Documentation for Neutron Project Simulation

Lee Tomlinson

July 30, 2009

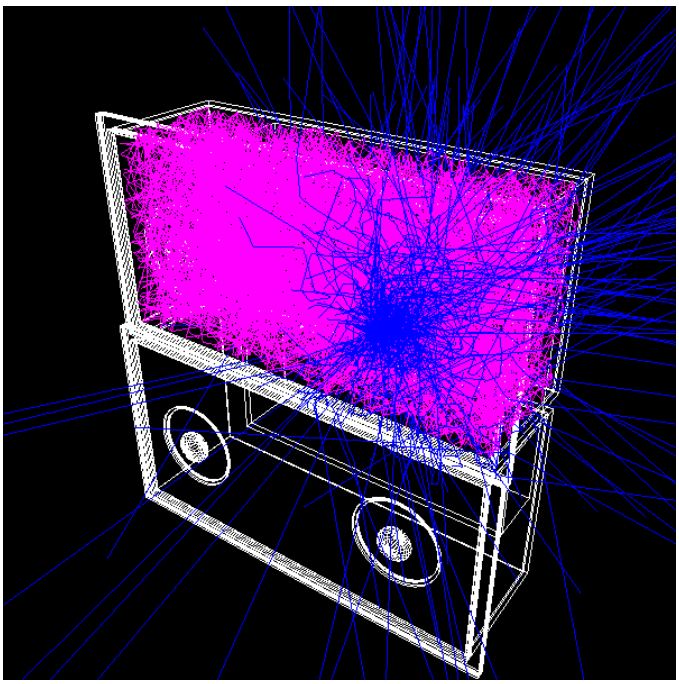

Figure 1: A screenshot of the Neutron Project Simulation in Geant4

1 Description

The purpose of this project is to simulate, using Monte Carlo methods, the passage of neutrons, emitted from a ^{252}Cf source, through two tanks of water. As the neutron is captured by the water, electrons traveling at superluminal velocities may be liberated, emitting Čerenkov radiation. Each tank is lined with diffuse-reflective Halon panels, such that any photon should eventually be captured by one of two photo-multiplier tubes (PMTs).

The simulation project consists of a number of C++ source and header files, located in the `src` and `include` sub-directories respectively. In the project's root directory is a file named `Neutron.cc`, which contains the main loop of the simulation. It is unlikely this file will need editing. To quickly build the simulation, one can simply call the `make` command with the project's root directory as the current directory. This will place the executable, `Neutron`, in the `bin/Linux-g++/` folder.

2 The detector

As mentioned, the experiment consists of two tanks filled with water. These tanks are placed on a stack of lead bricks to *reflect* neutrons, that leave through the bottom of the tank, back into the tank. Each tank is lined completely, with the exception of two approx. 8" diameter holes in the lid, by Halon panels that serve to reflect photons. Finally, two PMTs will protrude through these holes into the water. See Figure **??**, below.

The PMTs are registered as sensitive detectors in the simulation, and will record hits for the captured (Čerenkov) photons. See Figure 2 for an outline of how all the detector components fit together.

All components referenced in Figure 2, except World, may be turned on or off in the simulation. This will require recompilation. To choose which components to include, modify the boolean values located in the `NeutronDetectorConstruction.cc` file located in the `src` directory. The code will look like this:

```
// choose which components to simulate
bool incLead = true;
bool incTank = true;
bool incLid = true;
bool incLookingGlass = true;
bool incWater = true;
bool incRim = true;
bool incTeflon = true;
bool incPMT = true;
```

Below this is more code:

```
// set position of second detector relative to first
G4ThreeVector relativeTranslation(0.*in, 51.*in, -2.*in);

bool twoDetectors = true;
```

Setting this additional boolean value to true will enable both tanks—and corresponding detector parts—in the simulation. The `G4ThreeVector` variable

World The main world volume, made entirely of air, measures 16'0" cubed. The pointer to its logical volume is given by `logWorld` and the pointer to its physical volume is given by `physWorld`.

Lead Both tanks rest on a stack of lead that measures 80" by 84" in the x, y plane. Roughly half the stack is 16" high and the other half is 18" high. See Figure **??** for a schematic. Its logical and physical pointers are `logLead` and `physLead` respectively.

Tank Each tank measures 96" by 48" by $30\frac{3}{4}$" and has a lip that protrudes 3". See Figure **??** for a schematic. Its logical and physical pointers are `logTank` and `physTank` respectively. The physical volume is parameterized to obtain both tanks.

Lid Each lid measures 102" by 54" by $\frac{1}{2}''$ and contains two $20\frac{1}{2}$" diameter holes. Its logical and physical pointers are `logLid phyLid` respectively. The physical volume is parameterized to obtain both lids.

Looking Glass These are a temporary fixture, replaced by the PMTs. `physLookingGlassA` and `-B` are distinct physical volumes derived from the same logical volume `logLookingGlass`. The two physical volumes are parameterized to obtain all four glasses.

Water The water volume occupies the total internal volume of the tanks—

approx. $94\frac{1}{2}$" by $48\frac{1}{2}$" by $29\frac{3}{4}$".

Its logical and physical pointers are `logWater` and `physWater` respectively. The physical volume is paramaterized to obtain both volumes of water.

Rim These are the aluminium rims. The pointer to the logical volume is `logRim`, and the pointers to the derived physical volumes are `physUpperRim` and `lowerRim`. Both physical volumes are parameterized to obtain all four rims.

Teflon This represents the Halon panels.

Figure 2: Description of the detector geometry

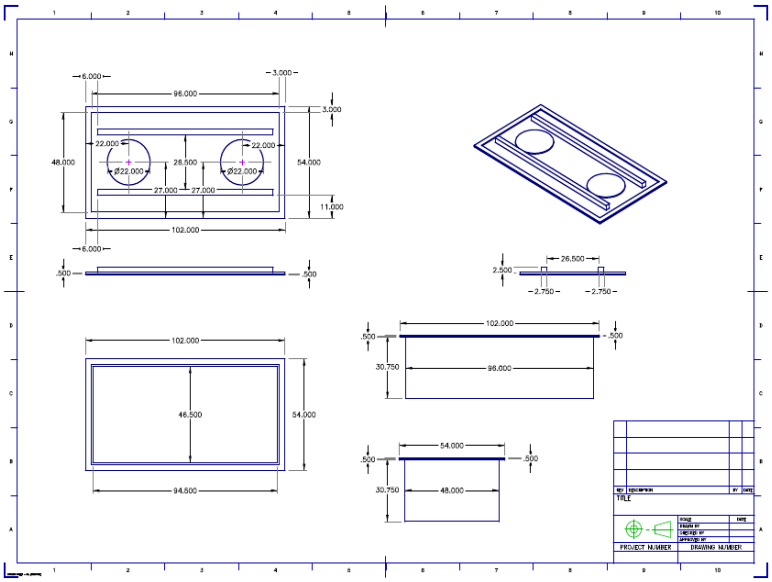

Figure 3: Schematic of the detector construction

should not need to be changed. This is the relative position of the two tanks as the stand on the lead stack.