

GEANT4 Truck scanner

Emil Rofors

emil.rofors@nuclear.lu.se
Division of Nuclear Physics

October 2018

A truck has been modeled and imaged with simulated neutrons, gammas and muons. Cylinders of water, tungsten and gold were placed in the truck as well as three uranium cubes. The three uranium samples are labeled a, b and c where a sits on the floor, b is by the roof and c is by the roof in a 0.5 m thick plastic box.

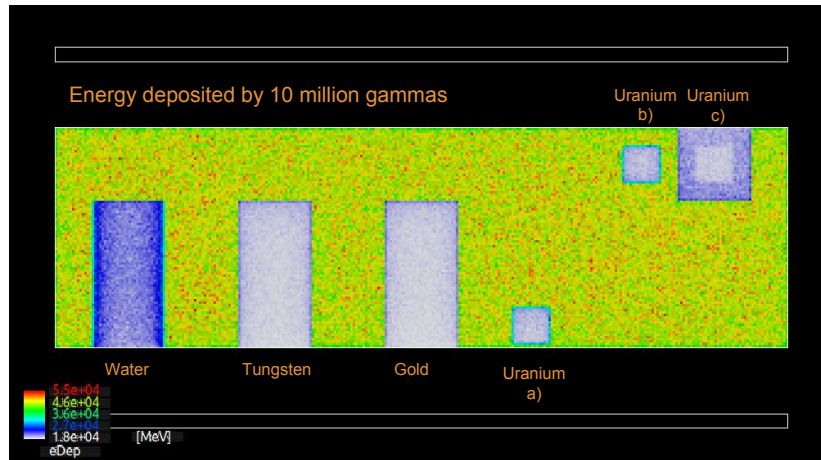


Figure 1: The energy deposited in a plastic scintillator by 10 million incoming 5 MeV gammas.

Gamma and neutron images are shown in Fig 1 and 2. The gammas has an easier time seeing uranium c) that is inside the plastic box than the neutrons. The neutrons show no elemental distinction for these samples.

Fig. 3 show the scattered angle as calculated using two muon detector planes. A limitation of this technique is seen comparing uranium a and b. Uranium a and b are the same size and material but at different heights in the truck. The higher height of uranium b results in higher scattering angles being registered. This height dependency blurs out the ability to differentiate elements such as

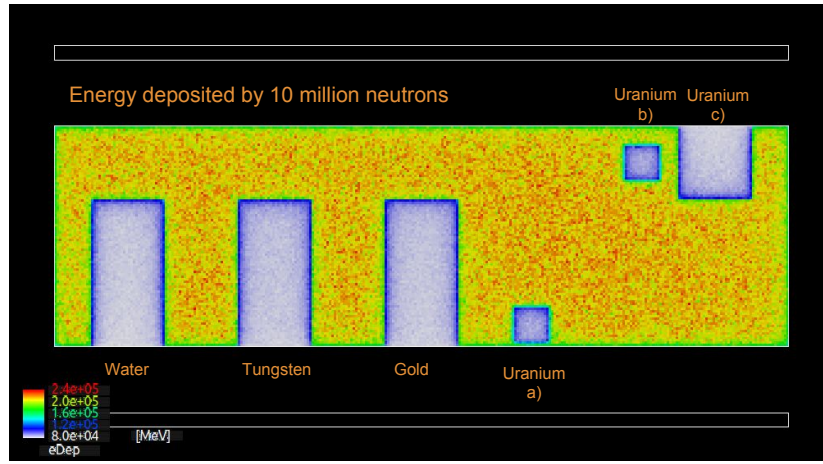


Figure 2: The energy deposited in a plastic scintillator by 10 million incoming 15 MeV neutrons. The uranium c in the plastic box is not found.

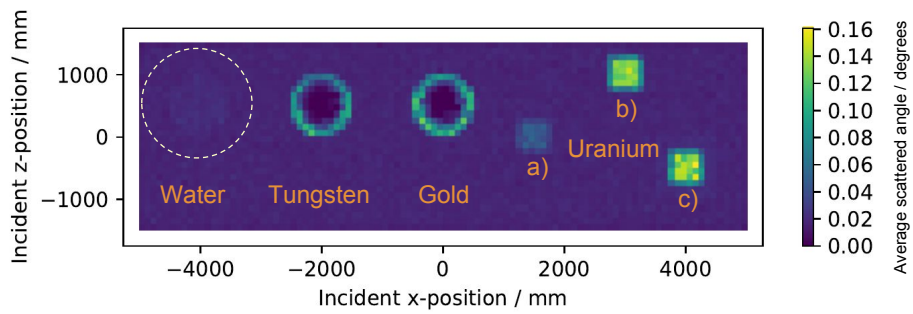


Figure 3: The average angle of scattered 4 GeV muons hitting the truck from above.

the tungsten and gold cylinders. The low-angle scattering area in the center of the cylinders are most likely a sign that the number of counts in these regions are lower, rather than that the scattering angle should actually be lower. By increasing the size of the lower muon detector, to catch more high angle scattered muons, this problem could be mitigated. Light elements like the water and the plastic box around uranium c are barely seen at all.

The code

The code of the main cc-file is built on Hand-in2 and attached in Appendix A. The muon detectors are sensitive detectors that register the world coordinates of the first step an event's muon takes in them. In the Run::RecordEvent func-

tion, it is checked whether both muon detectors were hit. If they were, the coordinates are printed to file using G4cout in combination with the command /control/cout/setCoutFile. This was done as I had a lot of trouble writing to files or filling Ntuples in multithreaded mode and setCoutFile lets each thread write to their own file. The files are then merged with a command like cat *data.csv ; merged.csv .

Appendix A. Code of main program

Listing 1: truck_imager.cc

```

1 #include "DetectorConstruction.hh"
2 #include "ActionInitialization.hh"
3
4 #ifdef G4MULTITHREADED
5 #include "G4MTRunManager.hh"
6 #else
7 #include "G4RunManager.hh"
8 #endif
9
10 #include "G4UImanager.hh"
11 #include "FTFP_BERT_HP.hh"
12 #include "G4StepLimiterPhysics.hh"
13
14 #include "G4VisExecutive.hh"
15 #include "G4UIExecutive.hh"
16 #include "G4ScoringManager.hh"
17 // .....oooOO0OOooo .....oooOO0OOooo .....oooOO0OOooo .....oooOO0OOooo .....
18
19 int main(int argc, char **argv)
20 {
21     //Detect interactive mode (if no argument) and define UI session
22     G4UIExecutive *ui = 0;
23     if (argc == 1)
24     { //No commands line argument
25         //Let G4UIExecutive guess what is the best available UI
26         ui = new G4UIExecutive(argc, argv);
27     }
28
29     // Construct the default run manager
30     // Note that if we have built G4 with support for Multi-threading we set it
31 #ifdef G4MULTITHREADED
32     G4MTRunManager *runManager = new G4MTRunManager;
33     //Set the default number of threads to be the number of available cores of t
34     //If not specified use 2 threads

```

```

35     runManager->SetNumberOfThreads(G4Threading::G4GetNumberOfCores());
36 #else
37     G4RunManager *runManager = new G4RunManager;
38 #endif
39
40     // Activate UI-command base scorer
41     G4ScoringManager *scManager = G4ScoringManager::GetScoringManager();
42     scManager->SetVerboseLevel(1);
43     // Mandatory user initialization classes
44
45     //=====
46     //The Geometry
47     //Sets up truck, gamma/neutron detector, two muon sensitive detectors
48     //and hidden objects in the truck
49     DetectorConstruction *detector = new DetectorConstruction;
50     runManager->SetUserInitialization(detector);
51
52     //=====
53     //The Physics
54     G4VModularPhysicsList *physicsList = new FTFP_BERT_HP;
55     physicsList->RegisterPhysics(new G4StepLimiterPhysics());
56     runManager->SetUserInitialization(physicsList);
57
58     //=====
59     // User action initialization
60     runManager->SetUserInitialization(new ActionInitialization());
61
62     // Visualization manager construction
63     G4VisManager *visManager = new G4VisExecutive;
64     // G4VisExecutive can take a verbosity argument - see /vis/verbose guidance.
65     // G4VisManager* visManager = new G4VisExecutive("Quiet");
66     visManager->Initialize();
67
68     // Get the pointer to the User Interface manager
69     G4UImanager *UImanager = G4UImanager::GetUIpointer();
70
71     if (argc > 1)
72     {
73         // execute an argument macro file if exist
74         G4String command = "/control/execute ";
75         G4String fileName = argv[1];
76         UImanager->ApplyCommand(command + fileName);
77     }
78     else
79     {
80         //We have visualization, initialize defaults: look in init_vis.mac macro

```

```

81     UImanager->ApplyCommand("/control/execute init_vis.mac");
82     if (ui->IsGUI())
83     {
84         UImanager->ApplyCommand("/control/execute gui.mac");
85     }
86     ui->SessionStart();
87     delete ui;
88 }
89 // Job termination
90 // Free the store: user actions, physics_list and detector_description are
91 // owned and deleted by the run manager, so they should not be deleted
92 // in the main() program !
93
94 delete visManager;
95 delete runManager;
96
97 return 0;
98 }
99
100 // .....oooOOOOOooo ..... oooOOOOOooo ..... oooOOOOOooo ..... oooOOOOOooo .....

```
