G4BEAMLINE SIMULATIONS FOR DETECTOR DEVELOPMENT*

Thomas J. Roberts, Kevin Beard (Muons, Inc., Batavia, Illinois, USA 60510)

Abstract

The G4beamline program [1] is a useful and steadily improving tool to quickly and easily model beamlines and experimental equipment without user programming. As it is based on the Geant4 toolkit [2], G4beamline includes most of what is known about the interactions of particles and matter, including time-varying electromagnetic fields. We are continuing the development of G4beamline to extend its domain to include detector systems. G4beamline is open source and is freely available at http://g4beamline.muonsinc.com.

INTRODUCTION

G4beamline [1] is a general and flexible tool for the simulation and evaluation of beamlines and related systems, using single particle tracking. The Geant4 toolkit [2] has proven to be an excellent choice for the basis of the program, as it is comprehensive, accurate, and actively supported by a strong and vibrant collaboration. Unlike many physics programs, particular attention has been paid to G4beamline's user interface, with the result that it has a considerably shorter learning curve than most comparable programs. Most accelerator physicists can read and understand its description file without reference to the documentation, and can learn how to develop their own simulations with minimal effort. Extensive online help is available within the program to assist users in developing their simulations. Figure 1 shows the G4beamline graphical user interface (GUI) screen, including an index and the beginning of its Help text.

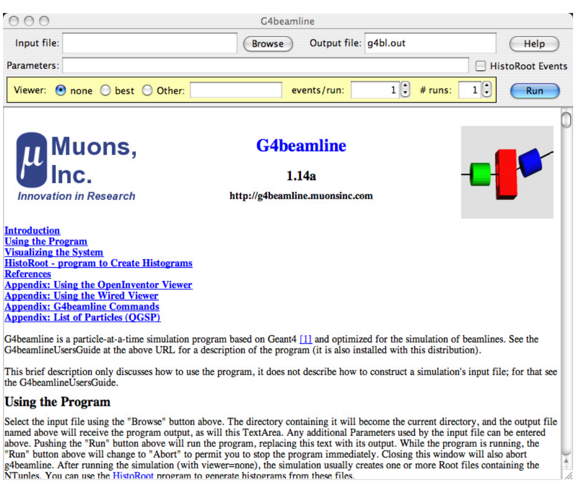

Figure 1 : The G4beamline GUI screen.

To facilitate the generation of histograms and plots, the G4beamline distribution includes the historoot program that provides a user-friendly graphical interface to Root [3]. While general Root programming can be used to create plots, most users find the interface shown in Figure 2 to be more usable and efficient.

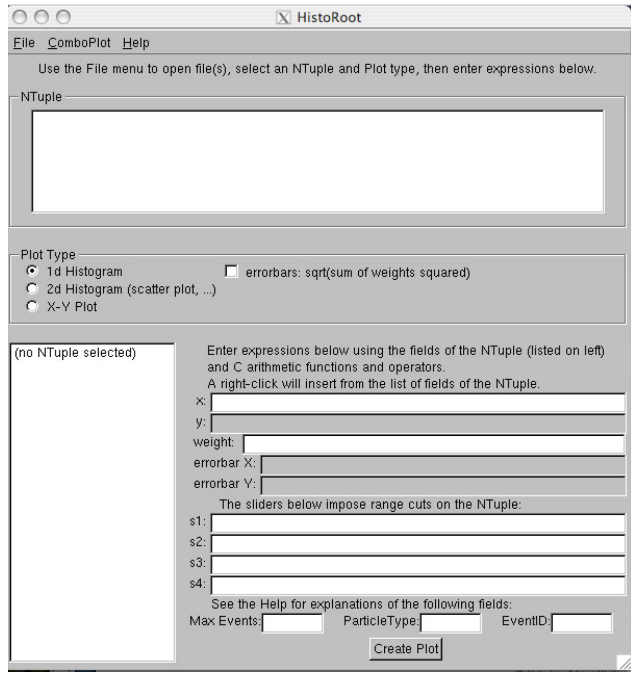

Figure 2 : The historoot GUI screen.

At present, G4beamline has extensive features for specifying and simulating beamlines, but cannot simulate detectors or include their data into its output file. Our plans include extending the program to interface to the Geant4 facilities for detectors: active volumes, hits, and digitization. These capabilities will enable the user to generate a Root output file that is similar in many ways to the data files from actual experiments. In addition, we will interface to the existing Geant4 processes for optical photons and very low energy physics, extending them where necessary to be able to simulate the inner workings of a large class of detectors.

DESCRIPTION

As the G4beamline code was designed from its inception to be easily extensible, we can add these new features to it without disturbing its current capabilities, or forcing existing users to change their input decks. This is easily accomplished by simply implementing the new features in new commands; adding new commands merely requires deriving a new class for each of them from the appropriate base class, and putting their source code into the proper directory. In particular, no changes to existing code are required to add new commands (the list of commands is dynamically built from the files located in the source directory). There will be a new infrastructure

* Supported in part by DOE STTR grant DE-FG02-06ER86281.

required to implement the expanded Root output file capabilities.

Another common feature of detectors is that they can be segmented, with a separate readout per segment. To make it easy to simulate a large number of identical detectors (a multi-wire proportional chamber, a scintillating fiber hodoscope, etc.), an interface to the Geant4 "parameterized placement" will be implemented. The user will define a single wire or fiber, and then use a single command to place many copies into a single plane or unit. The output file will include identification of which detector was hit, as well as position information, energy deposited, timing information, etc.

Initially we will focus on three detector types during the implementation of the new capabilities:

1. A bulk detector that responds to the energy deposited in its volume (e.g. a Sodium Iodide crystal).
2. A plane of identical detectors that respond with individual hits (e.g. a proportional wire chamber).
3. A new type of fast timing detector with better than 5 picoseconds resolution (see below).

In the style of G4beamline, these will all be extensively configurable by the user, but with sensible defaults to simplify learning how to use them.

BULK DETECTOR

This will be a generic detector that responds with a pulse-height output related to the energy deposited in the detector volume. Initially it will have a simple, user-configurable mapping from energy deposition to pulse height to permit the code to be used for various physical situations. Relaxation time of the detector will also be user configurable.

PLANE OF IDENTICAL DETECTORS

This will be a generic detector consisting of a plane of identical segments that responds with a list of hit segments. It will have a user-configurable probability of reporting a hit based on the number of charged particles that traverse each individual segment's volume. That will permit the code to be used for different detector technologies (e.g. multi-wire proportional chambers, scintillating fiber hodoscopes, etc.).

FAST TIMING DETECTOR

Initial prototype work by an informal collaboration [4] has shown that it is feasible to develop detectors for charged particles that have considerably better timing resolution than current detectors: resolutions less than 5 picoseconds appear likely. This technology is based on a micro-channel plate coupled to a collection anode; when combined with integrated readout electronics it offers the possibility of segmentation in two dimensions with pixels on the order of half a centimeter square (and perhaps smaller). To optimize the design, detailed simulations of the detector are required; this includes particle simulations as well as electronics simulations that will be performed separately. One of the key enablers of this technology is the ability to design a custom ASIC with the bandwidth to implement multiple TDCs with picosecond resolution. The basic concept is shown in figure 3.

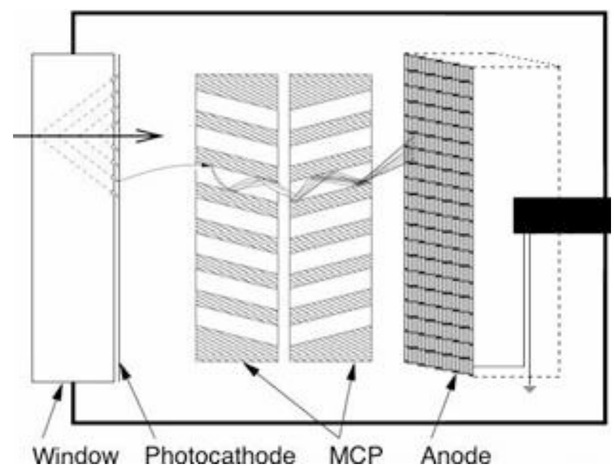

Figure 3 : Schematic of the fast timing detector. A relativistic charged particle produces Cherenkov light in the window. This radiation is converted into electrons by a photocathode. The electrons produce a shower in the micro-channel plates (MCP), and the electrons are deposited on the segmented anode to be detected. Not shown: the anode has equal-time connections from each segment to the output. The drawing is not to scale.

SUMMARY

As the Geant4 toolkit already has most of the capabilities and physics processes required, the actual software development required for these enhancements will be rather modest. The new capabilities will permit G4beamline to simulate a modest particle experiment (simple trigger, tracking, particle ID, basic calorimetery), as well as the beamlines and systems that it already handles. In addition, it will become possible to simulate the inner workings of a large class of interesting detectors.

REFERENCES

[1] G4beamline – http://g4beamline.muonsinc.com
[2] Geant4 Toolkit – http://geant4.cern.ch
[3] Root – http://root.cern.ch
[4] Fast Timing Workshop – http://www.hep.anl.gov/ertley/tof/talks.html