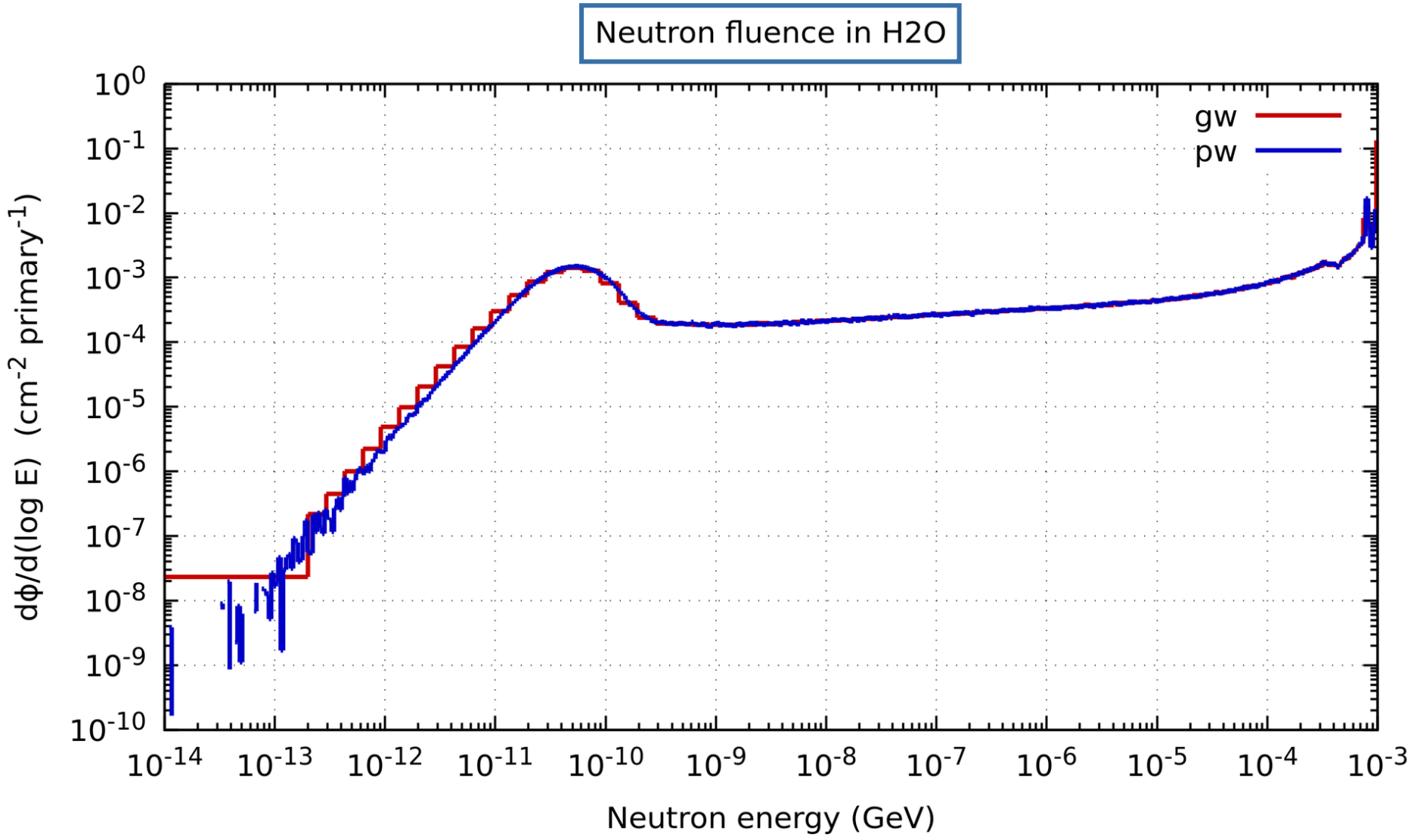
- **Scoring** (n fluence in the water and n fluence from the external layer to the void:

```
USRTRACK                                  Unit: 21 BIN ▾      Name: n_water
      Type: Log ▾      Reg: H2O ▾                               Vol: =4/3*pi*(body(sph2,4)**3-body(sph1,4)**3)
      Part: NEUTRON ▾  Emin: 1E-14        Emax: =1*MeV          Bins: 500
USRBDX                                    Unit: 22 BIN ▾      Name: n_emitted
      Type: Φ1,LogE,LinΩ ▾  Reg: LAYER ▾  to Reg: VOID ▾        Area: =4*pi*body(sph3,4)**2
      Part: NEUTRON ▾       Emin: 1E-14   Emax: =1*MeV          Ebins: 500
                            Ωmin:         Ωmax:                 Ωbins:
```

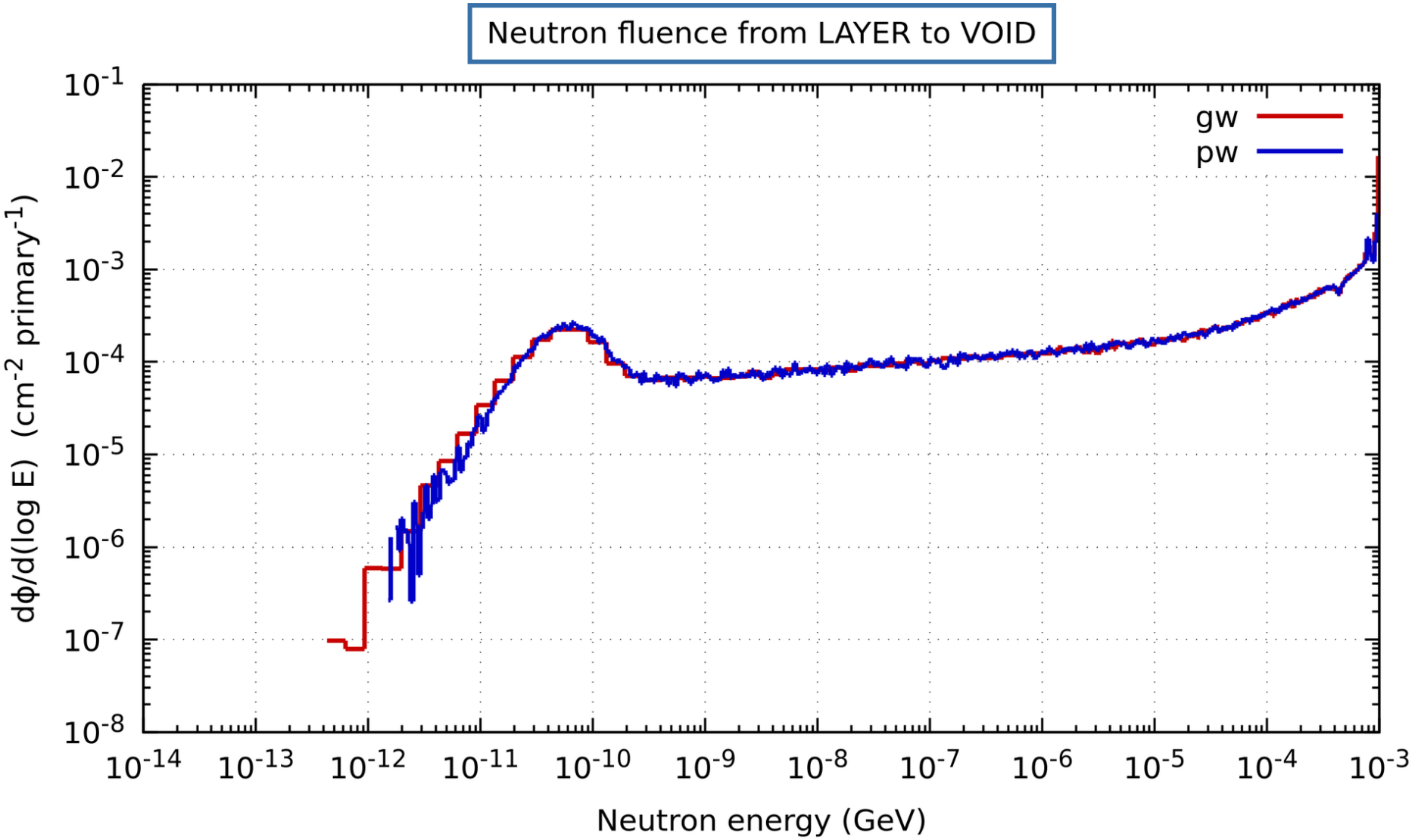
01 – Run, process, and plot

- Go to the Run tab and get ready to run the two already prepared runs:
  - `run/pw` with the `pw` directive active
  - `run/gw` with the `pw` directive inactive
- Both with 5 cycles, 25000 primaries per cycle
- Run! Process! Go to the Plot tab, and complete the placeholder plots:
  - ”fluence_in_water”: Plot the output from unit 21 of both runs in the same plot
  - ”fluence_from_layer_to_void”: Plot the output from unit 22 of both runs in the same plot
  - Set linewidth 2, Xmin=1e-14, Xmax=1e-3
  - Log scale Y
  - **Log scale X: please take measures to avoid misrepresenting spectra (lethargy scale!)**
  - Add appropriate labels for the X and Y axes
- For gnuplot gourmets: `set xtics 10; set ytics 10; set grid;`
  `set form xy “10^{%L}”`
- Can you explain the differences?

01 – GW vs. PW – Results



01 – GW vs. PW – Results



02 – Thin layer of ^{10}B

- All subsequent runs are with pointwise interactions (`pw` active)

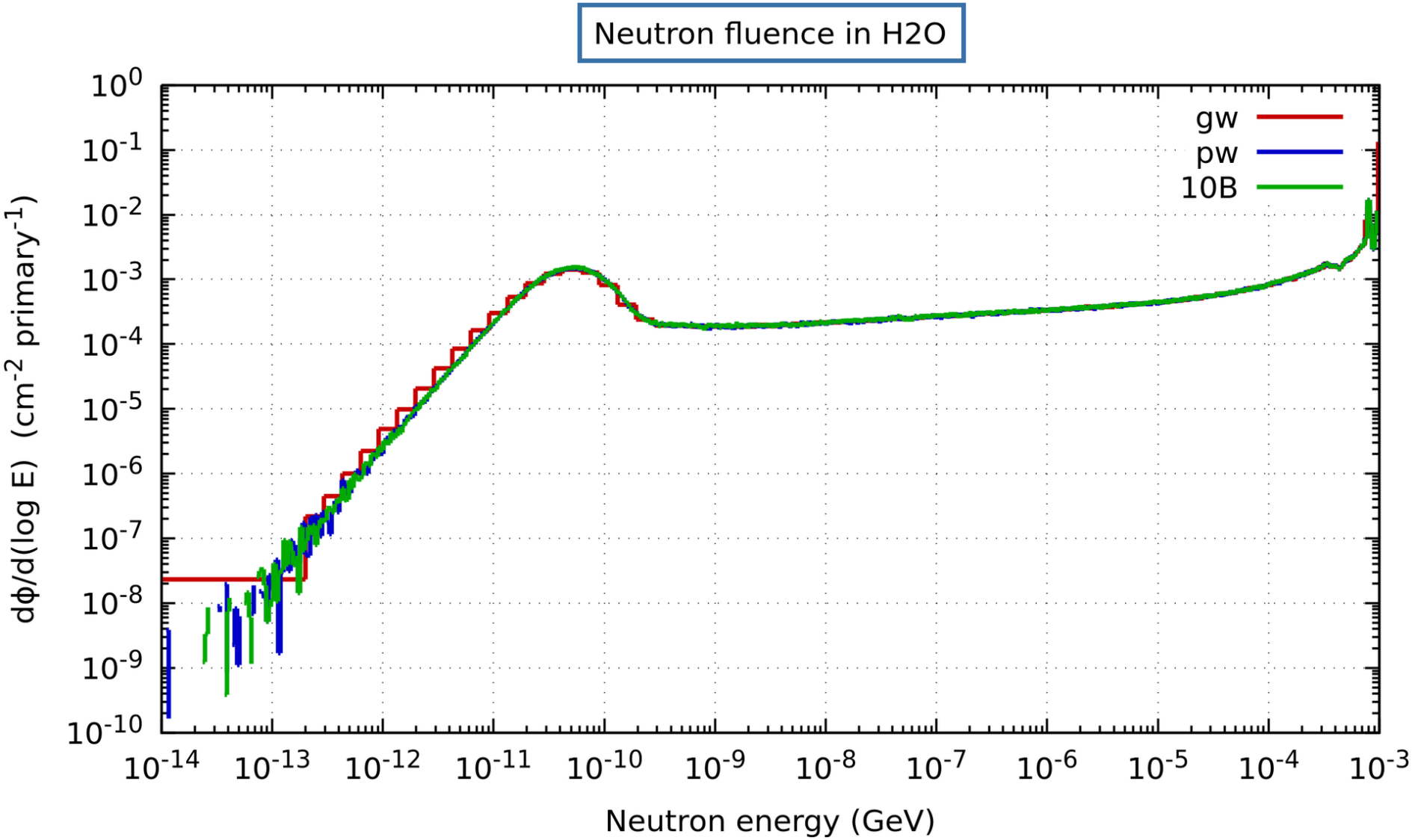
- Conditionally to the `10B` preprocessor variable being active:
  - Assign `BORON10` to the 100 μm `LAYER` region
  - Note the **`MATERIAL`** card defining the `BORON10` material (monoisotopic boron with ^{10}B, not natural composition)

- Add a new `run/10B` with both `pw` and `10B` variables active (all other variables off)
  No more group-wise runs from now on.

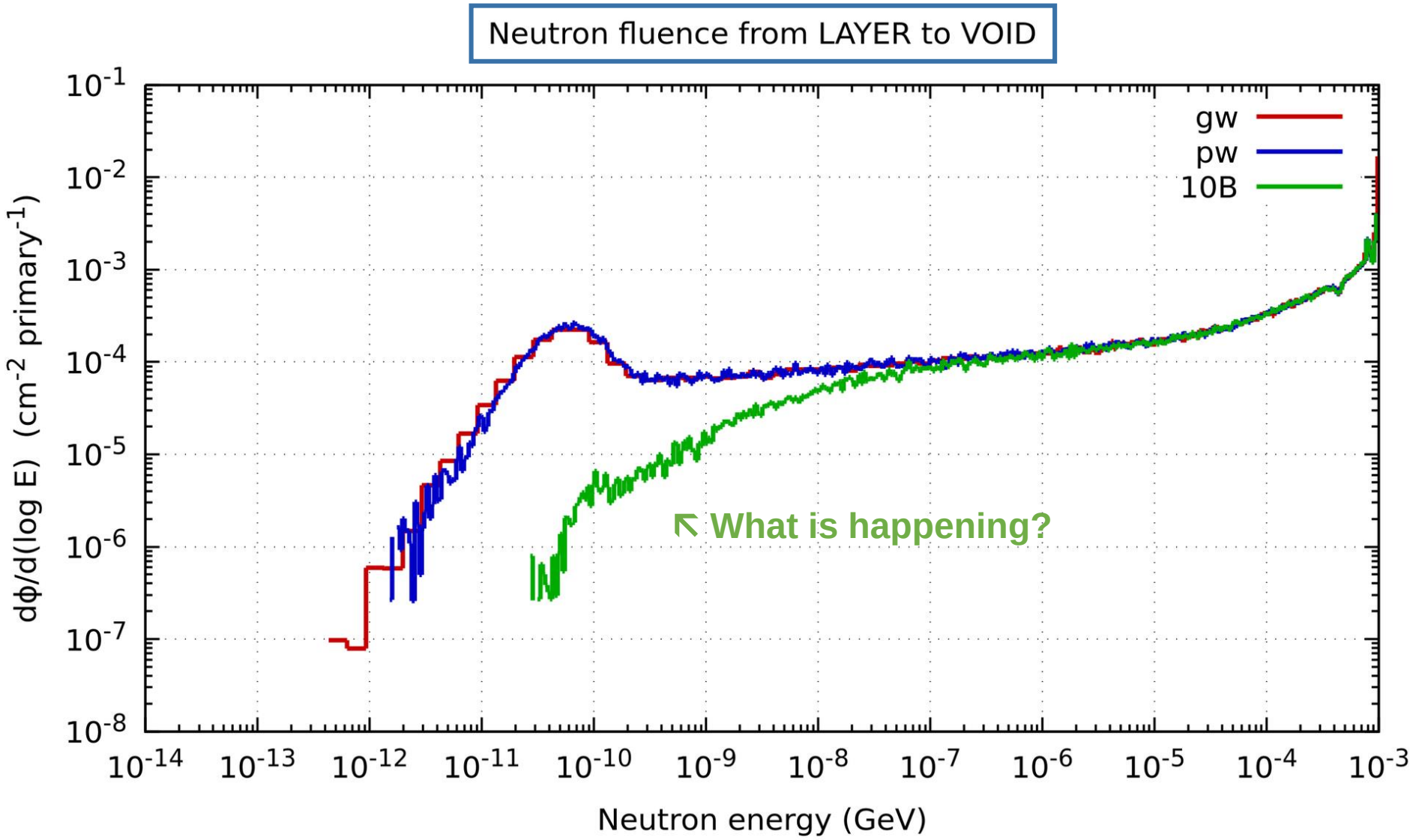
- Run! Process!
- Add the n fluences to the two plots

- What happened? Hint: slides of the first part of the lecture….

02 – Thin layer of ^{10}B – Results



02 - Thin layer of ^{10}B - Results



03 – Thin layer of Cd

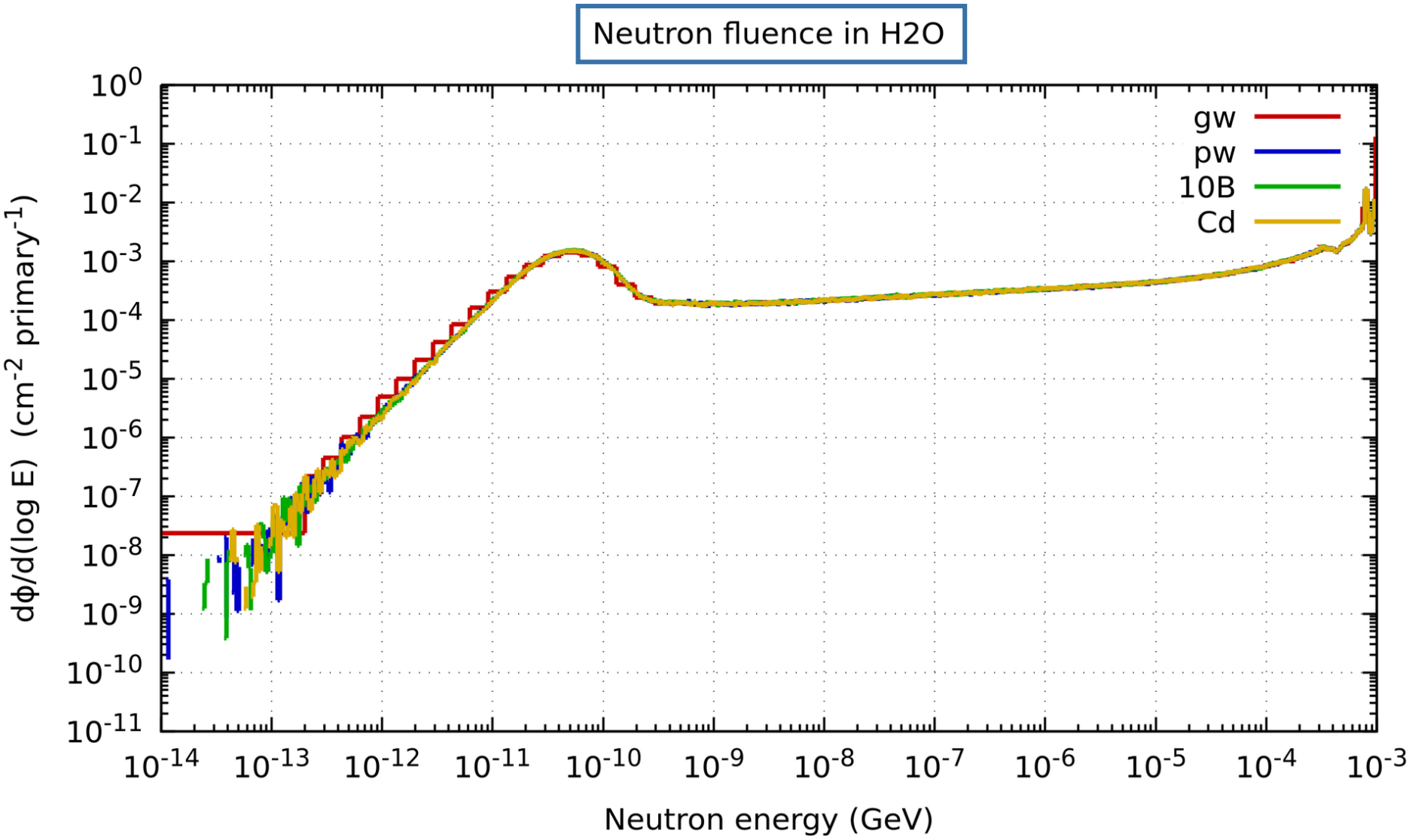
- Conditionally to the `Cd` preprocessor variable being active:
  - Note the **`MATERIAL`** card defining the Cd material with natural composition
  - Assign `CADMIUM` to the 100 μm `LAYER` region

- Add a new `run/Cd` with both `pw` and `Cd` variables active (all other variables off)

- Run! Process!
- Add the n fluences to the two plots. Maybe move the plot key to the bottom (too crowded)

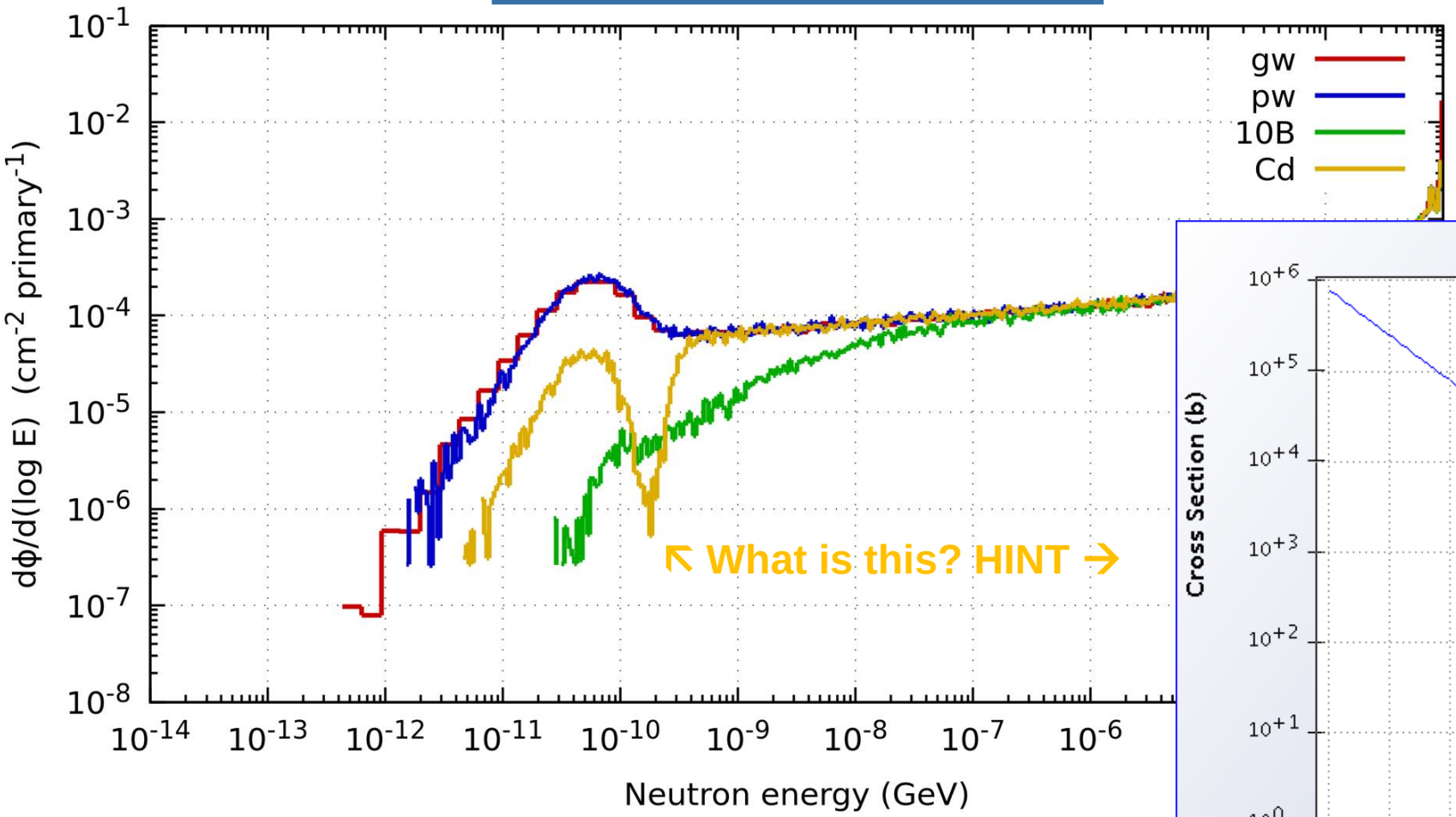
- What happened? Hint: slides of the first part of the lecture….

03 – Thin layer of Cd – Results

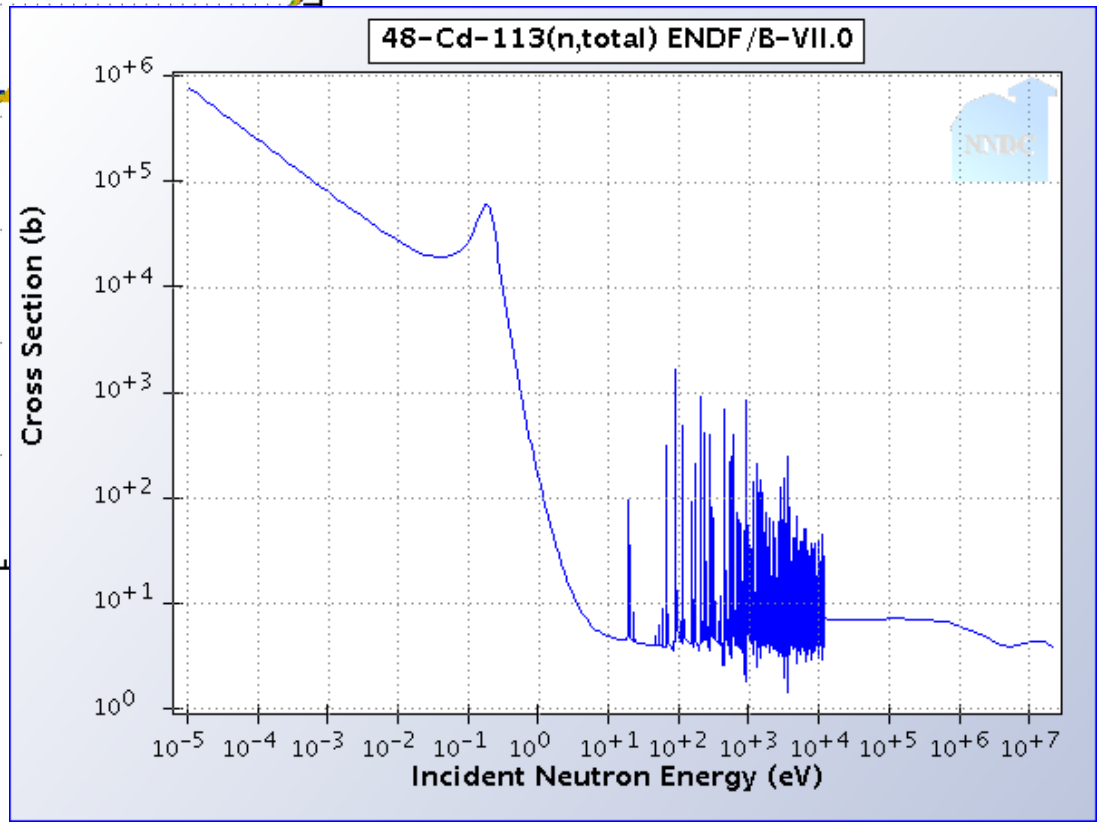


03 – Thin layer of Cd – Results

Neutron fluence from LAYER to VOID



↖ What is this? HINT →



04 – Thick layer of graphite

- Conditionally to the preprocessor variable `graphite` being active:
  - Change the thickness of `LAYER` to 5 cm

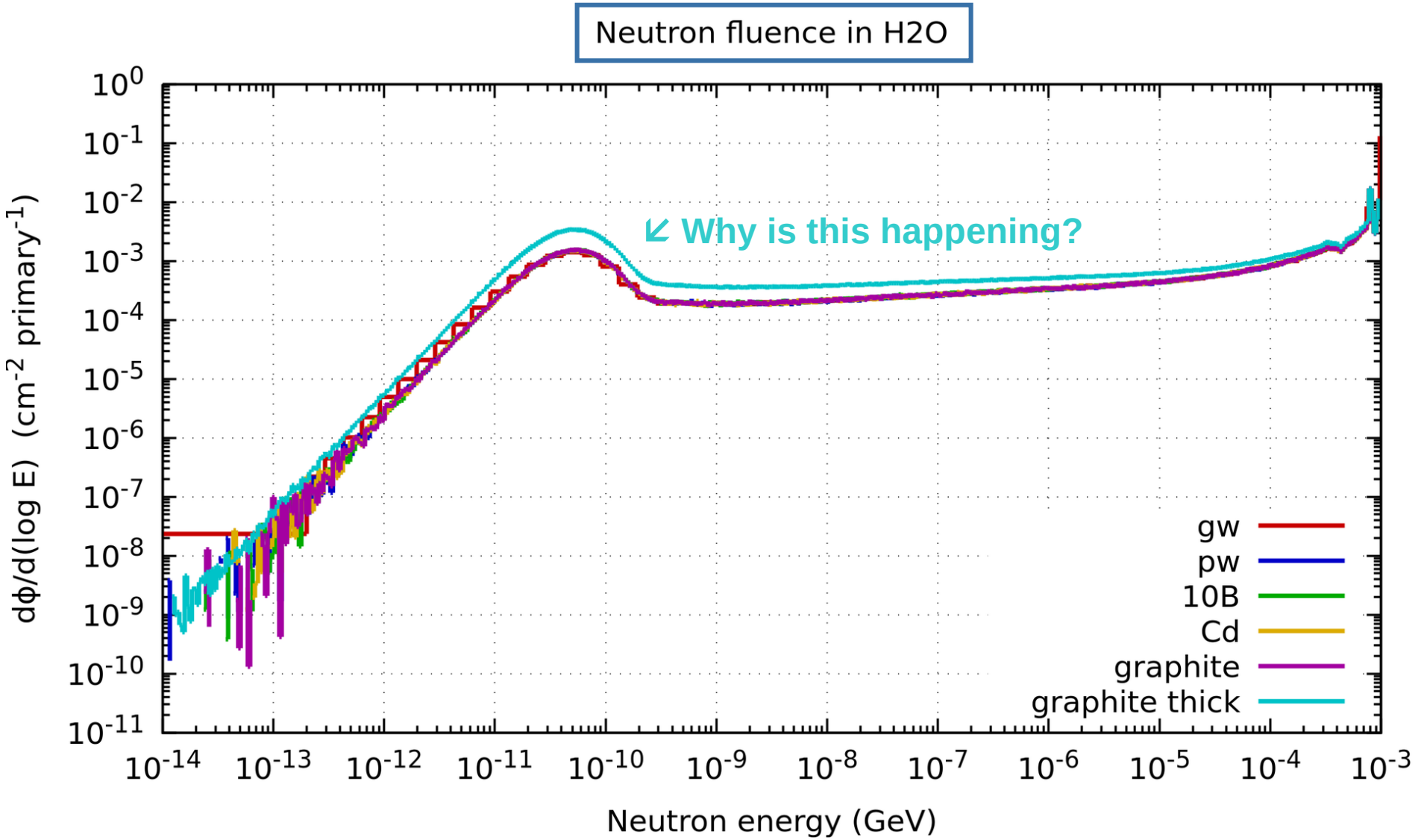
```
#if graphite
        R(sph3) = 10
#else
        R(sph3) = 5*cm + 100*um
#endif
```

  - Set the `LAYER` material to `CARBON`

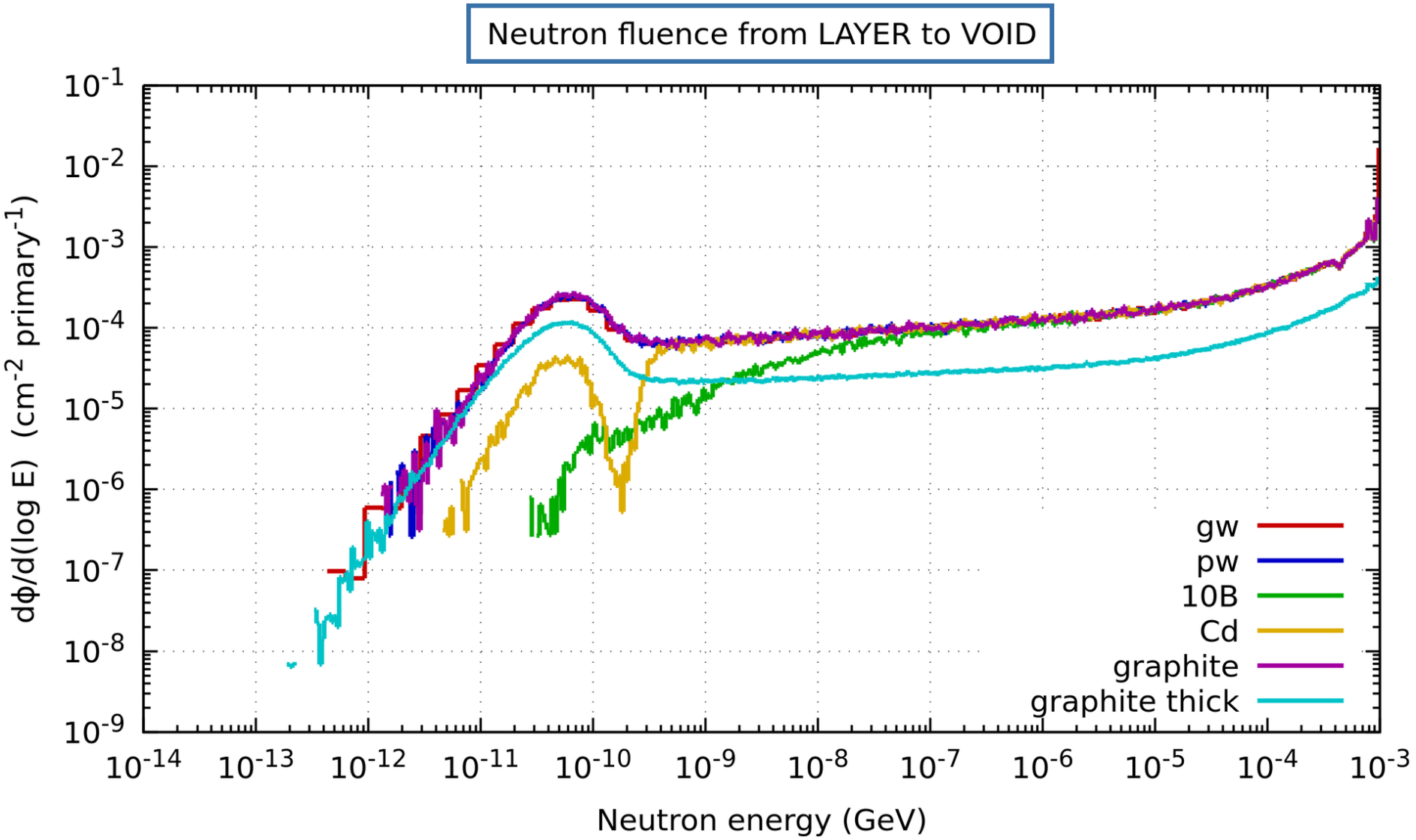
- Add a new `run/graphite` run with `pw` and `graphite` active

- Run! Process!
- Add the n fluences to the two plots
- What happened?

04 – Thick layer of graphite - Results



04 – Thick layer of graphite - Results



05 – Binding effects (optional)

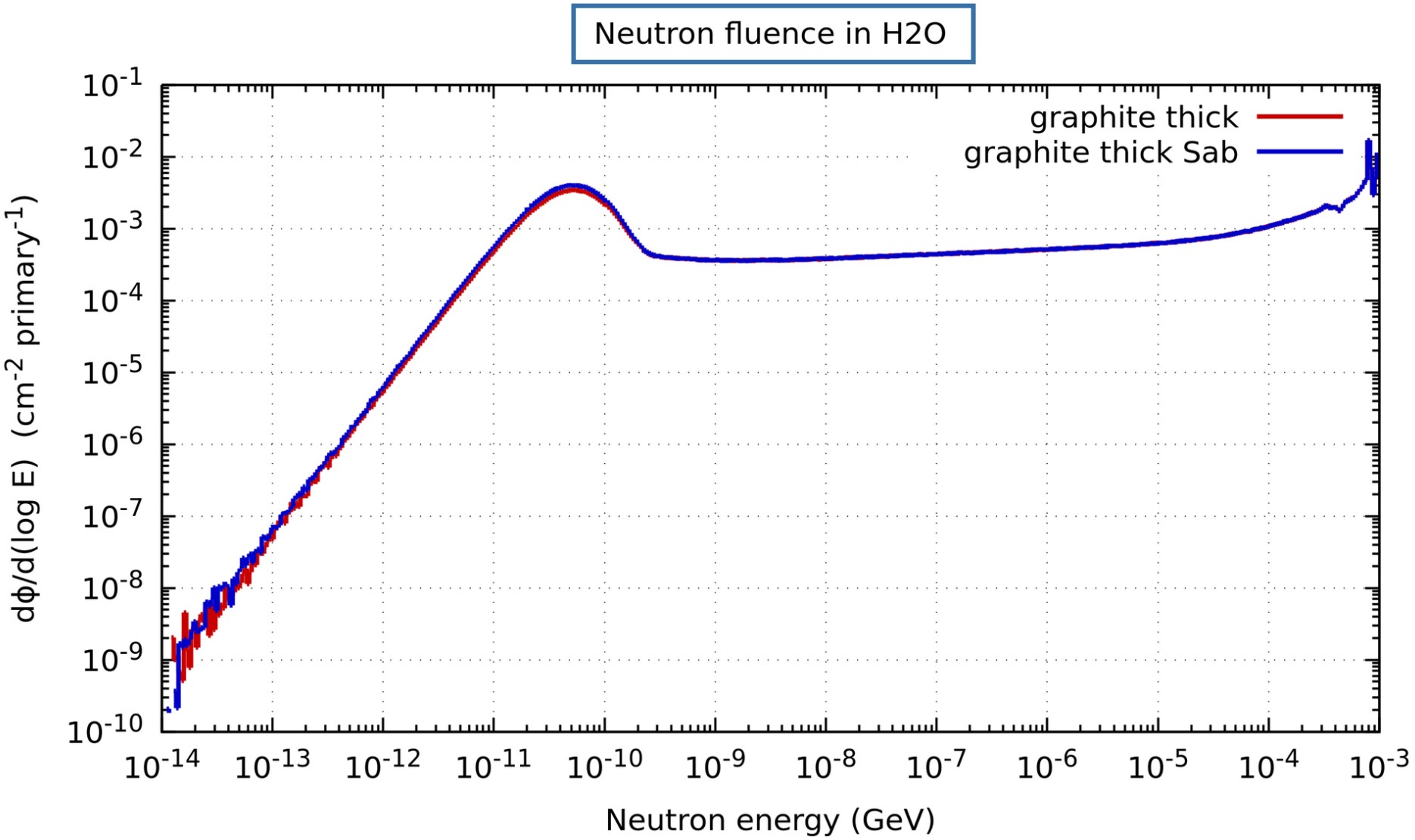
- Conditional to the preprocessor variable `binding` (as well as `pw`) being active:
  - Add a **`LOW-PWXS`** card to select graphite binding environment for `CARBON`

- Add `run/graphitebinding` with `pw`, `graphite`, and `binding` active

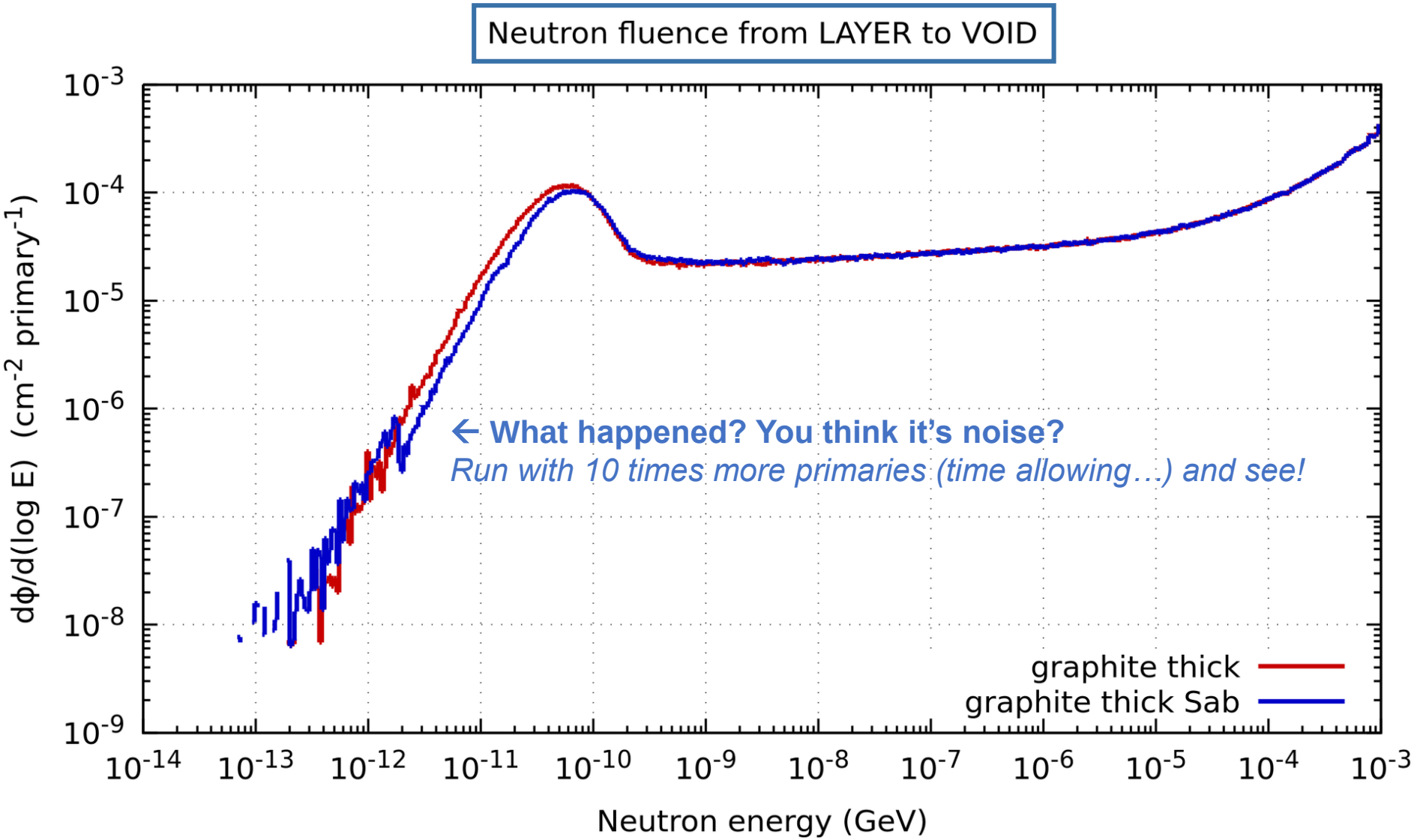
- Run! Process!
- Add the n fluences to the two plots (maybe untick the other plots to resolve better)

- What happened?

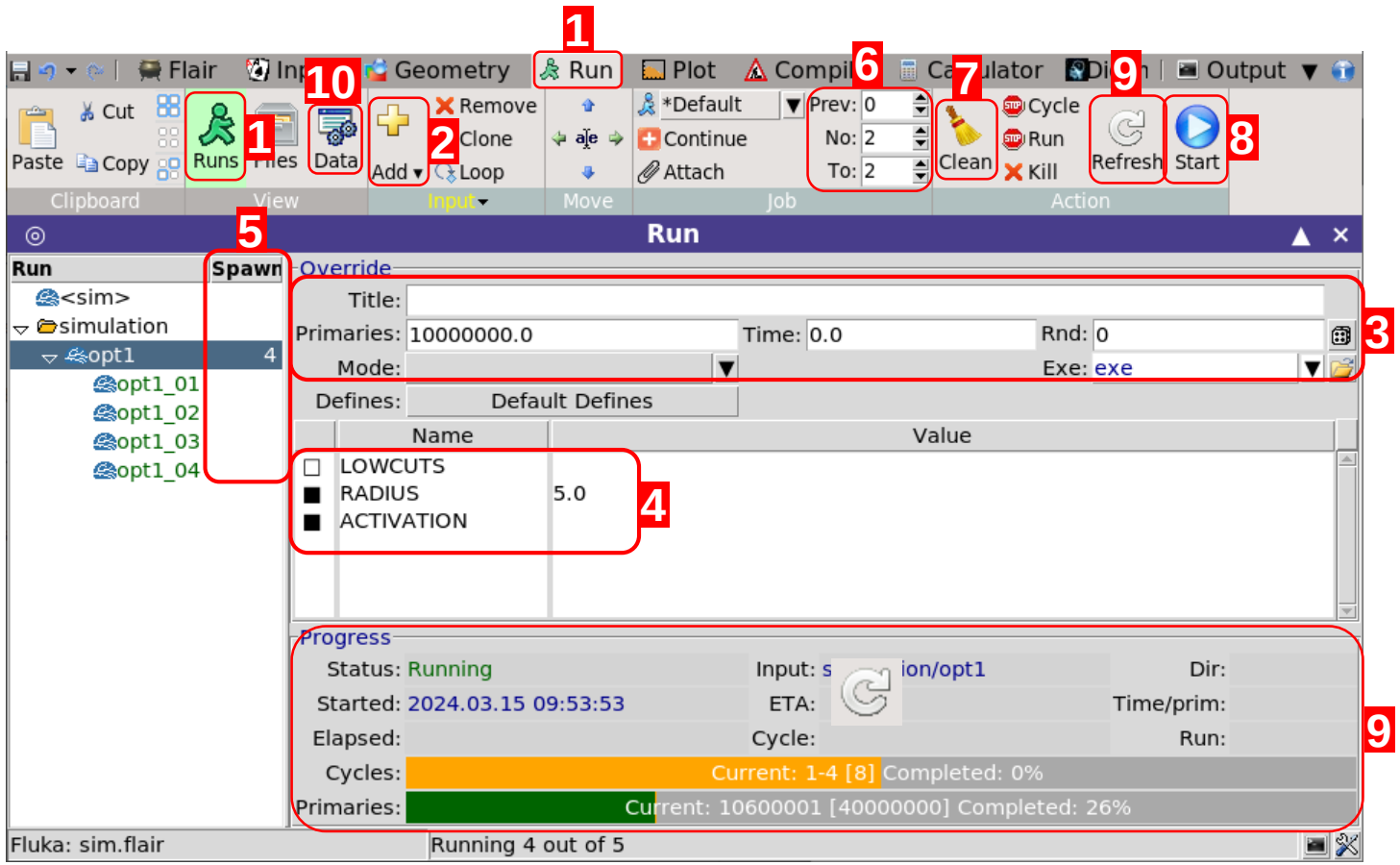
05 – Binding effects – Results



05 – Binding effects – Results



Flair Cheat Sheet



Remember!

- You can **STOP** or **KILL** the run.
- You can edit your input while the simulation runs.

!!! WARNING !!!

- Mind the memory and CPU usage of your simulations!

1. Go to the ***Run* tab**, select ***Runs* view**.
2. Add **new folder** + Add **new run**.
3. Override the input run info:
   - **Number of primaries**
   - Title / Max. time per cycle / Seed / Exec.
4. Override/Define variables.
5. **Recommended:** Increase number of spawns
6. Set number of cycles per spawn
   - Recommend at least 5 cycles in total.
   - `num_cycles_tot = num_cycles_per_spawn * num_spawns`
7. ***Clean*** run files after change to input or run settings.
8. Click ***Start*** to launch the simulations.
9. Monitor the progress. Click ***Refresh*** to force update.
10. After all cycles end:
    - Go to the ***Data*** ( ) tab.
    - Click ***Process*** ( ) to combine all cycles and create simulation data files.
    - You may need to refresh ( ) and scan ( ) if detectors are missing.


FLUKA