



McStas Union components

What they are and how to use them

PRESENTED BY MADS BERTELSEN
2021-03-03



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 823852

Introduction

Who am I?

- Worked with McStas last 9 years, now McStas developer
- PhD from University of Copenhagen under Kim Lefmann in 2017
- Now Post Doc at ESS Data Management and Software Centre, funded by PaNOSC and HighNESS
- Writes software, happy when someone wants to use it and learn about my projects!



guide_bot Automatic guide optimization given requirements and overall geometry
(MATLAB)

Union components Modular samples, multiple scattering within
McStas

McStasScript Flexible McStas Python API



Introduction

Difficulties with realistic samples

- Sample holders with complicated geometry
- Consists of many different materials
- No clear order in which neutrons would pass through
- Could have co aligned crystals
- Twinned crystals
- It could all be in a sample environment



Photo by Pia Ray Jensen

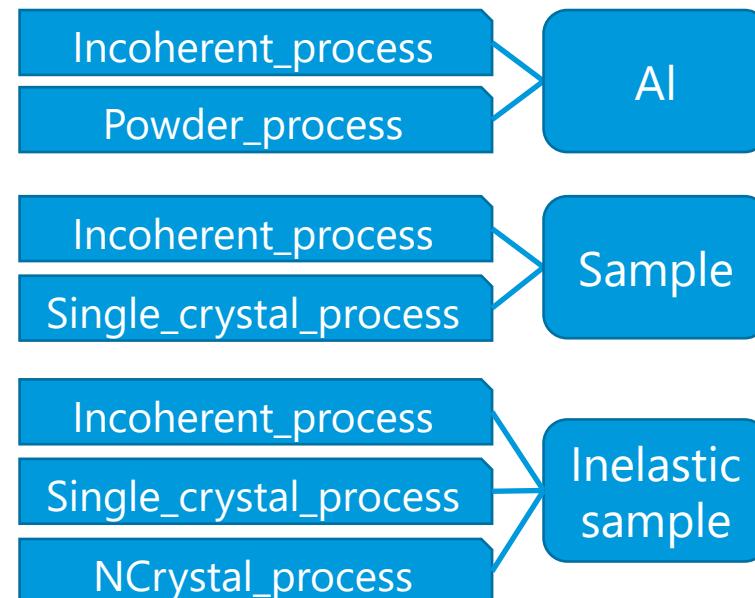


Physics

Modular physics description

- Combine simple physical processes
 - Incoherent scattering
 - Powder Bragg scattering
 - Single crystal Bragg scattering
 - Simple phonon
 - ...
- Create material definitions by combining an arbitrary number of these processes and an absorption cross section

Processes Materials



Physics

Modular physics description

```
COMPONENT Al_incoherent = Incoherent_process(  
    sigma = 4*0.0082, packing_factor=1,  
    unit_cell_volume = 66.4)  
AT (0,0,0) ABSOLUTE  
  
COMPONENT Al_powder = Powder_process(  
    reflections = "Al.laz")  
AT (0,0,0) ABSOLUTE  
  
COMPONENT Al = Union_make_material(  
    my_absorption = 100*4*0.231/66.4,  
    process_string = "Al_incoherent,Al_powder")  
AT (0,0,0) ABSOLUTE
```



Processes Materials



Absorption given as inverse penetration dep

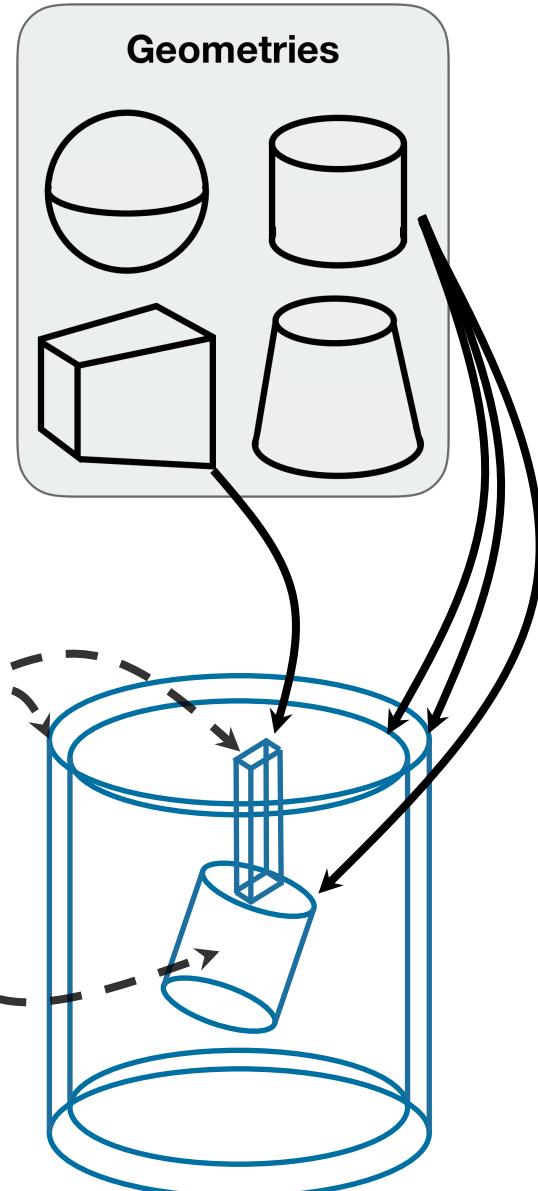
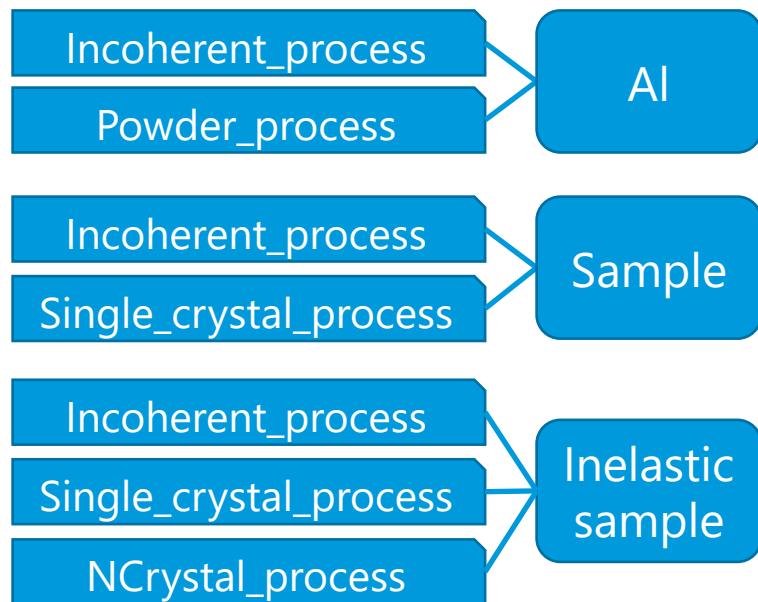
$$\text{my } [1/\text{m}] = \frac{\text{cross section per unit cell}}{\text{unit cell volume}}$$



Geometry

Modular geoemtry

- Construct the desired geometry from simple basic shapes
 - Sphere
 - Cylinder
 - Box
 - Cone
 - CAD model (stl files)
- Assign each geometry a material



Geometry

Priority

- Each geometry is given a priority
- When two geometries overlap, the overlapping region has the material of the highest priority
- This allows the user to create complex geometries



Aluminium
priority = 8.2

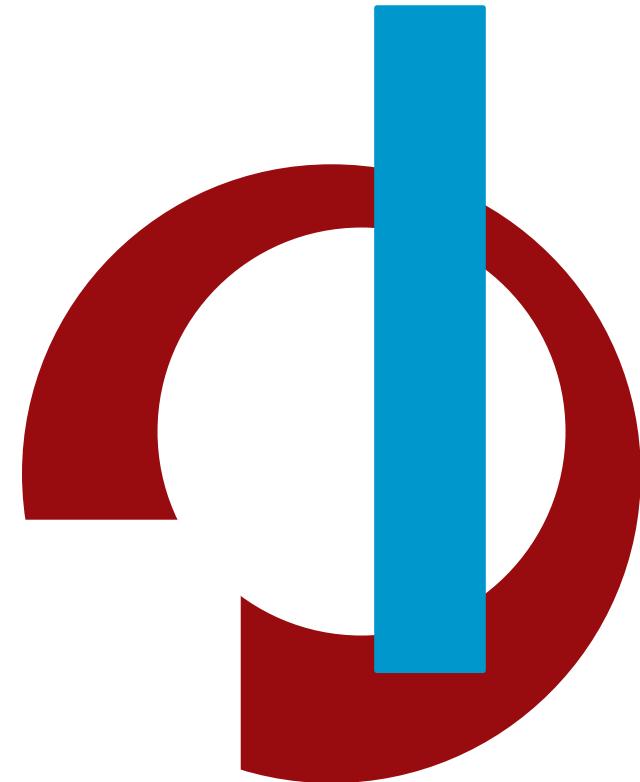
Vanadium
priority = 5.1



Geometry

Priority

- Each geometry is given a priority
- When two geometries overlap, the overlapping region has the material of the highest priority
- This allows the user to create complex geometries



Geometry

Modular geometry

```
COMPONENT cryostat_shell = Union_cylinder(  
    radius = 0.15, height = 0.4,  
    priority = 10, material_string = "AI")
```

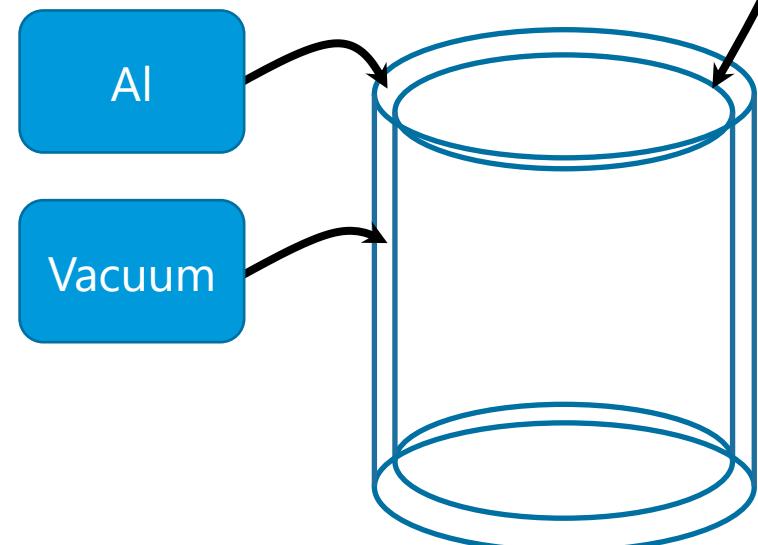
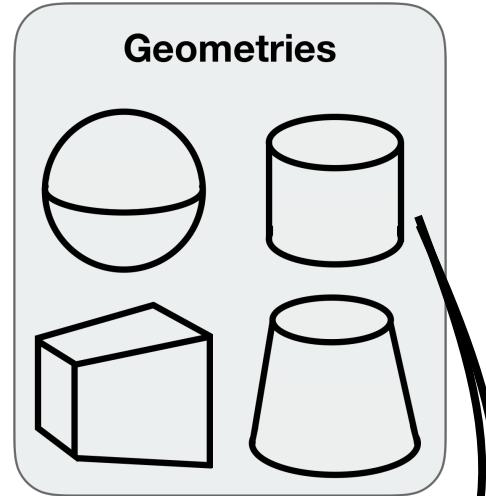
AT (0,0,0) RELATIVE target

ROTATED (0,0,0) RELATIVE target

```
COMPONENT cryostat_vacuum = Union_cylinder(  
    radius = 0.147, height = 0.39,  
    priority = 11, material_string = "Vacuum")
```

AT (0,0,0) RELATIVE target

ROTATED (0,0,0) RELATIVE target



Union master



Where the simulation actually happens

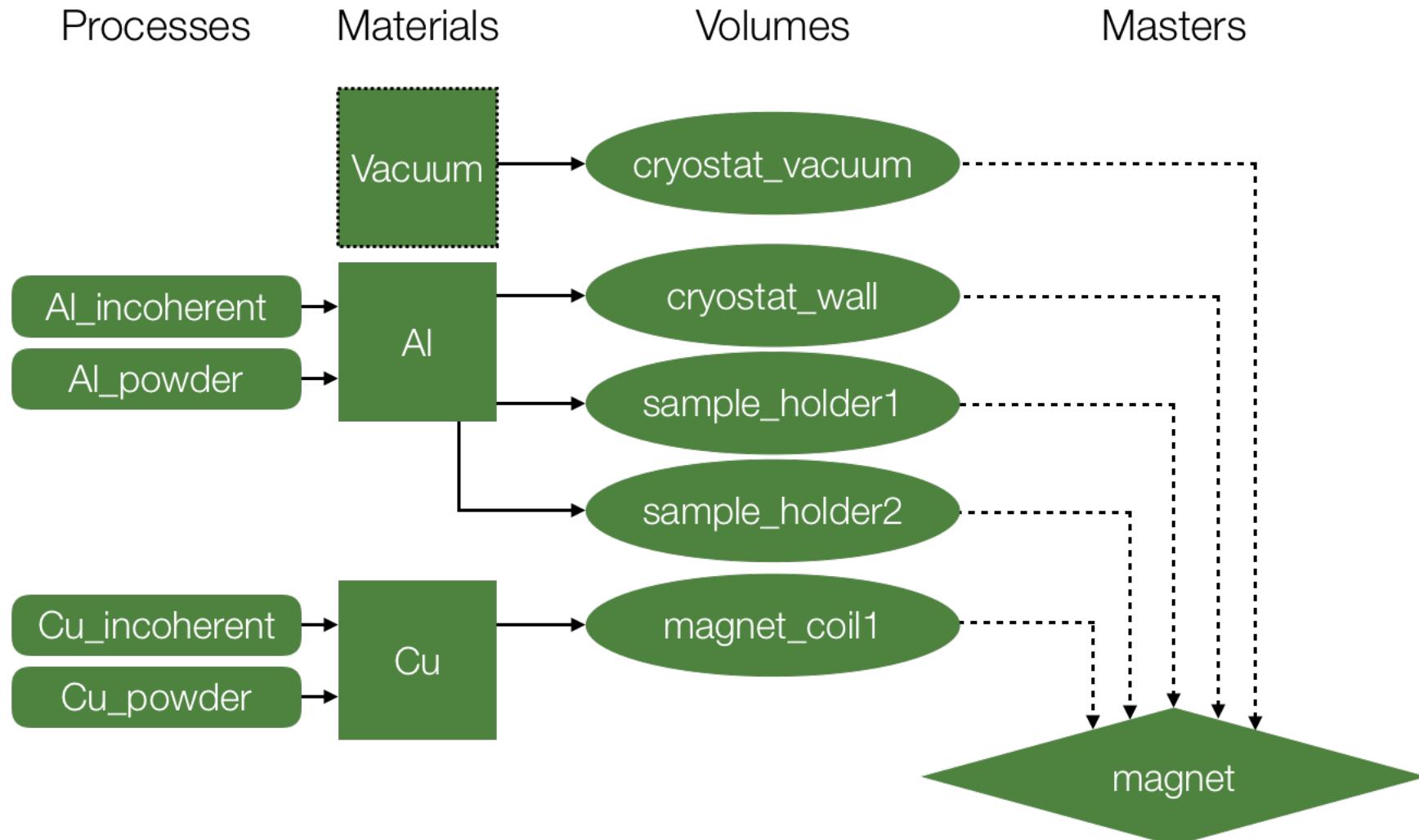
- All components discussed doesn't actually perform any simulation
- Process and geometry components describe the system to be simulated
- The **Union_master** component performs the simulation
- A Union_master must be inserted, and will simulate all Union geometry components defined so far
- No inputs are necessary, as all the information is already in the processes and geometry

```
COMPONENT master = Union_master()  
AT (0,0,0) ABSOLUTE
```



Union master

Overview of a Union setup



Introduction

How does this improve the situation?

- Sample holders with complicated geometry
- Consists of many different materials
- No clear order in which neutrons would pass through
- Could have co aligned crystals
- Twinned crystals
- It could all be in a sample environment

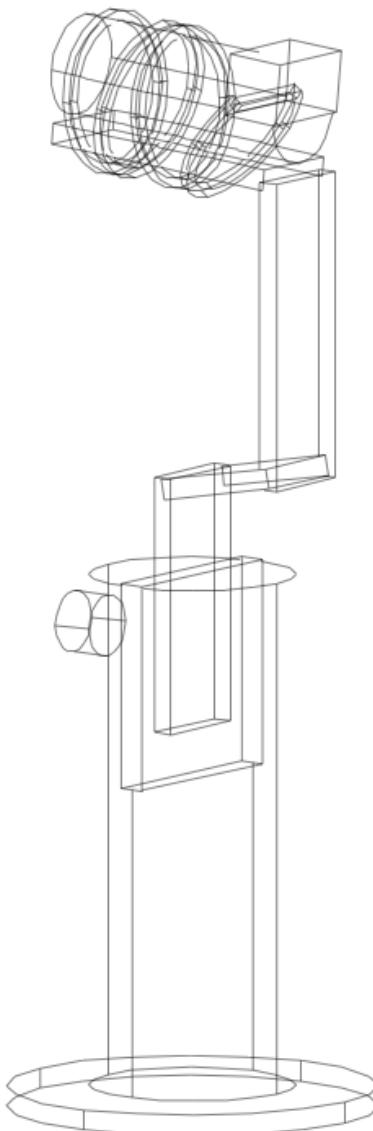


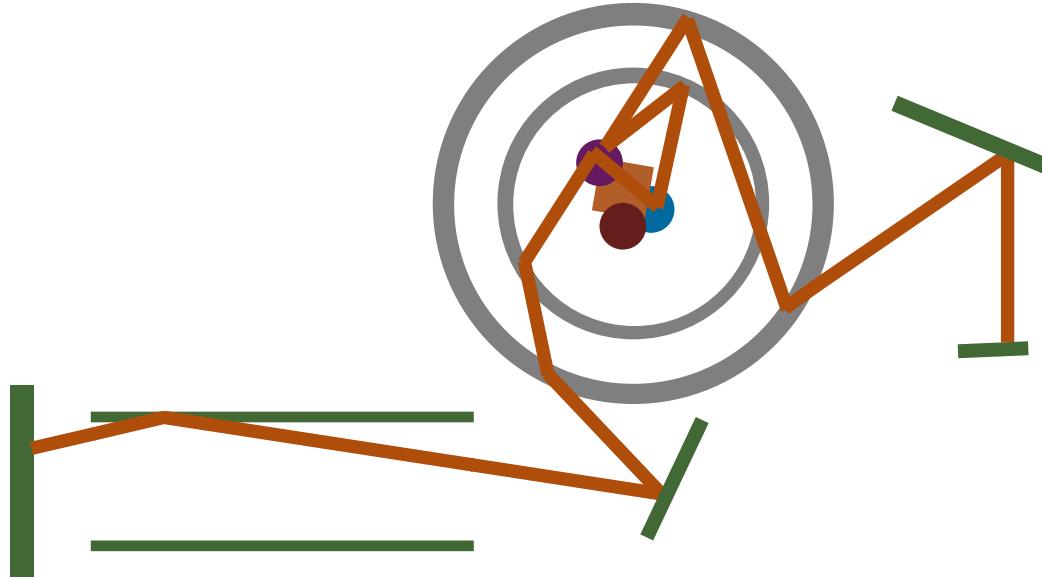
Photo by Pia Ray Jensen



Multiple scattering

Logic in McStas instrument

- Multiple scattering handled in Union_master component
- Neutron handed back to instrument component sequence



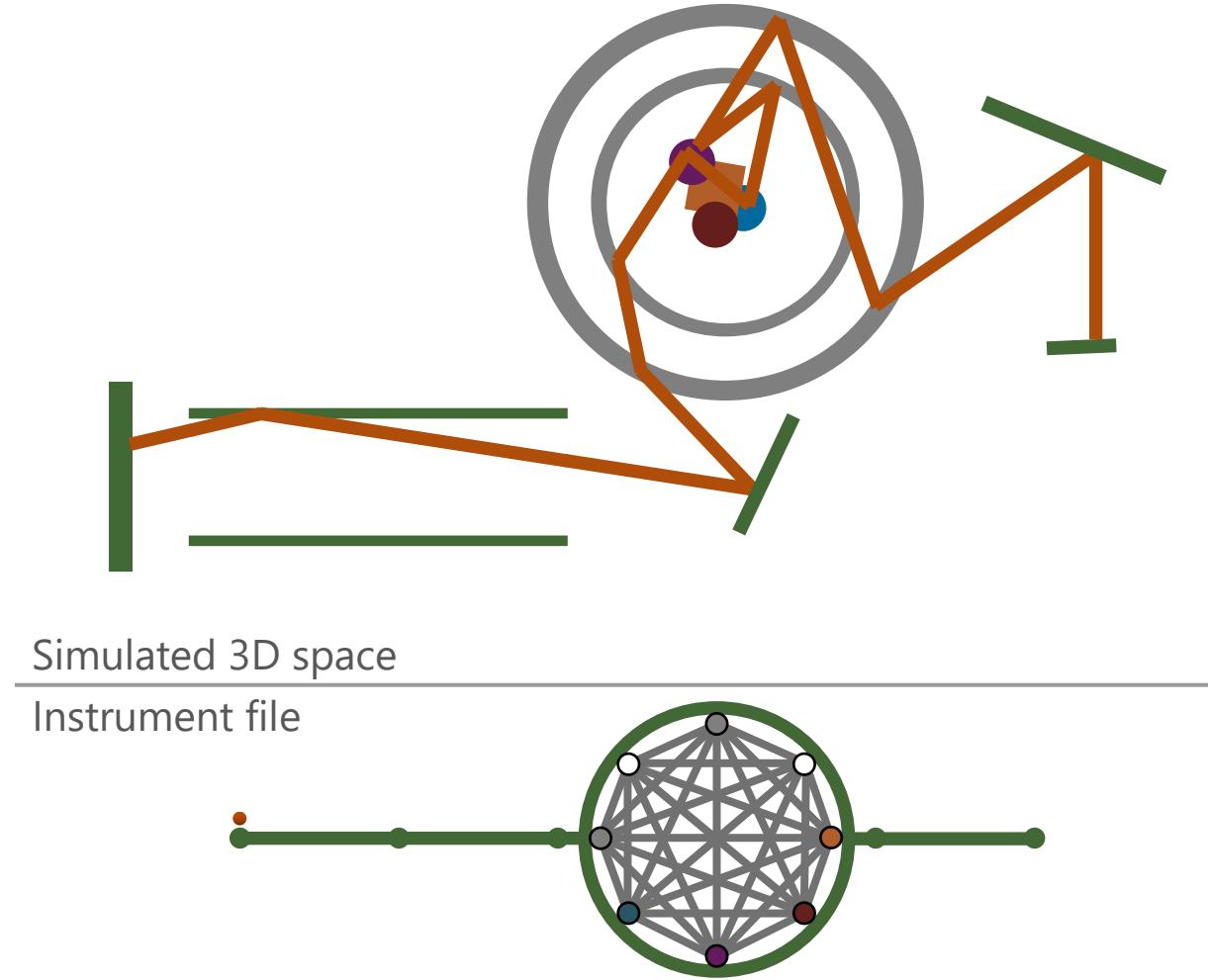
Simulated 3D space
Instrument file



Multiple scattering

Logic in McStas instrument

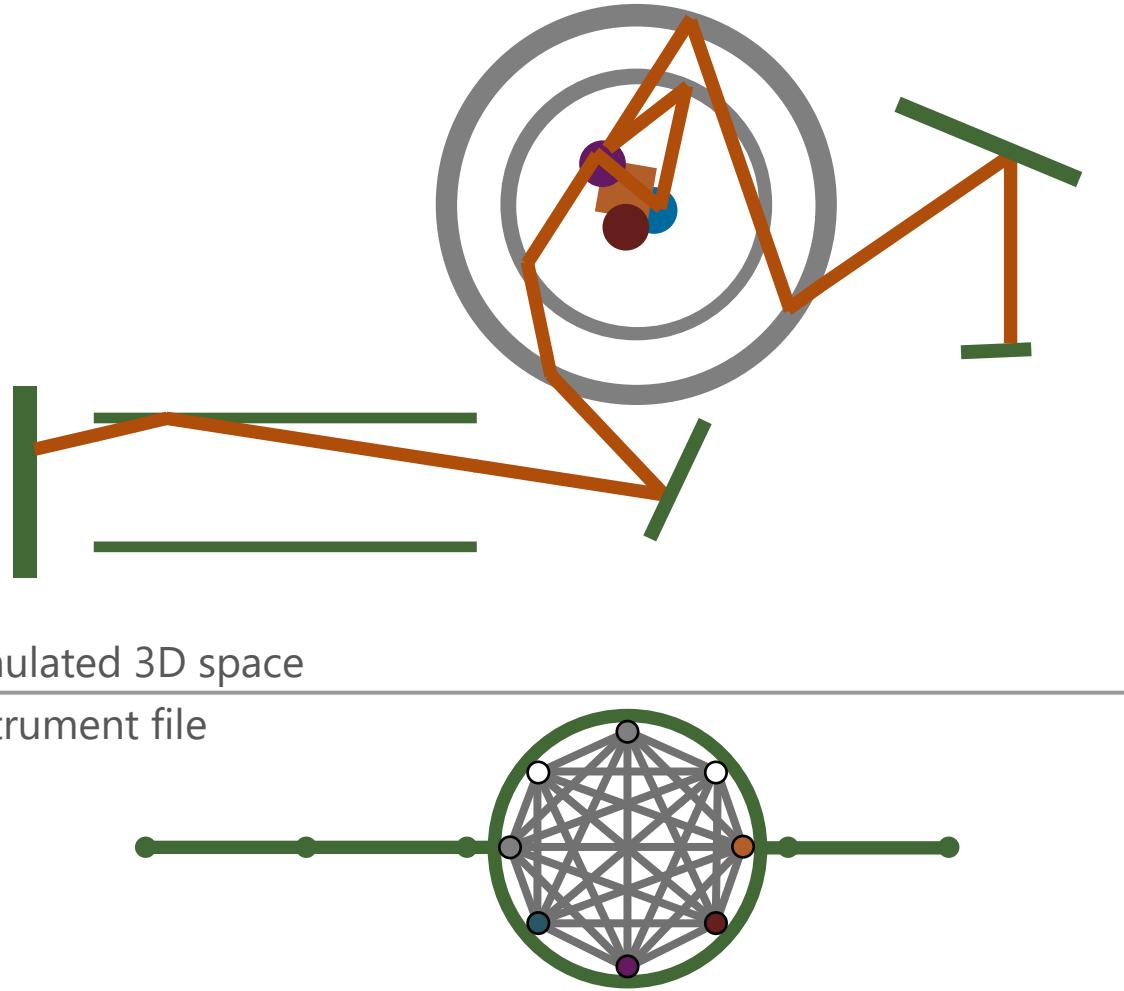
- Multiple scattering handled in Union_master component
- Neutron handed back to instrument component sequence
- Ray can go to geometries in any order



Multiple scattering

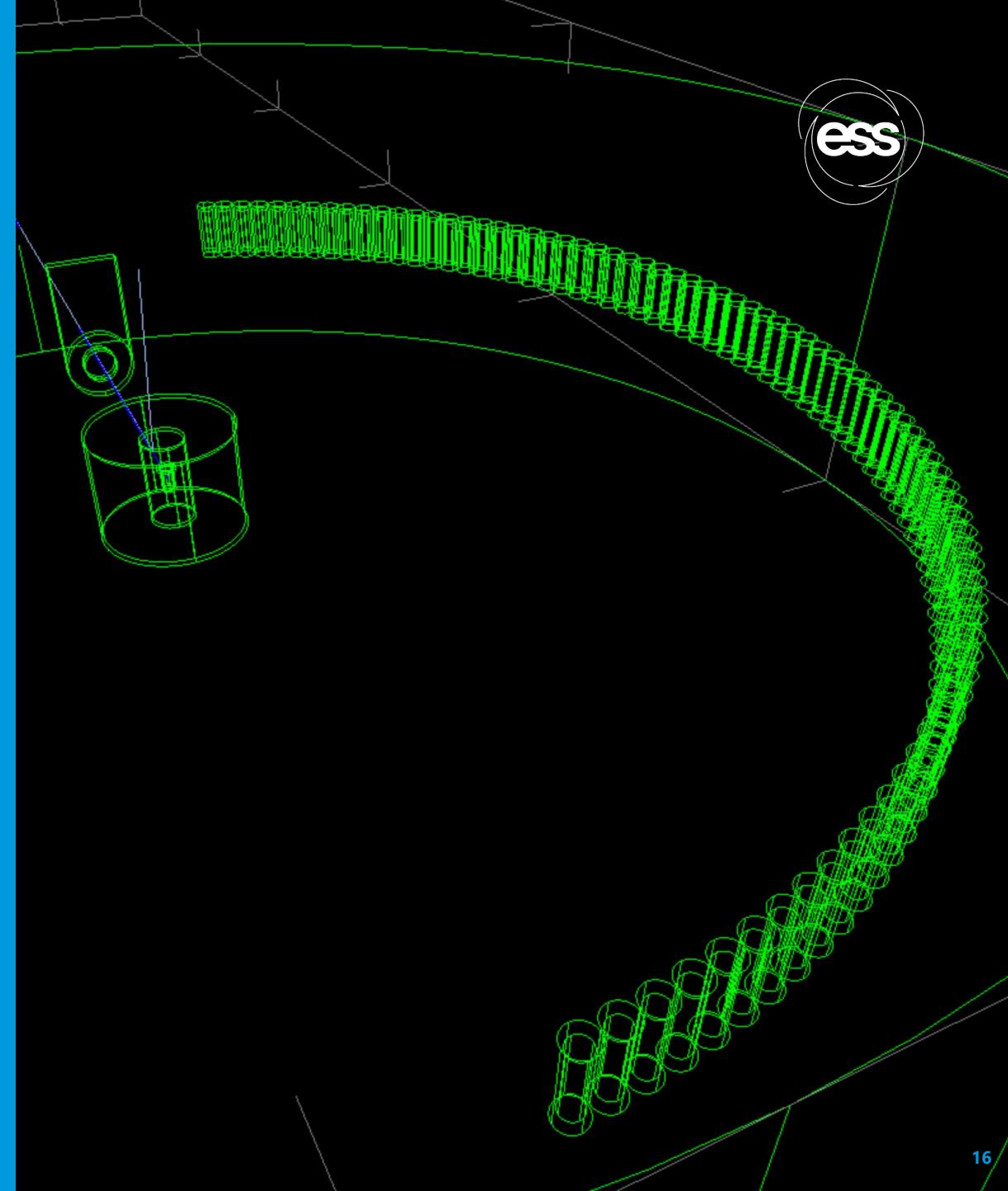
Logic in McStas instrument

- Multiple scattering handled in Union_master component
- Neutron handed back to instrument component sequence
- Ray can go to geometries in any order
- Internal logic network reduce number of intersection calculations, improving execution speed



2

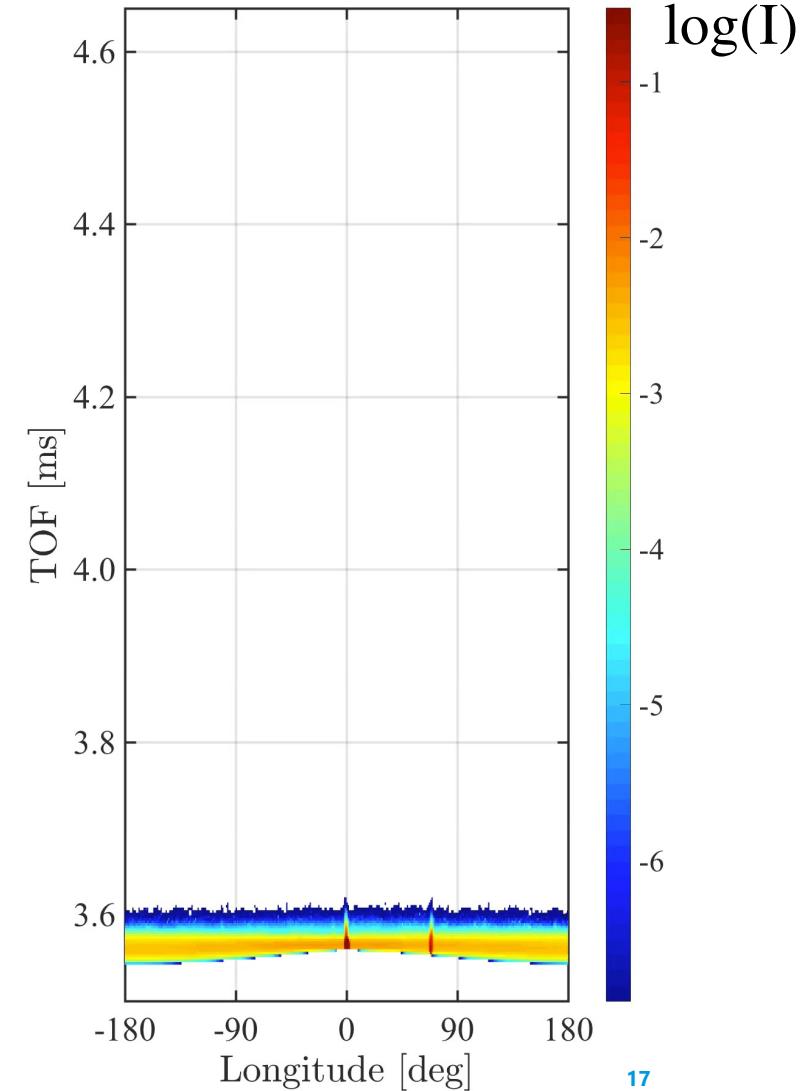
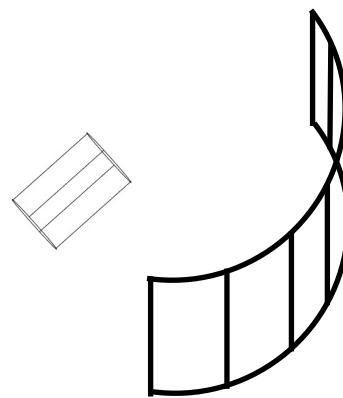
Demonstration



Demonstration

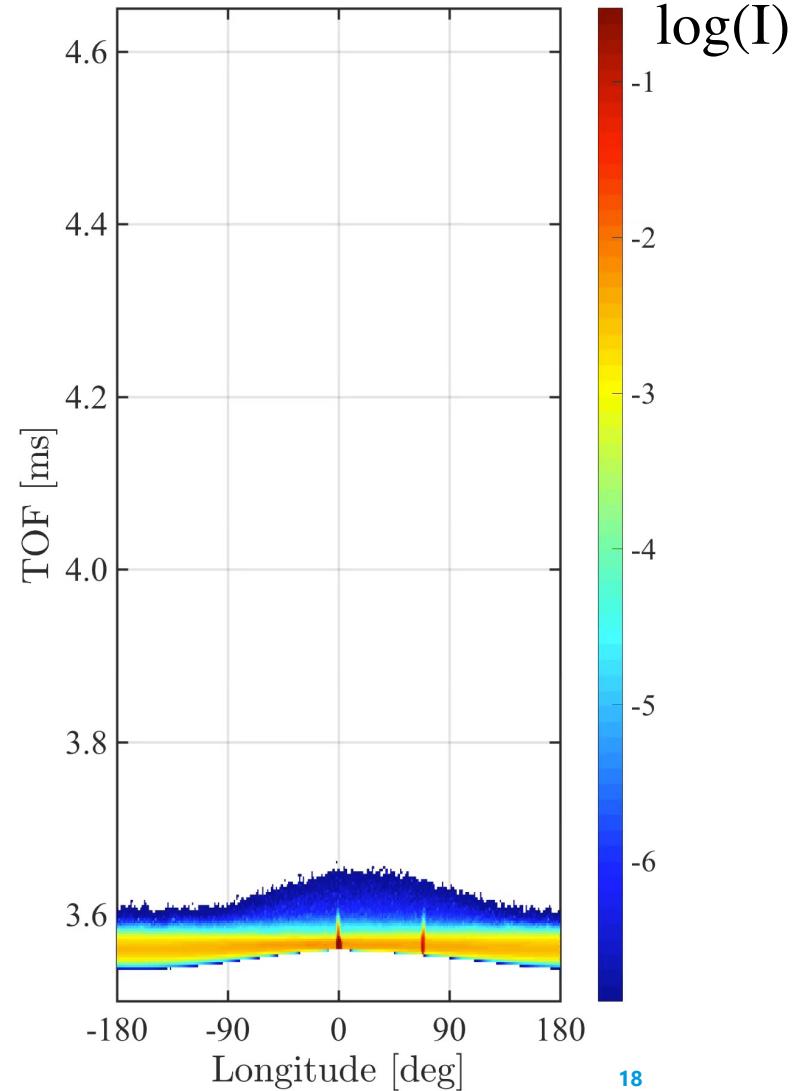
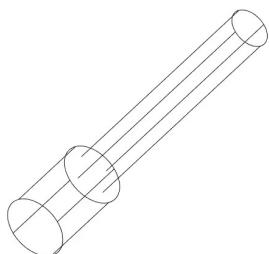
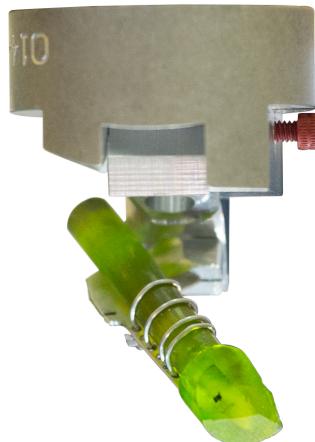


5 meV beam
→



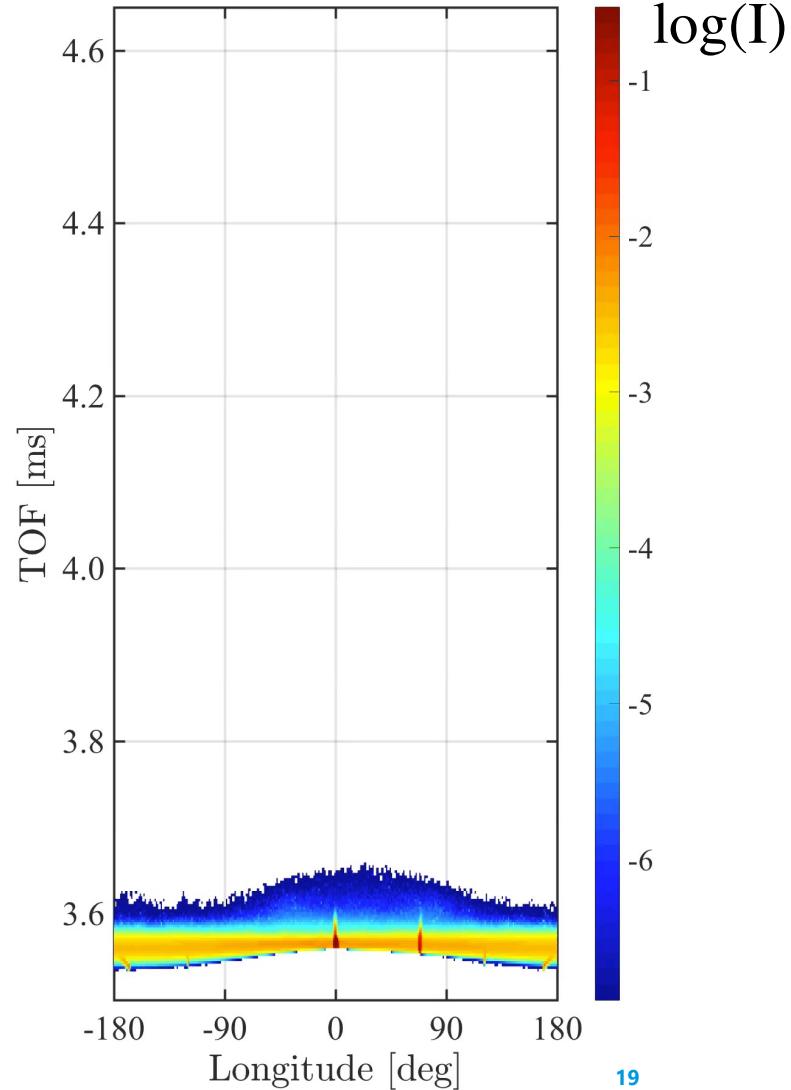
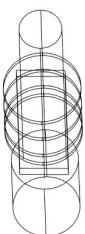
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 823852

Demonstration



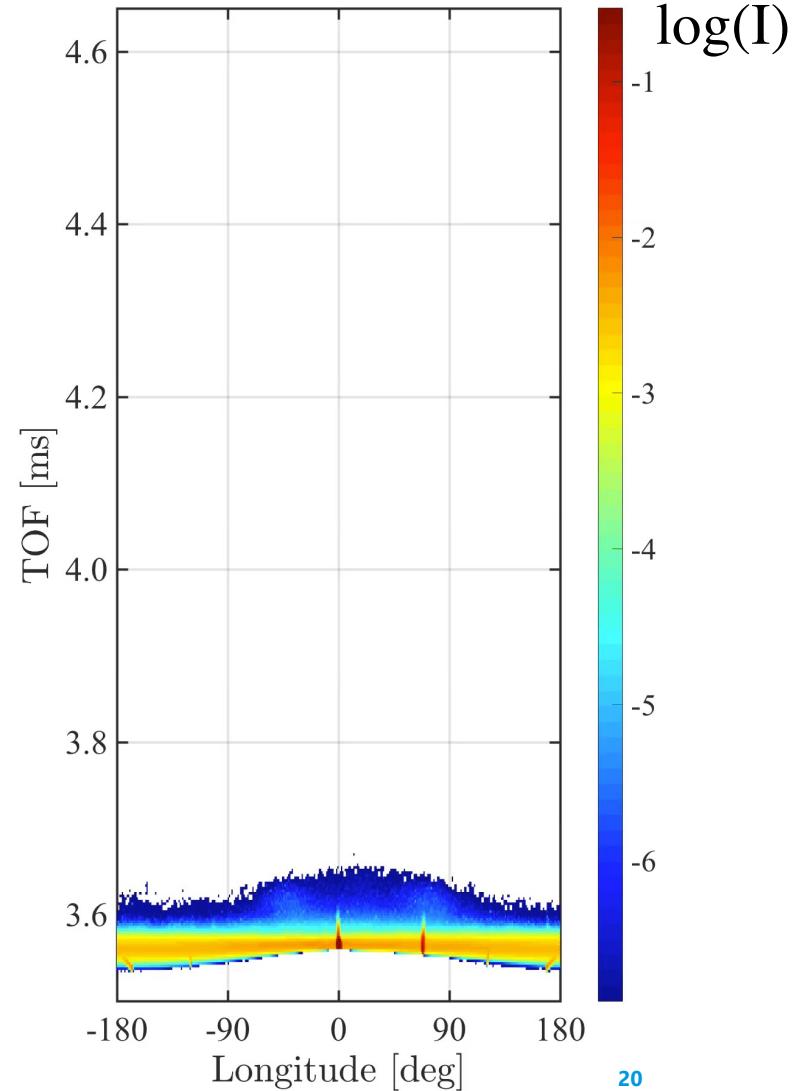
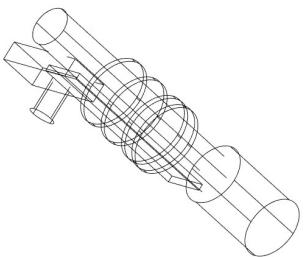
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 823852

Demonstration



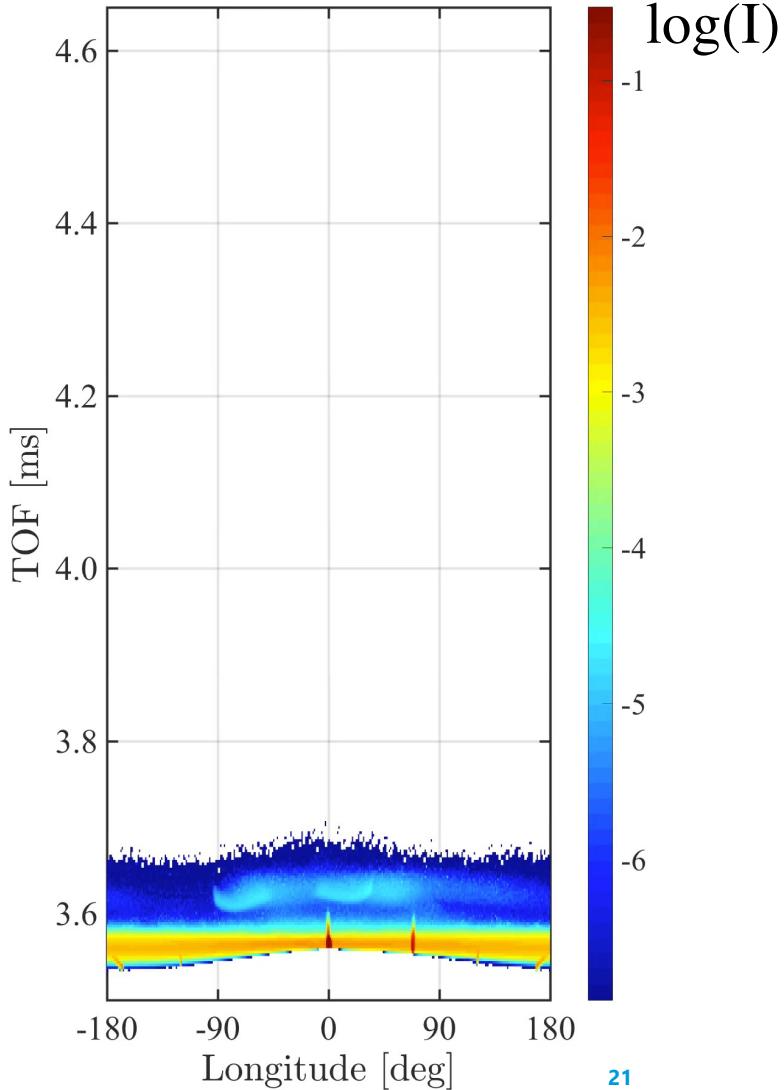
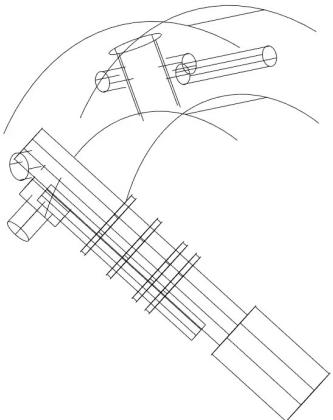
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 823852

Demonstration



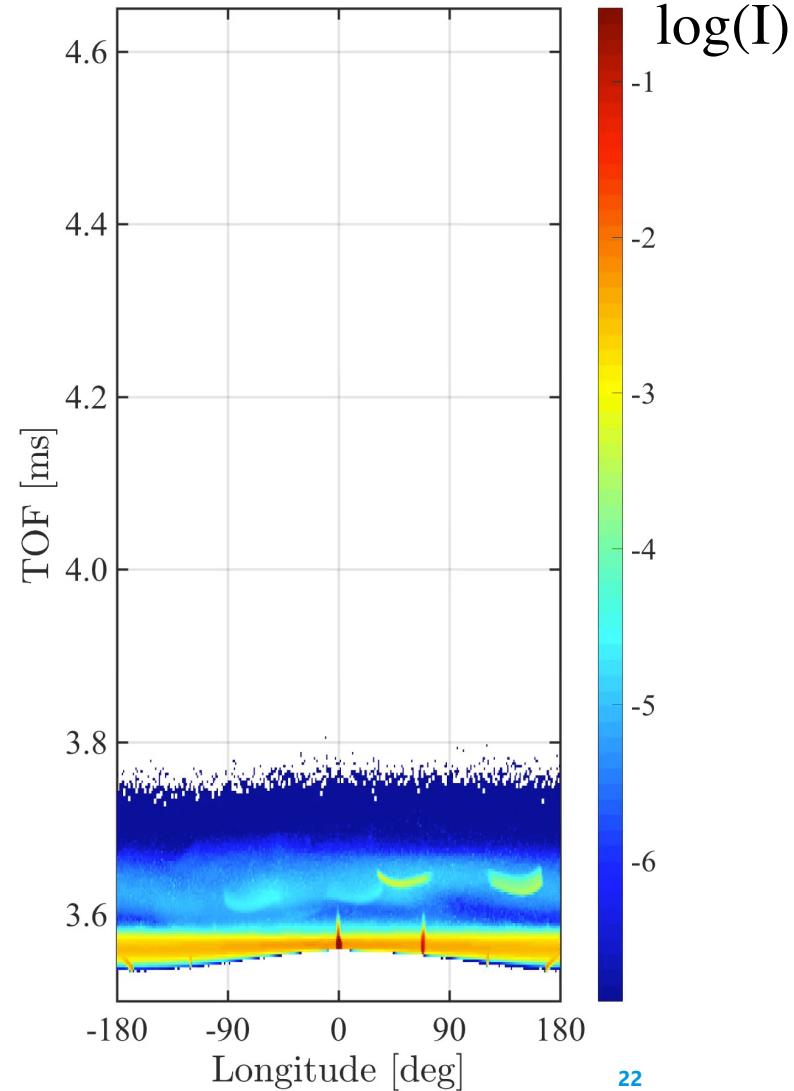
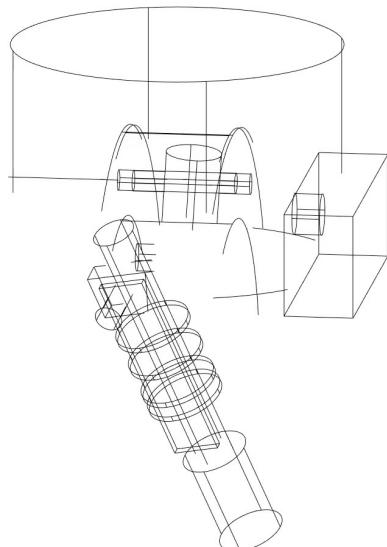
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 823852

Demonstration



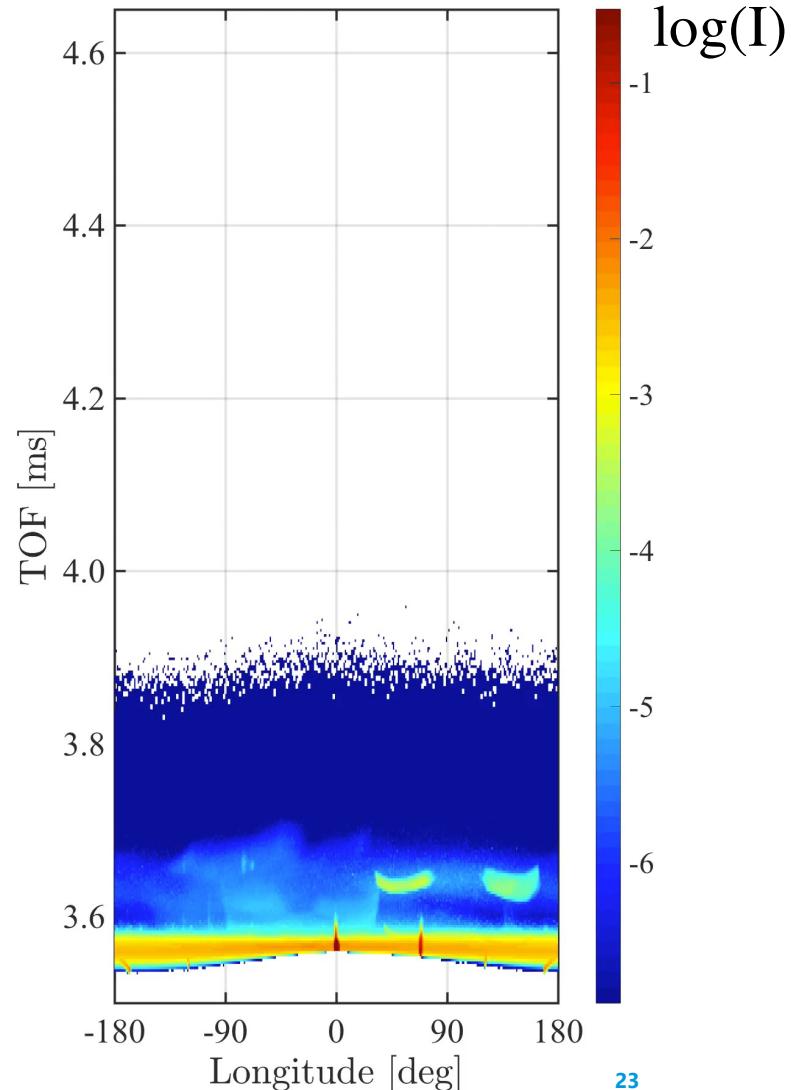
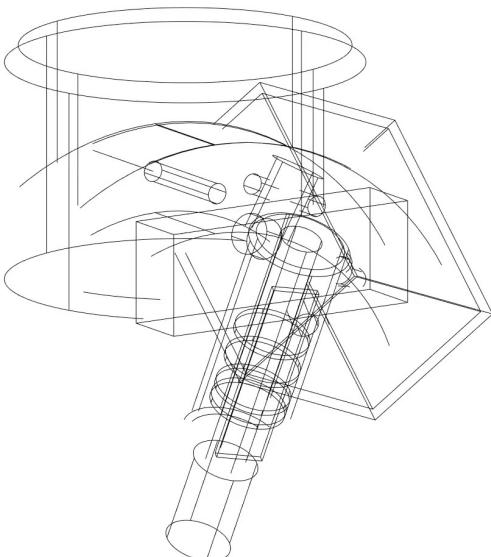
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 823852

Demonstration



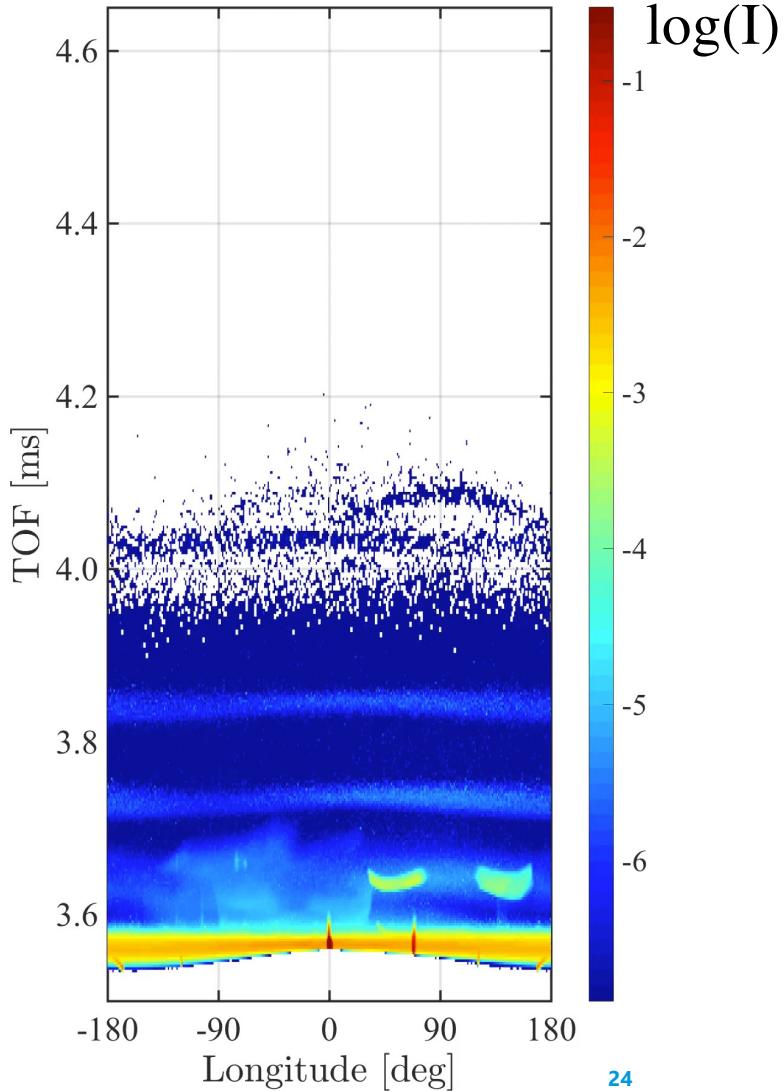
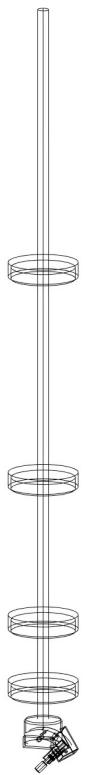
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 823852

Demonstration



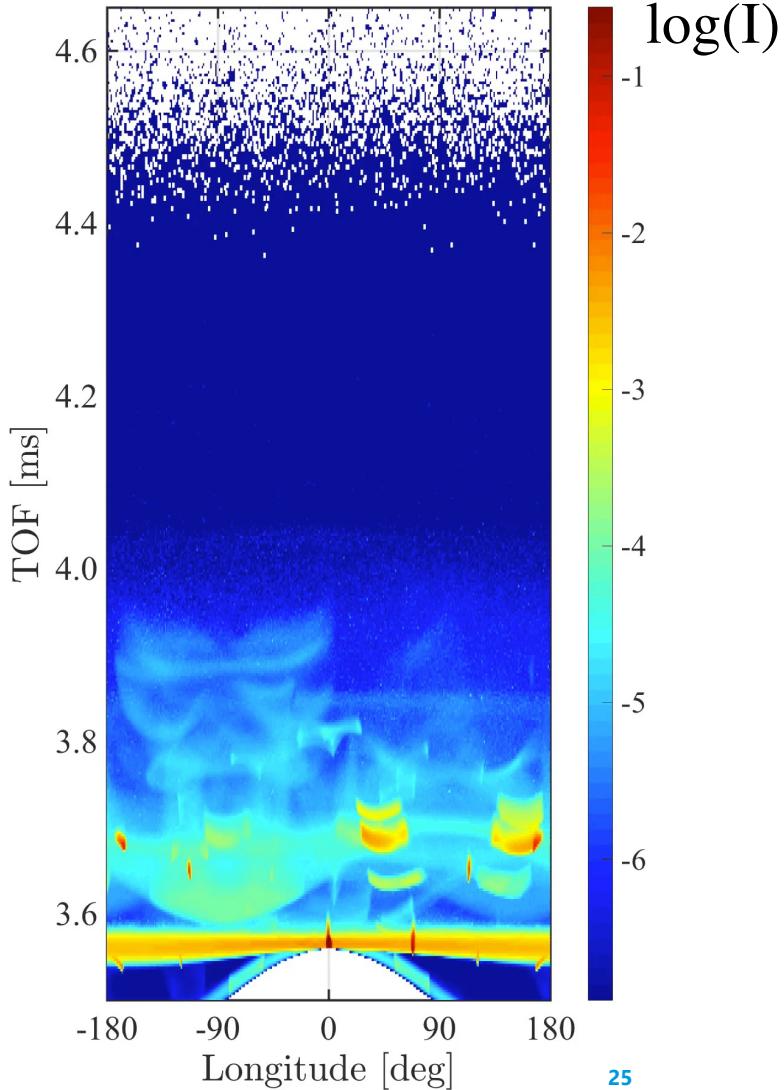
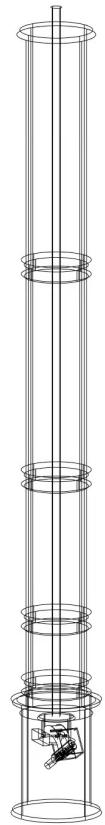
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 823852

Demonstration



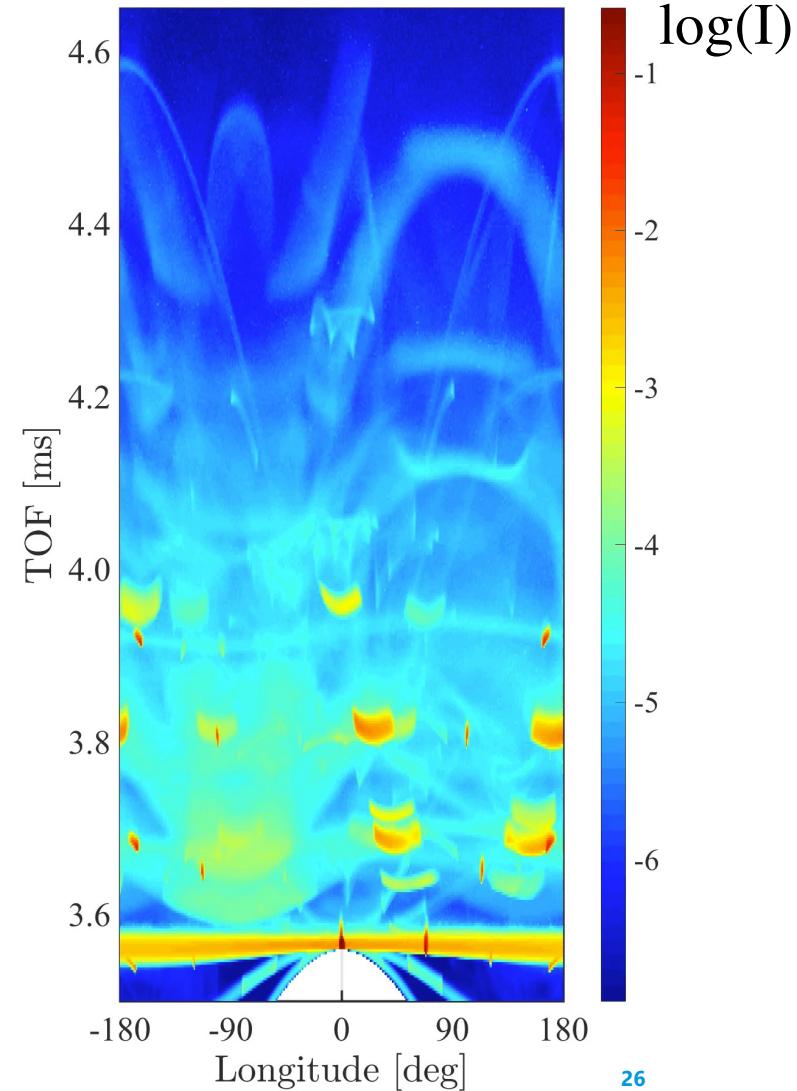
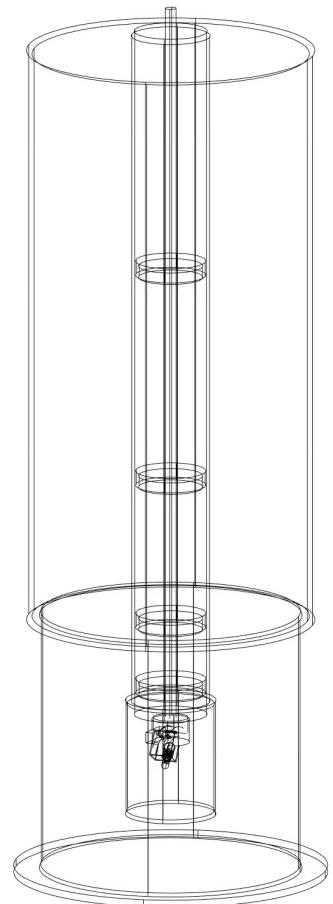
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 823852

Demonstration



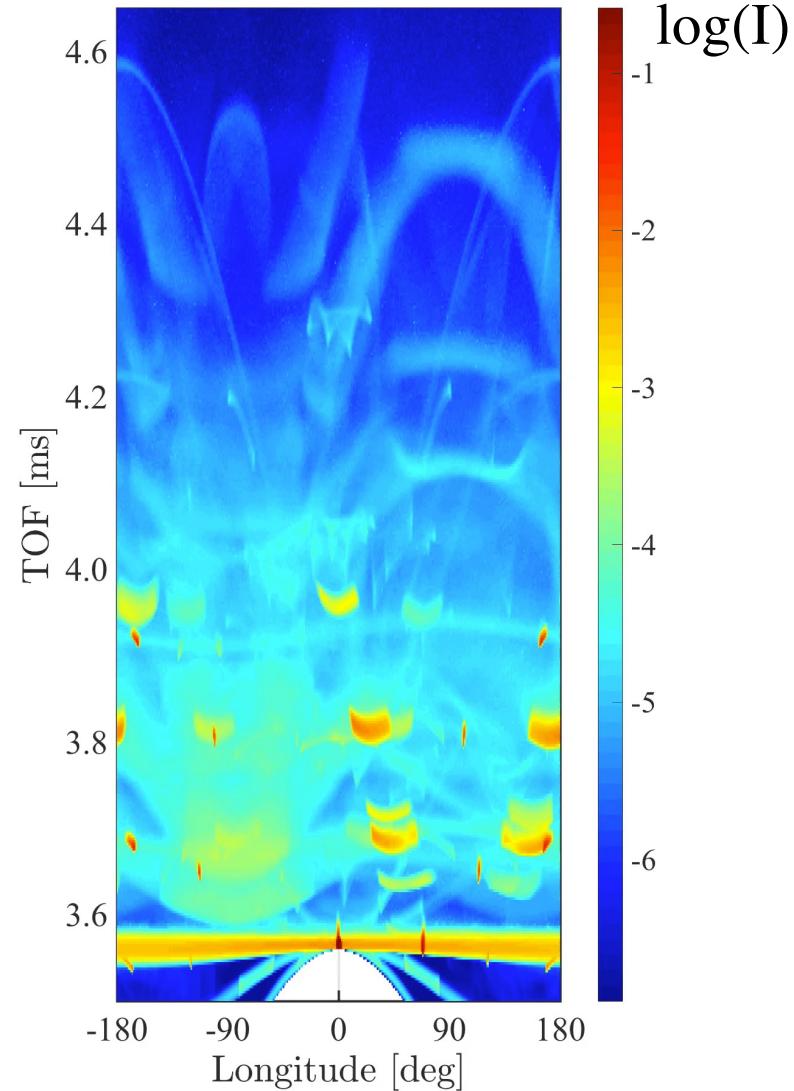
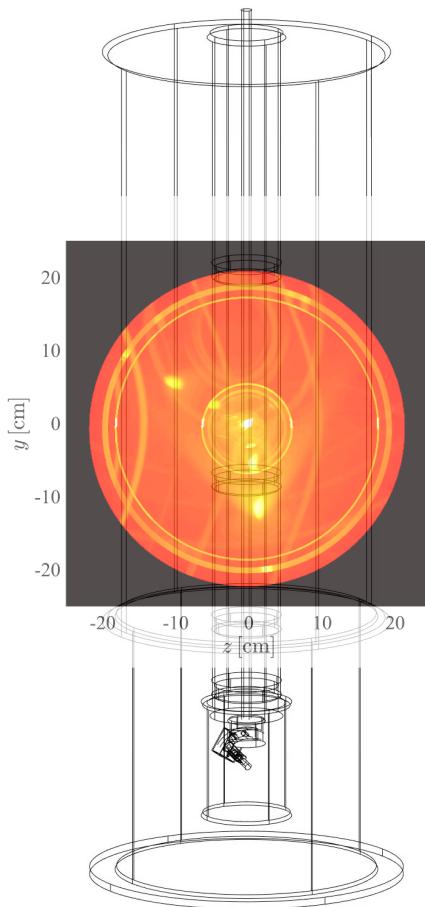
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 823852

Demonstration



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 823852

Demonstration

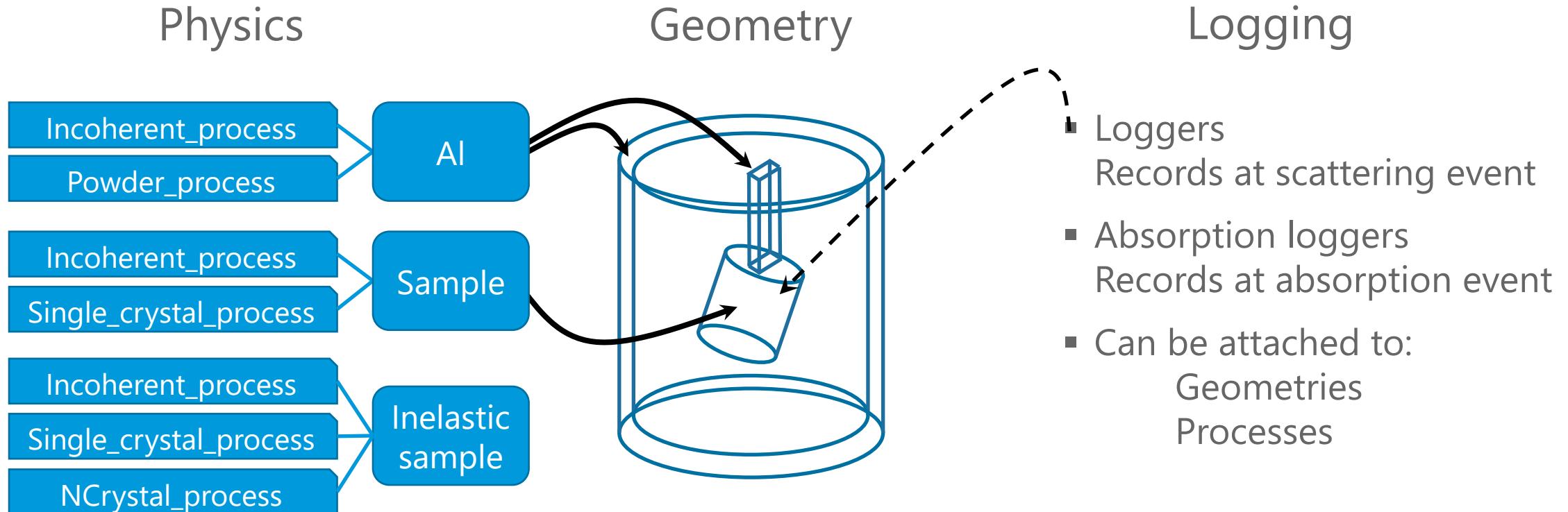


This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 823852

Logger components

Recording information during simulation

- Loggers can be used to understand what is going on during simulation
- Can either log everything, or be attached to a geometry or process



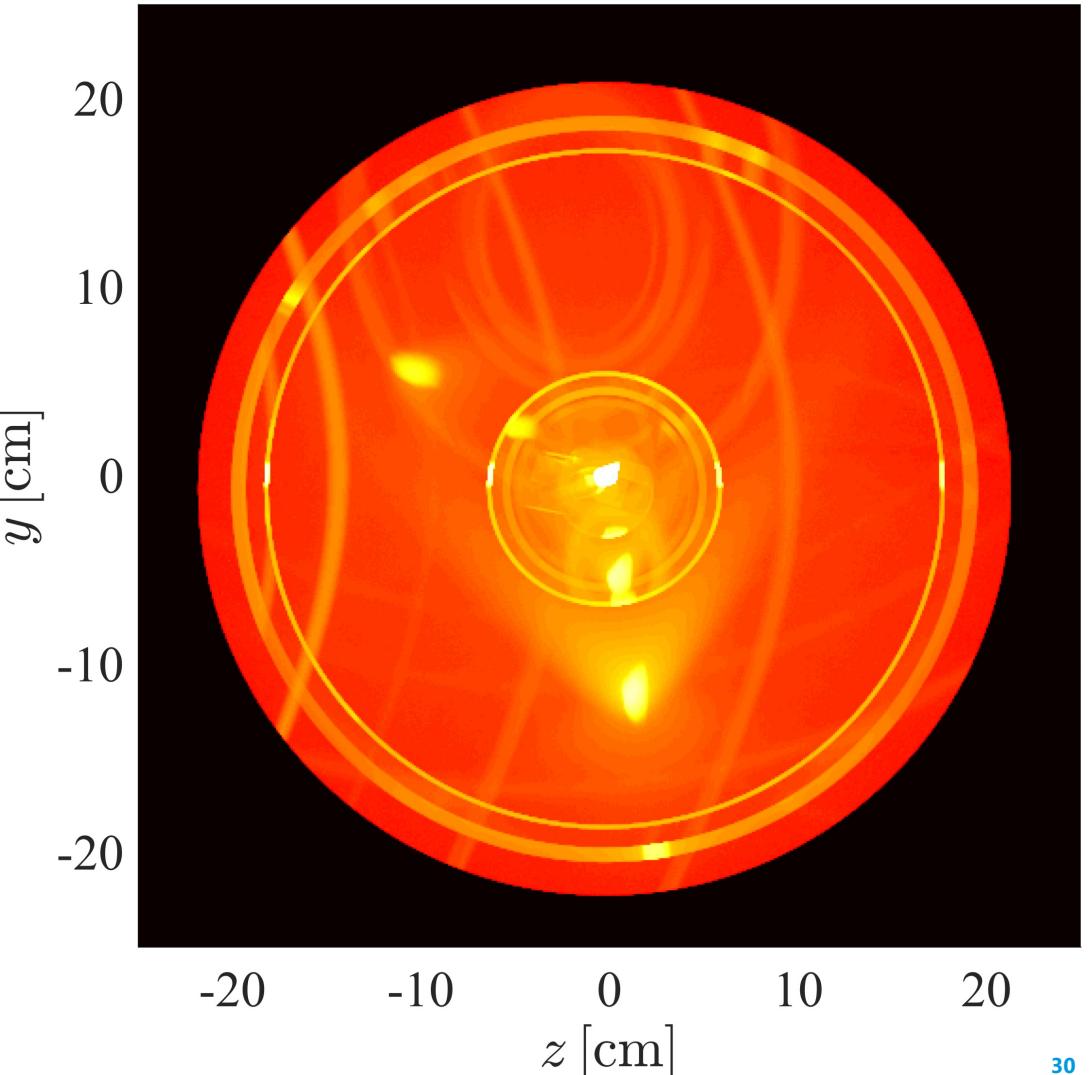
Logger components

Recording information during simulation



- Need to understand the simulation
- Record information during scattering
- Here spatial distribution of scattered intensity from above

```
COMPONENT scattering_zx = Union_logger_2D_space(  
    D_direction_1="z", D1_min=-0.15, D1_max=0.15,  
    n1=300,  
    D_direction_2="x", D2_min=-0.15, D2_max=0.15,  
    n2=300,  
    filename="scattering_zx.dat")  
AT (0,0,0) RELATIVE sample_position
```

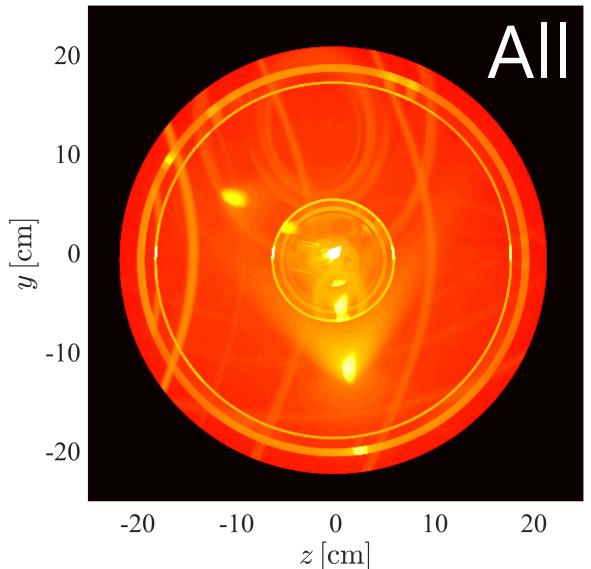
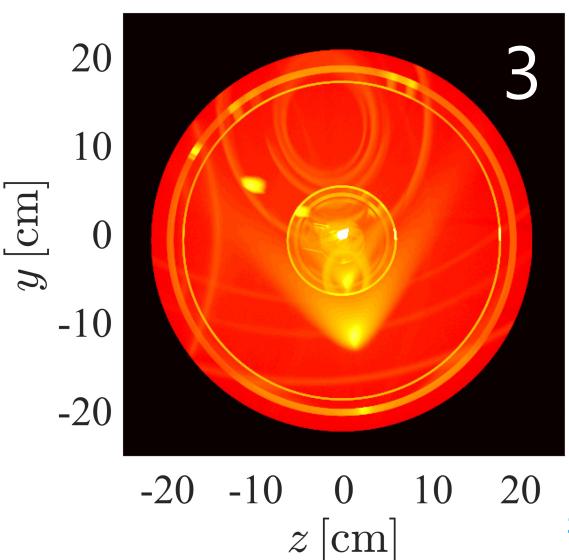
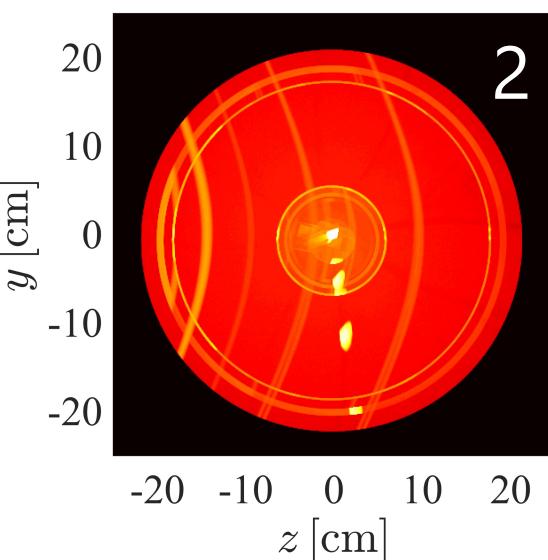
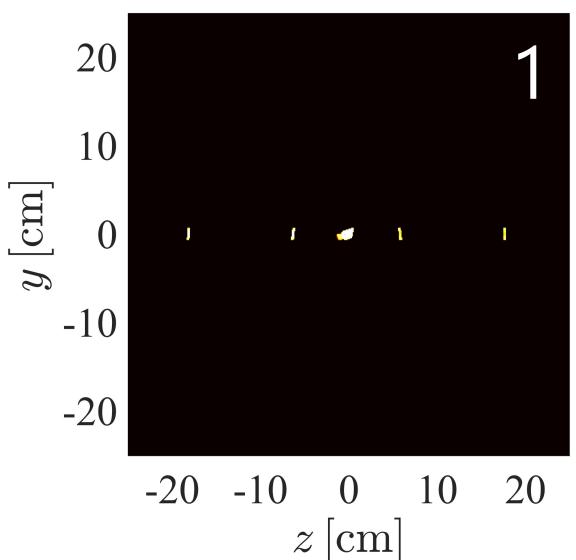


Logger components

Recording information during simulation



- A specific scattering order can be shown
- Improves understanding of what occurred

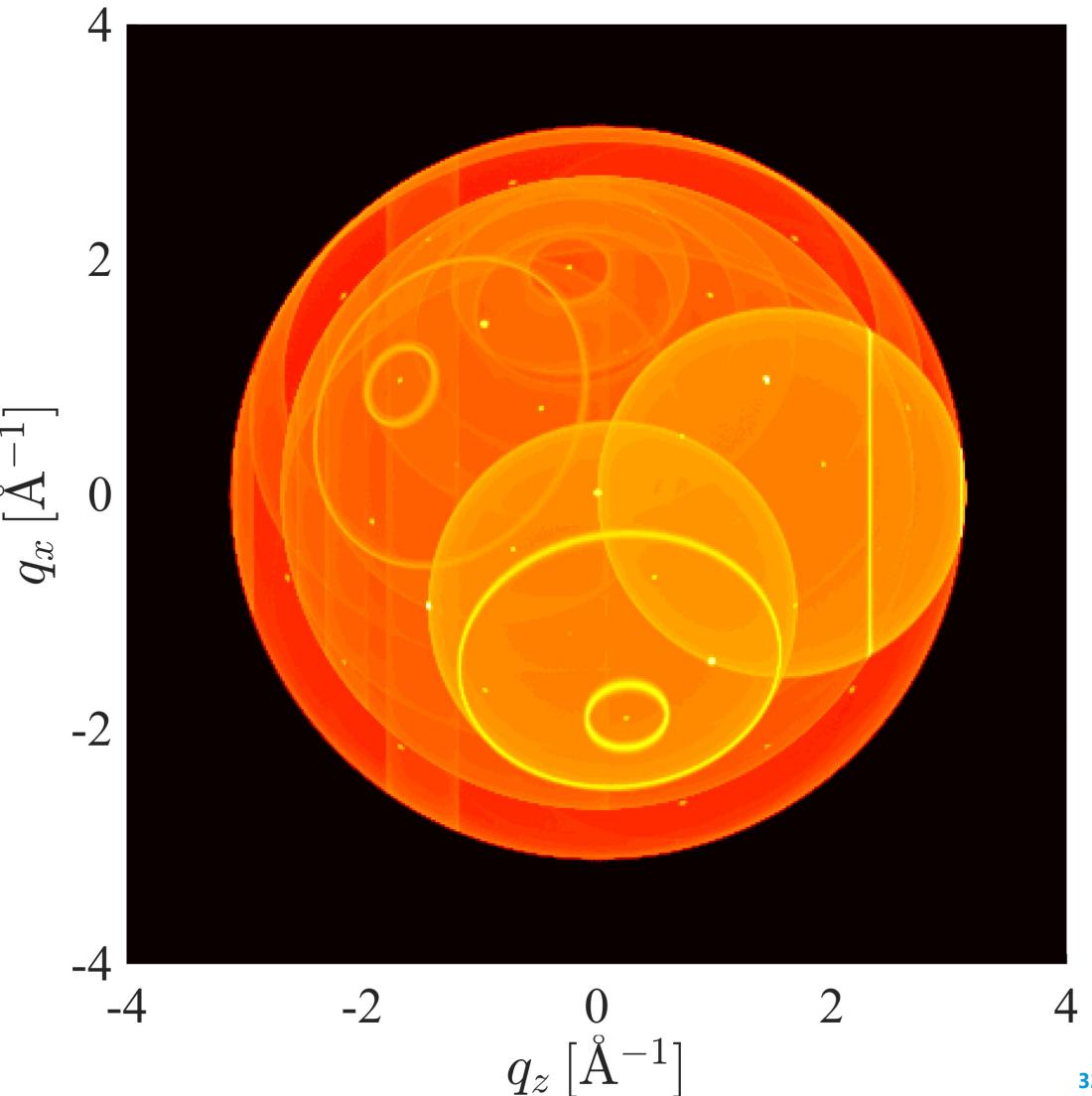


Logger components

Recording information during simulation



- Here the scattering vector is recorded
- Visible Bragg peaks and Debye Scherrer cones

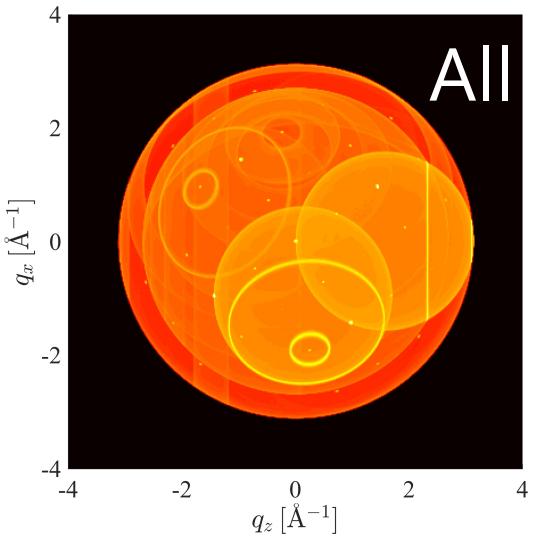
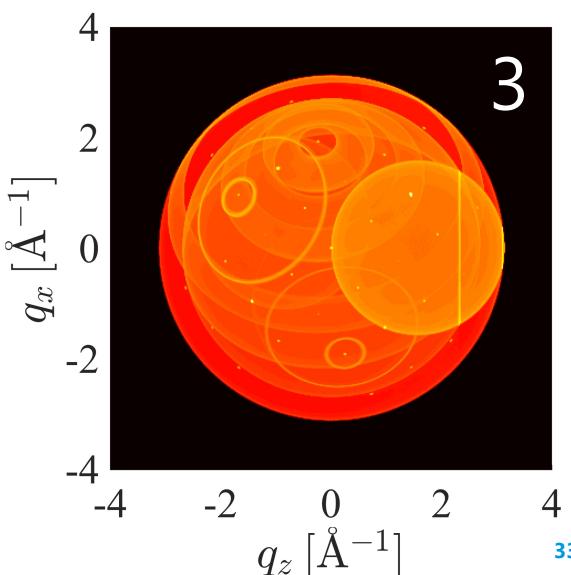
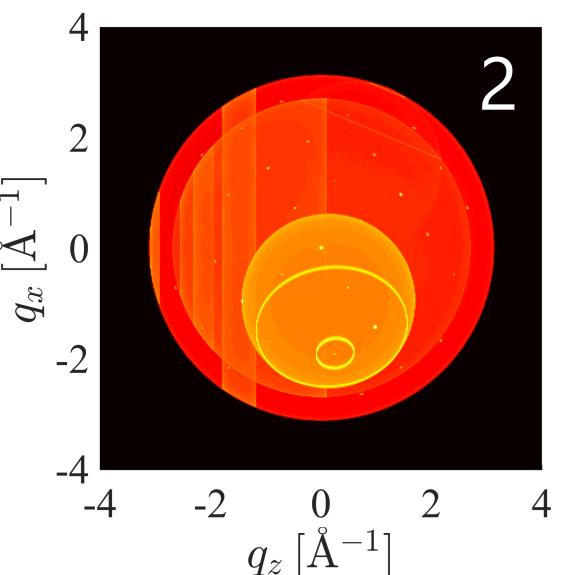
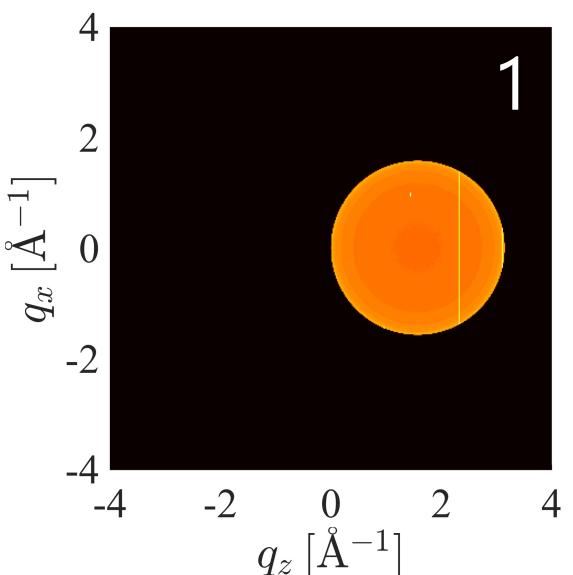


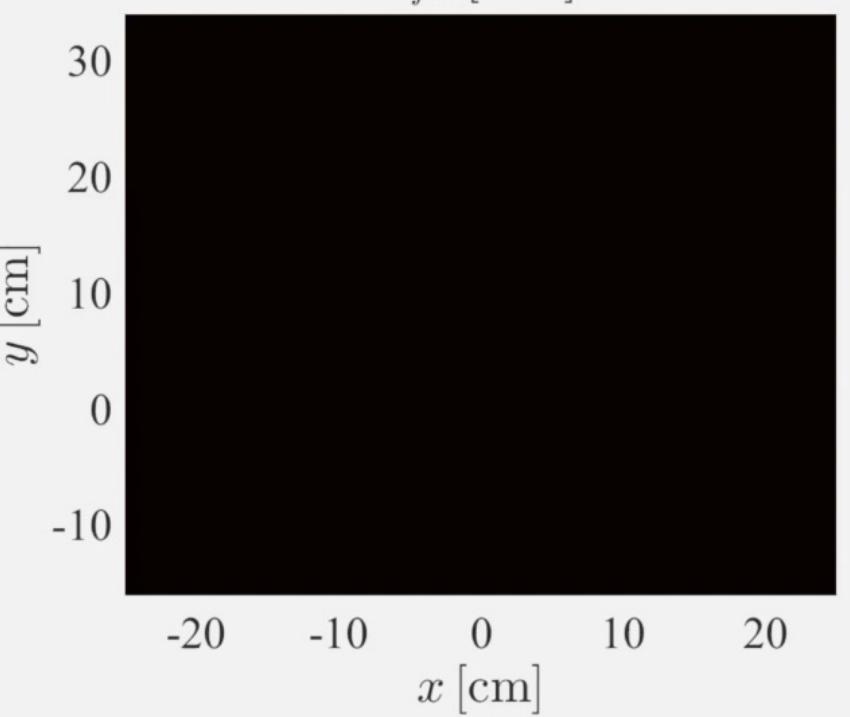
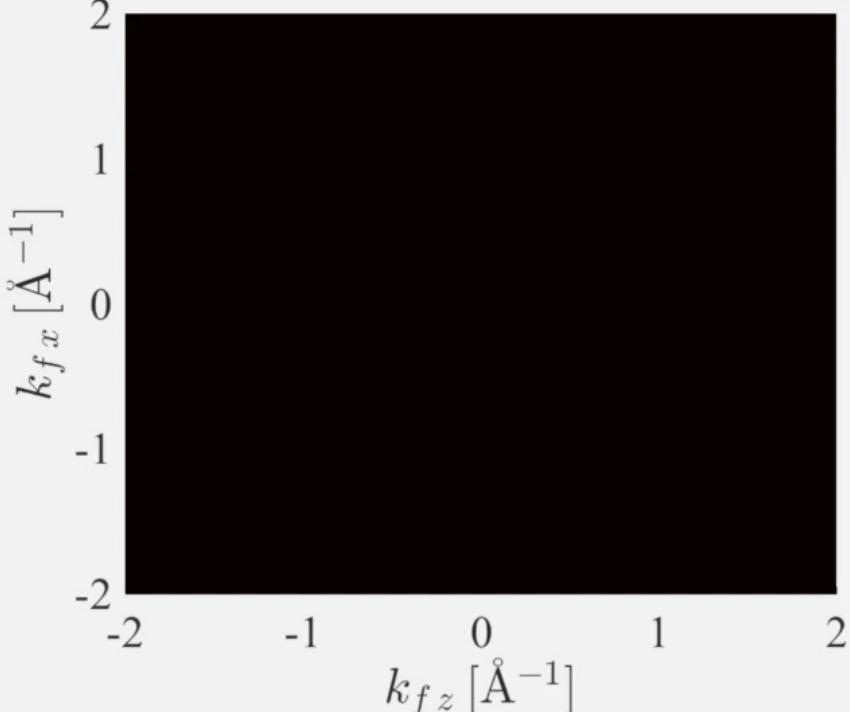
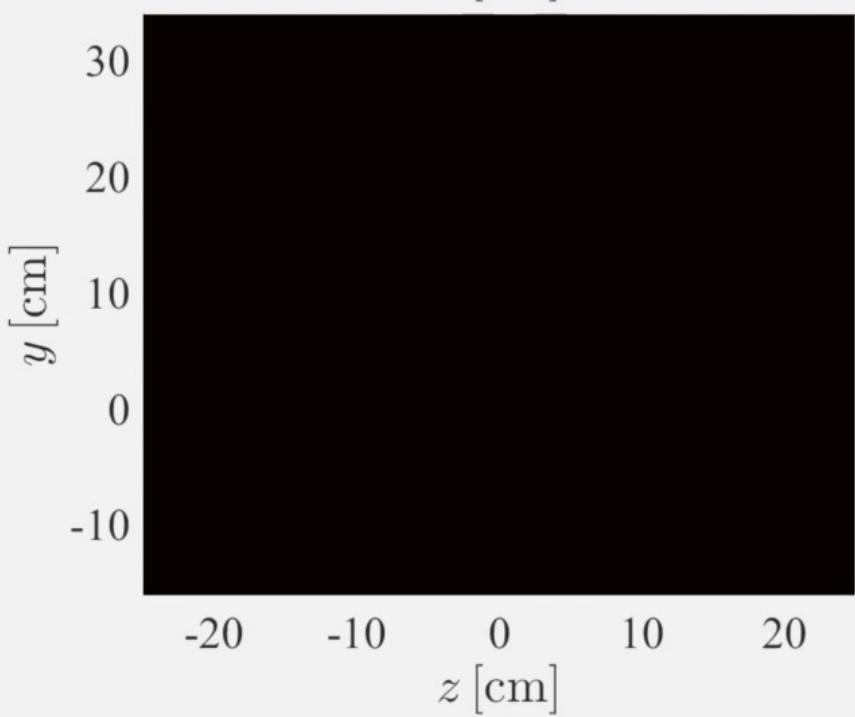
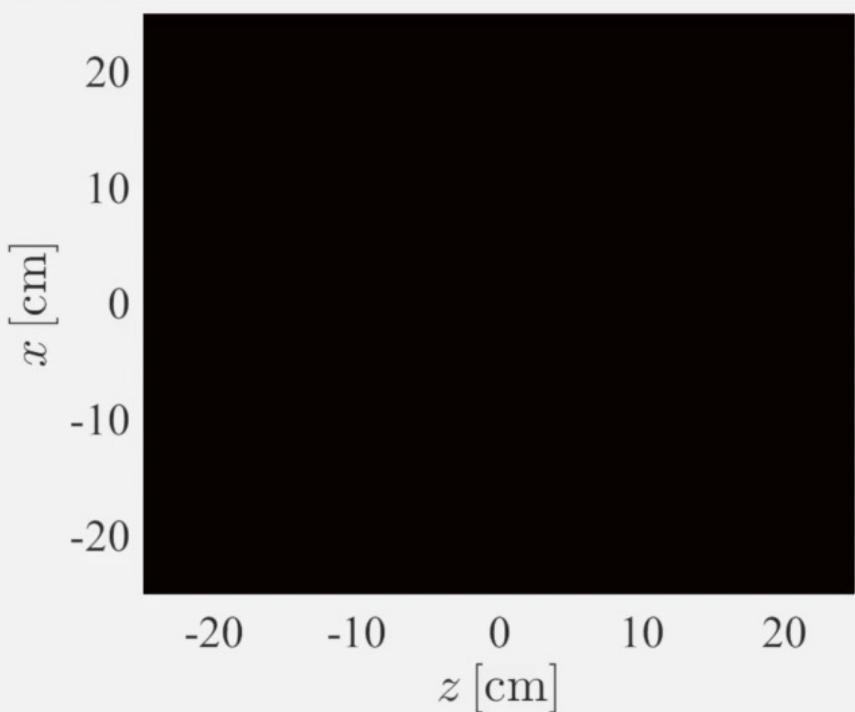
Logger components

Recording information during simulation



- A specific scattering order can be shown
- Improves understanding of what occurred

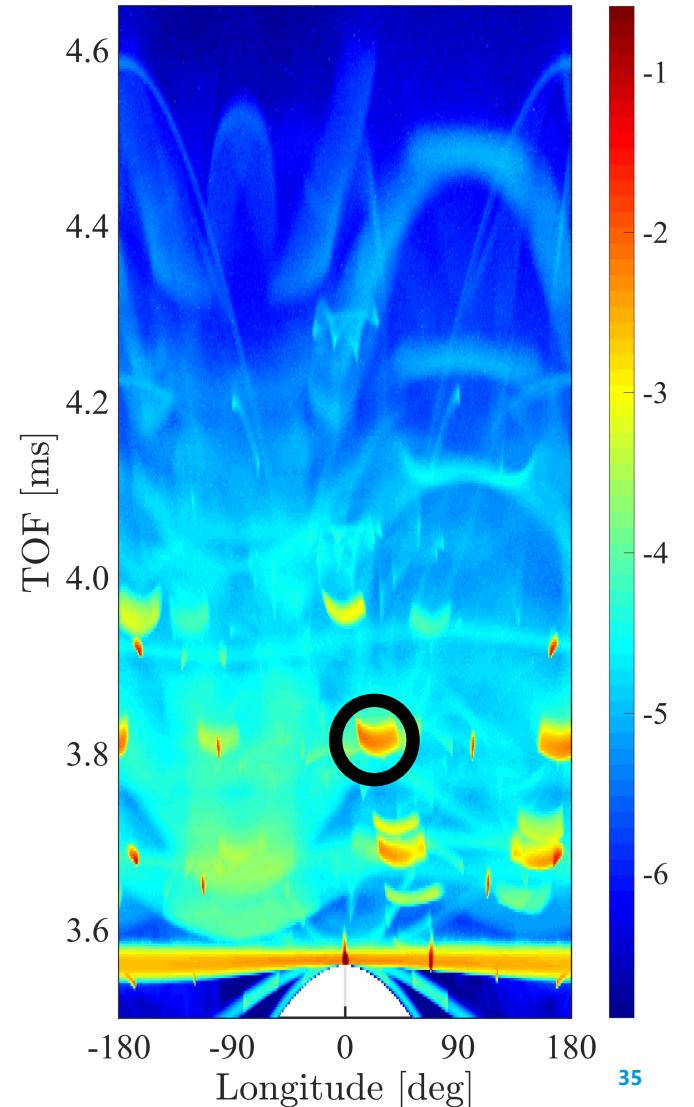
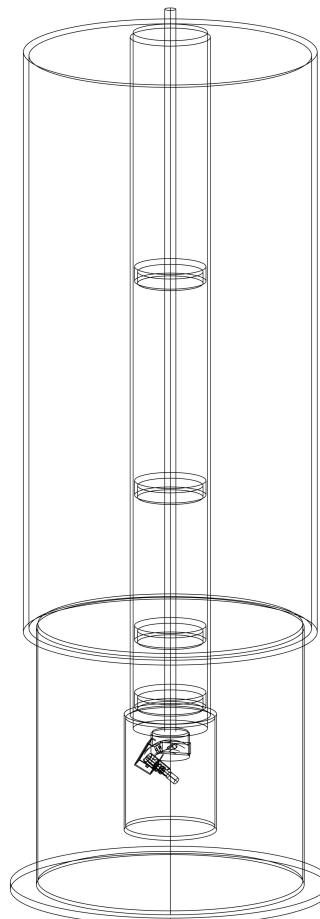




Conditional components

Limiting logger components to record interesting neutrons

- Wish to understand how individual background features arose



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 823852

Conditional components

Limiting logger components to record interesting neutrons

- Conditional components set up some condition upon the final neutron, such as if it hits a certain area
- Conditional components modify existing loggers with *target_loggers*
- Can specify many loggers with comma separated list
- Current conditionals
 - Union_conditional_PSD
 - Union_conditional_standard

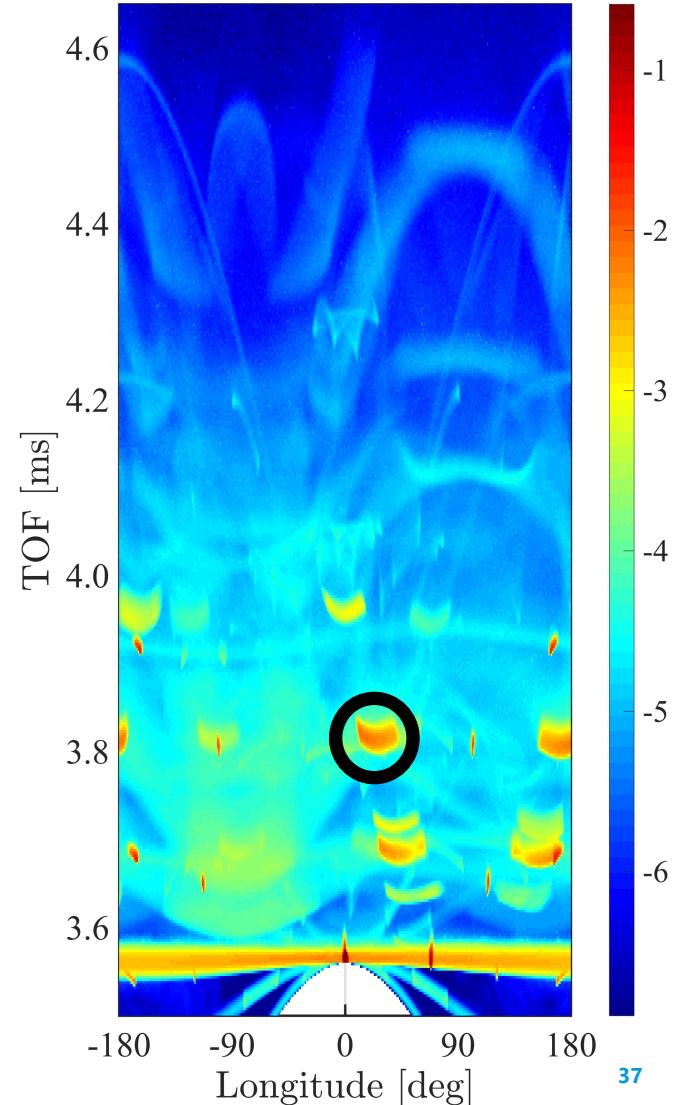
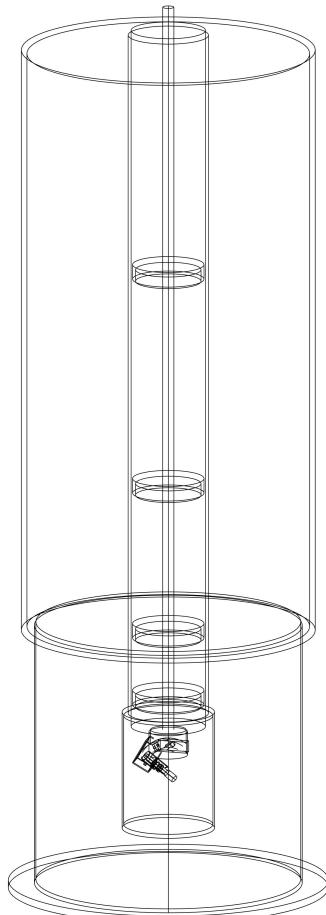
```
COMPONENT scattering_zx = Union_logger_2D_space(  
    D_direction_1 = "z", D1_min = -0.15, D1_max = 0.15, n1 =  
    300,  
    D_direction_2 = "x", D2_min = -0.15, D2_max = 0.15, n2 =  
    300,  
    filename = "scattering_zx.dat")  
AT (0,0,0) RELATIVE sample_position
```

```
COMPONENT conditional_psd = Union_conditional_PSD(  
    xwidth = 0.02, yheight = 0.02,  
    time_min = 2E-6, time_max = 8E-6  
    target_loggers = "scattering_zx")  
AT (0,0,1) RELATIVE sample_position
```

Conditional components

Limiting logger components to record interesting neutrons

- Wish to understand how individual background features arose



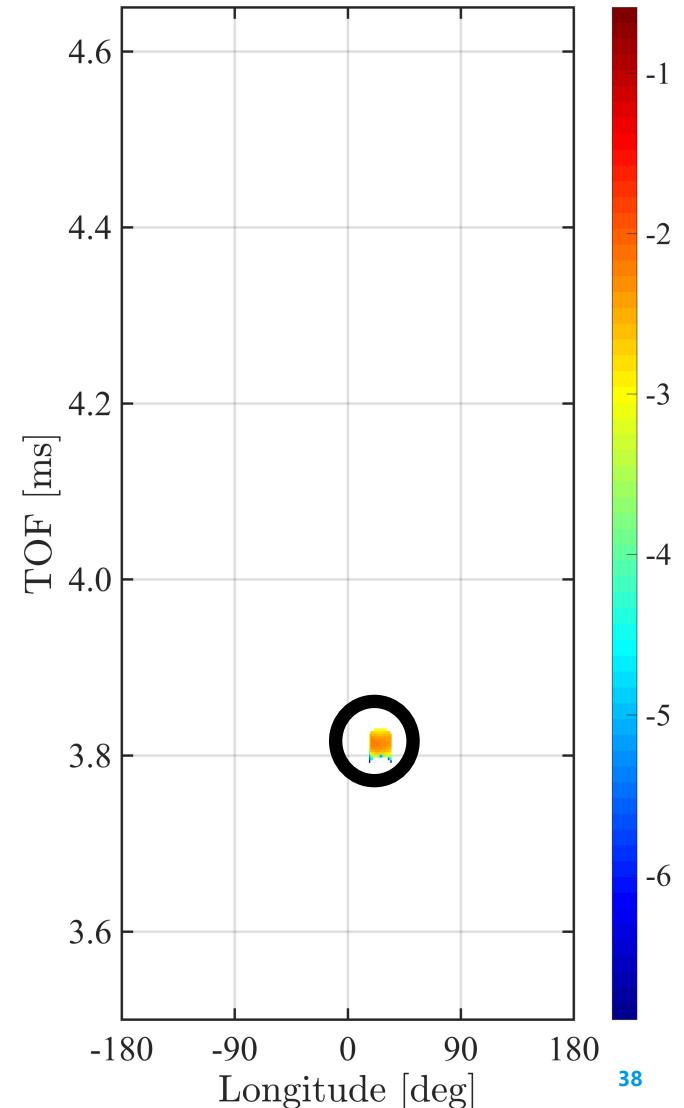
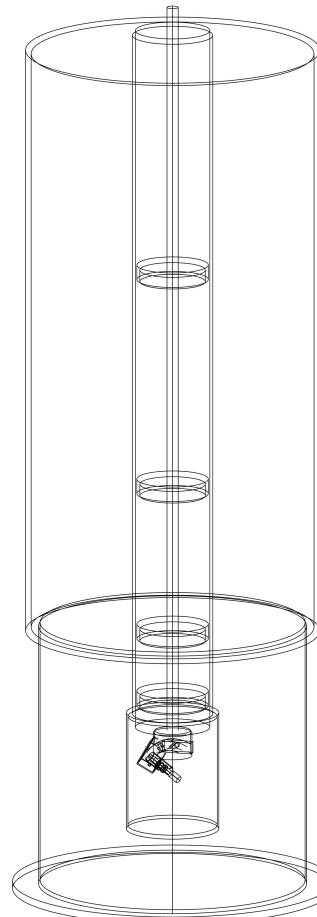
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 823852

Conditional components

Limiting logger components to record interesting neutrons



- Wish to understand how individual background features arose
- Select subset of output to be recorded in loggers using Union conditionals



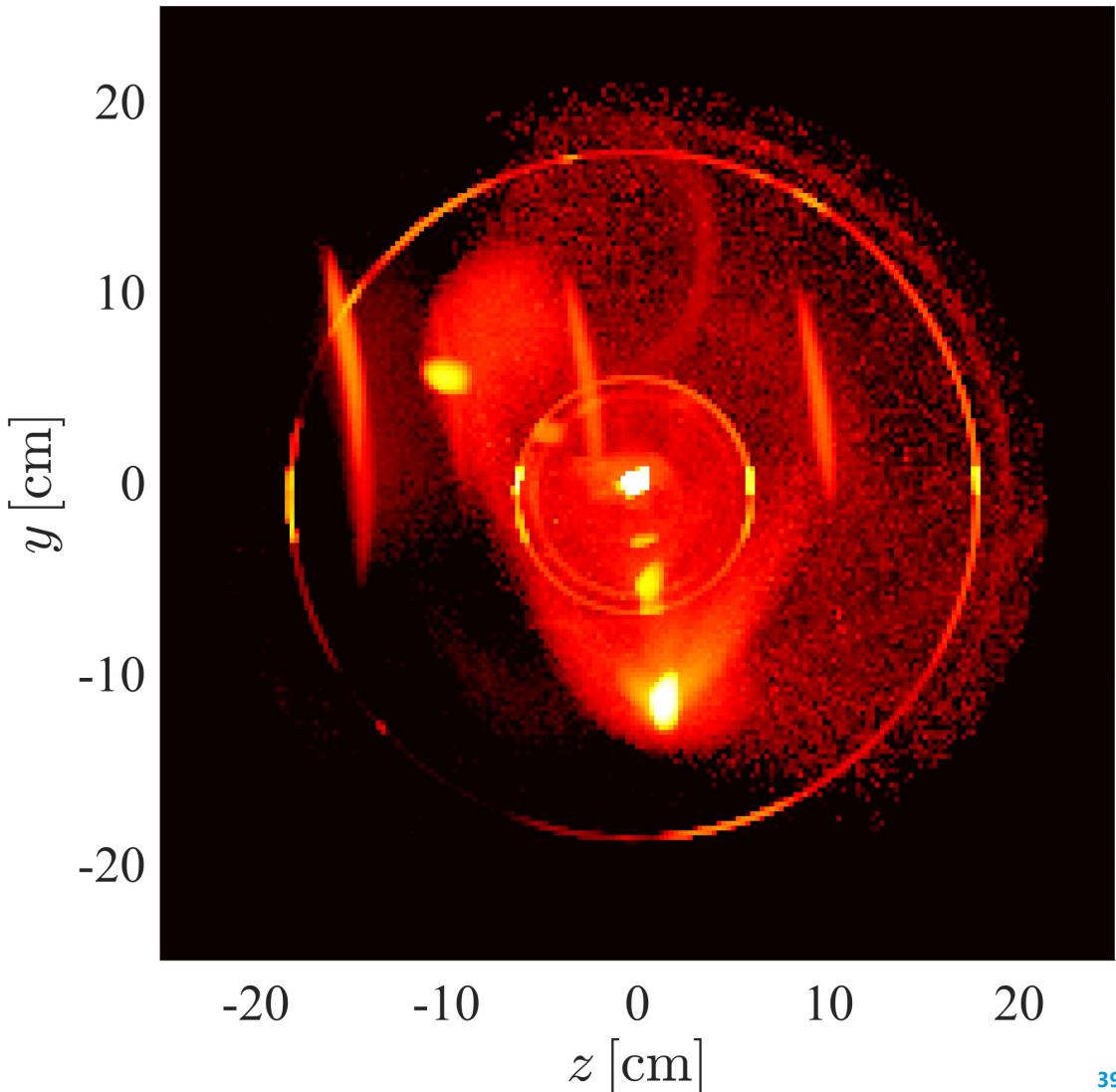
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 823852

Conditional components

Limiting logger components to record interesting neutrons



- Can limit the events that are recorded to the logger using conditionals
- Here only rays that contribute to the chosen background problem is seen
- Spatial distribution of scattered intensity



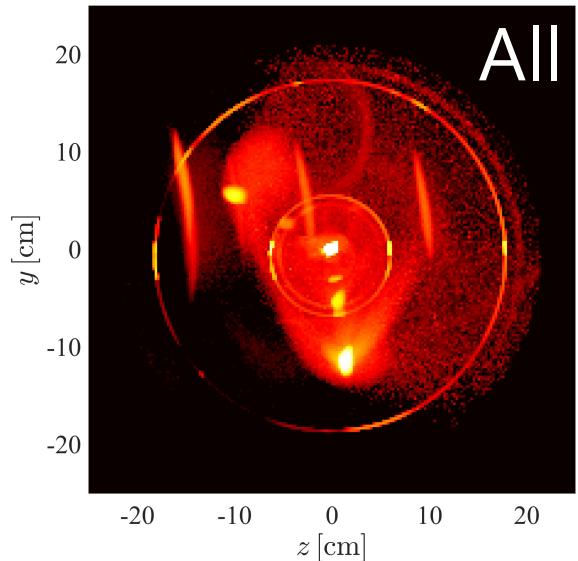
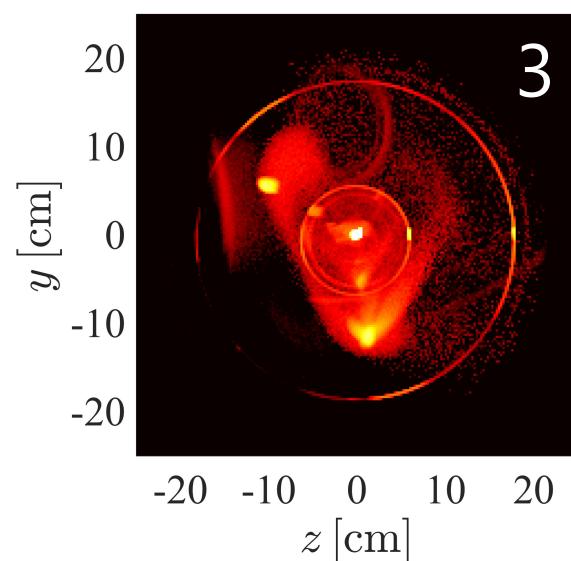
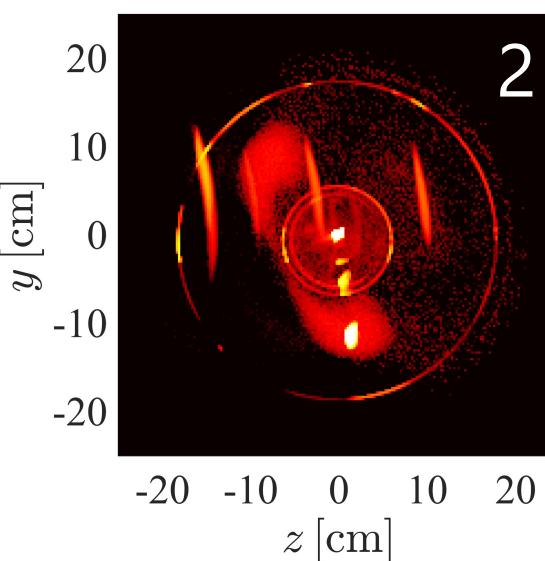
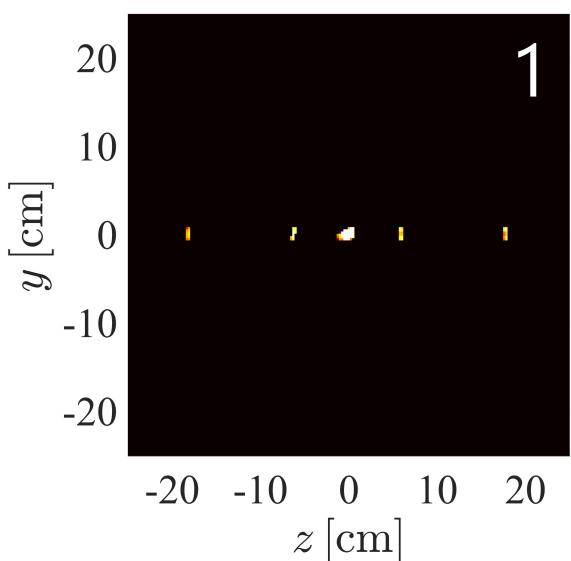
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 823852

Conditional components

Limiting logger components to record interesting neutrons



- The different scattering orders makes it easier to draw conclusions
- The sample is clearly involved

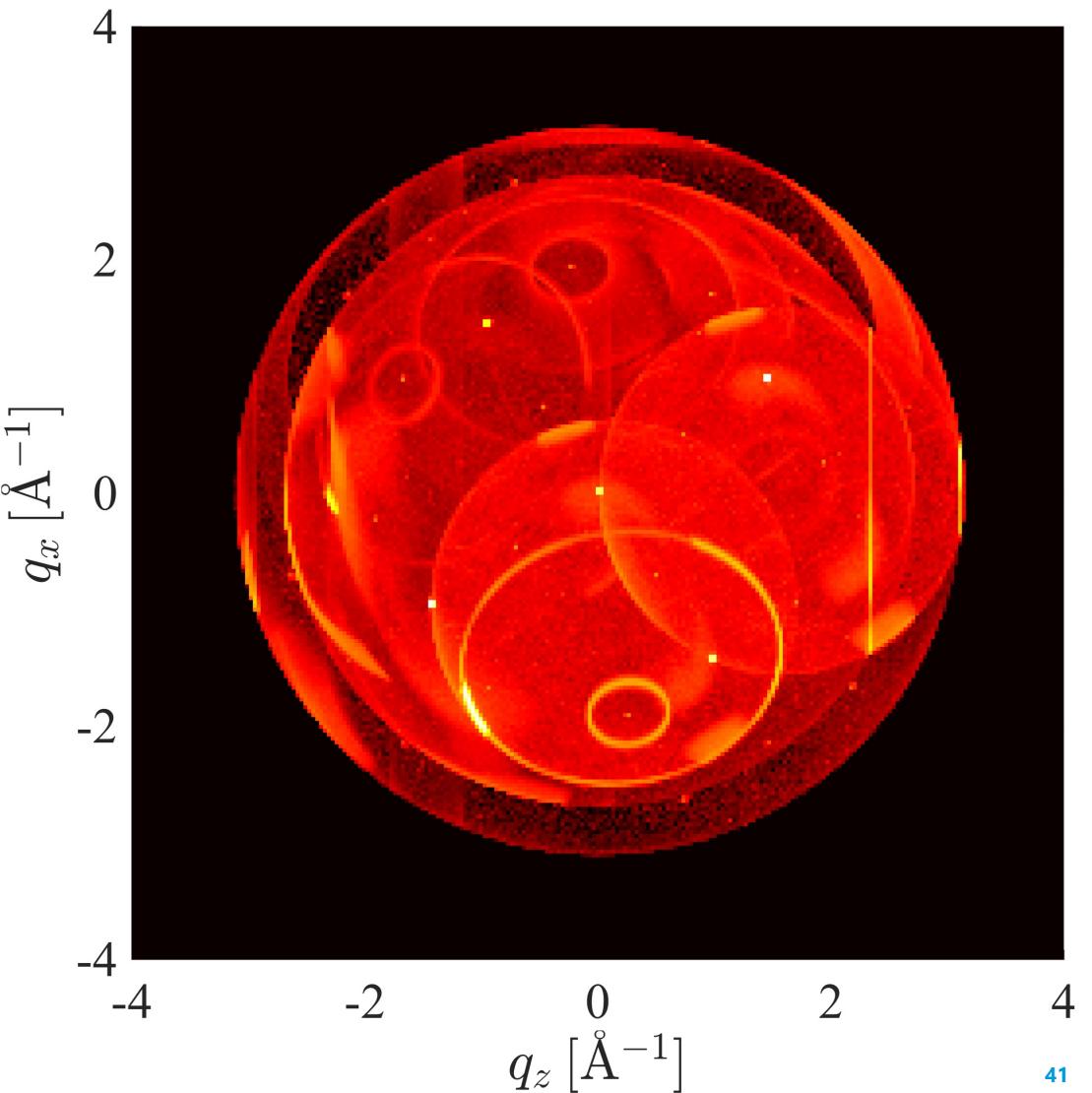


Conditional components

Limiting logger components to record interesting neutrons



- Scattering vectors of rays that contribute to the chosen background problem



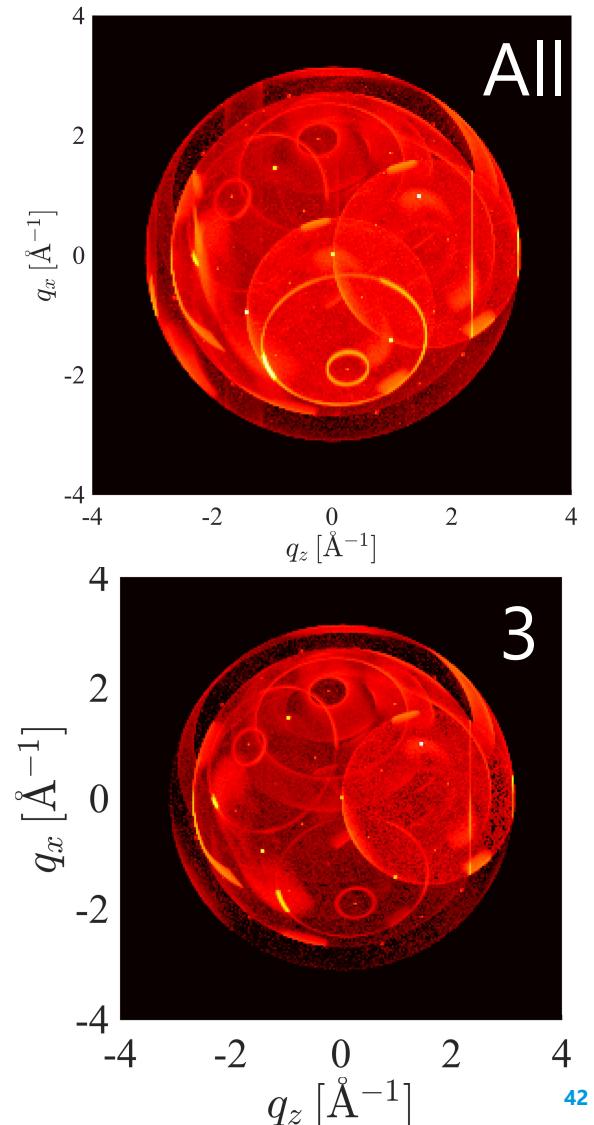
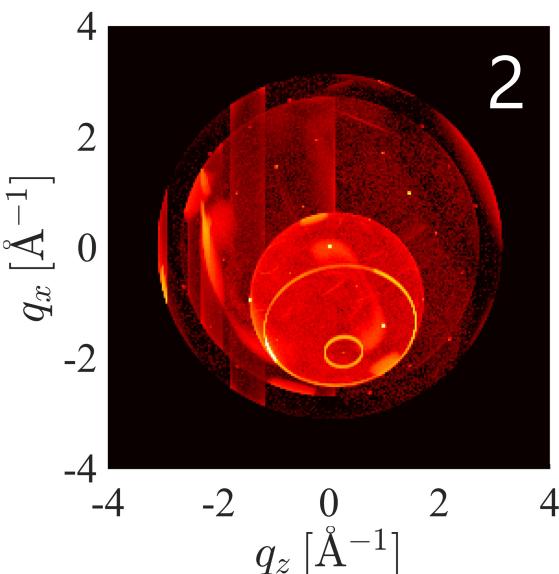
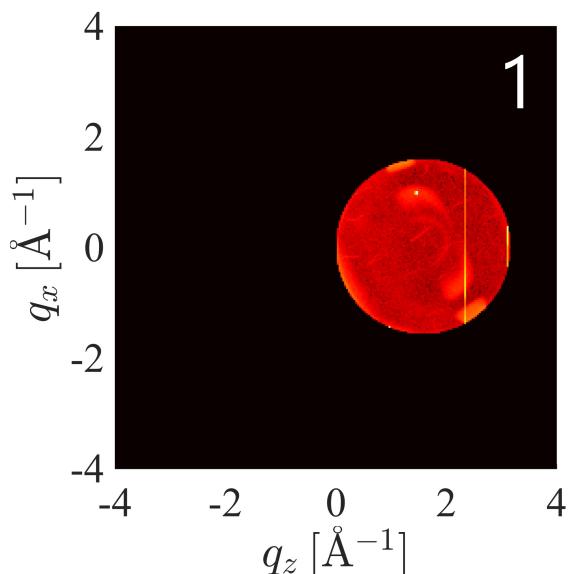
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 823852

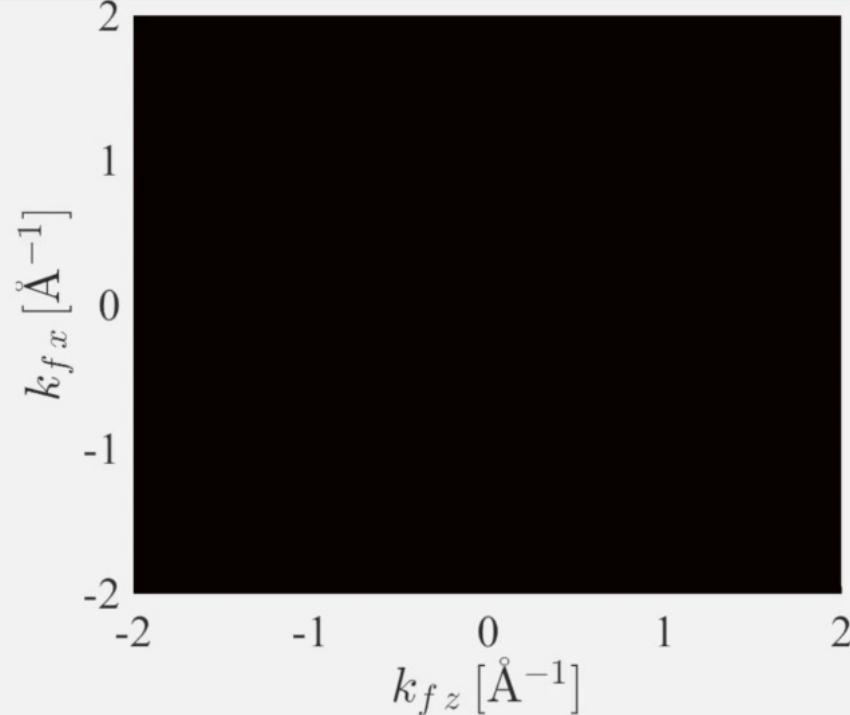
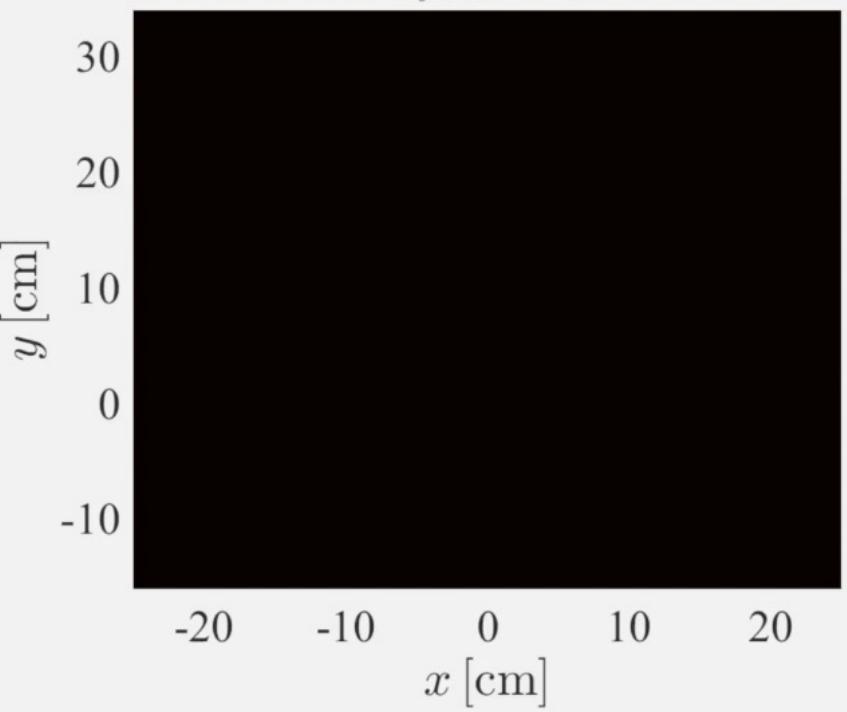
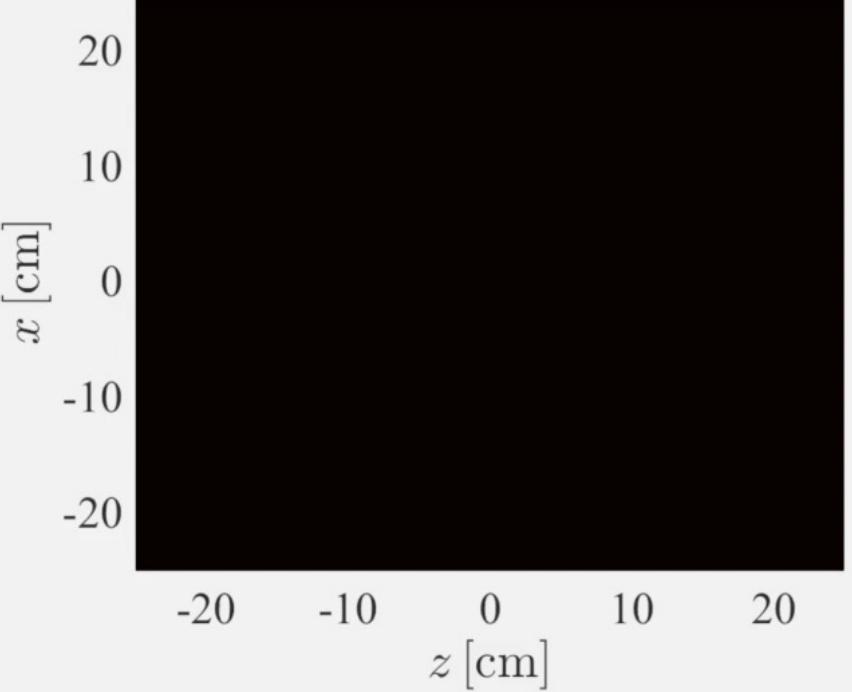
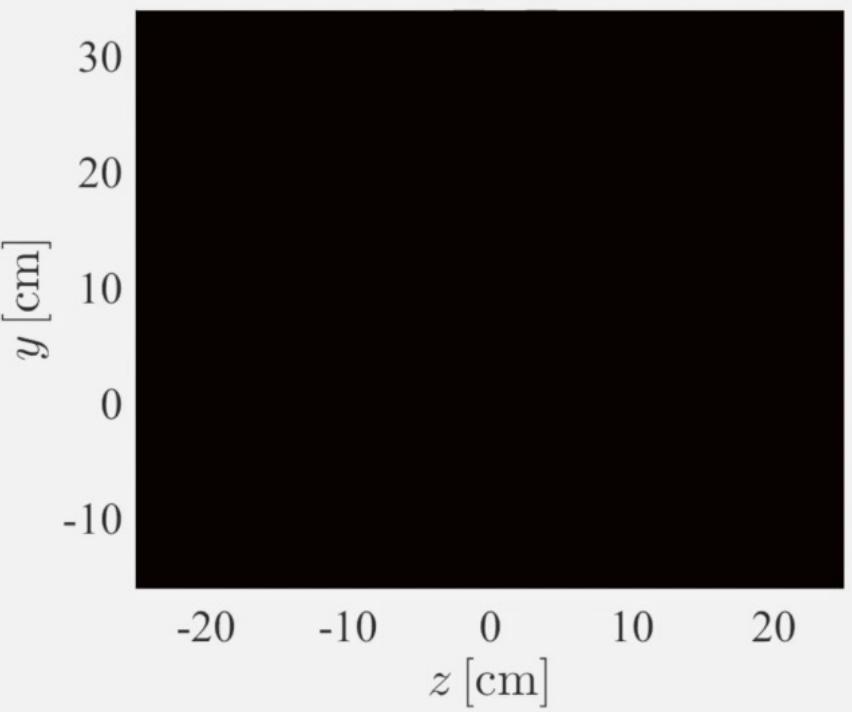
Conditional components

Limiting logger components to record interesting neutrons



- The different scattering orders makes it easier to draw conclusions
- Single crystal Bragg peak to powder Debye Scherrer cone





Summary



- Union components allow simulation of complex physics and geometry
- Full multiple scattering simulation provides surprising level of detail
- The modular nature makes it easier to expand
 - New geometry with a geometry component (Master project contributed cone and CAD)
 - New process component to add new physics
- Full tutorial available in McStasScript format!



Thanks for your attention



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 823852