

PARTICLE TRACKING IN MATTER DOMINATED BEAM LINES^{*}

T. Roberts[#], K. Beard, Muons, Inc., Batavia IL, USA

S. Ahmed, D. Huang, D. M. Kaplan, L. Spentzouris, IIT, Chicago, IL, USA

Abstract

The G4beamline program [1] is a useful and steadily improving tool to quickly and easily model beam lines and experimental equipment without user programming. Unlike most accelerator physics codes, it easily handles a wide range of materials and fields, being particularly well suited for the study of muon and neutrino facilities. As it is based on the Geant4 toolkit [2], G4beamline includes most of what is known about the interactions of particles with matter. We are continuing the development of G4beamline to facilitate its use by a larger set of beam line and accelerator developers. It is open source and freely available at <http://g4beamline.muonsinc.com>.

INTRODUCTION

As accelerator facilities become more complex and more expensive, accurate and comprehensive simulations of their performance are required long before construction begins. There are many choices and optimizations to be made, as well as new concepts to be explored, so flexible and user-friendly simulation programs become essential to streamline the design process. For future facilities such as muon colliders and neutrino factories, the muon cooling sections demand simulations that accurately compute the interactions of particles in matter, along with associated magnetic and RF fields. The Geant4 toolkit [2] is an excellent choice as the basis of such a program, as it is comprehensive, accurate, and actively supported by a vibrant collaboration. G4beamline [1] was conceived as a user-friendly interface between accelerator physicists and the C++ code of Geant4 to facilitate the rapid evaluation of new concepts and design changes by physicists without the complexities of C++ programming. An important aspect of G4beamline is that its description of the simulated system is far more comprehensible by unspecialized physicists than the corresponding Geant4 C++ code would be, and no more complicated than is the system itself.

DESCRIPTION

An obvious aspect of G4beamline is that its user interface has been designed with physicist-users in mind. The system to be simulated is described in a single ASCII file using an object-oriented language specifically designed for this application. Most accelerator physicists can read and understand such a descriptive file without reference to the G4beamline documentation, and can learn how to develop their own simulations with minimal effort.

^{*} Work supported in part by USDOE STTR Grant DE-FG02-08ER86281.

[#] tjrob@muonsinc.com

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 a hyperlinked index and the beginning of its Help text.

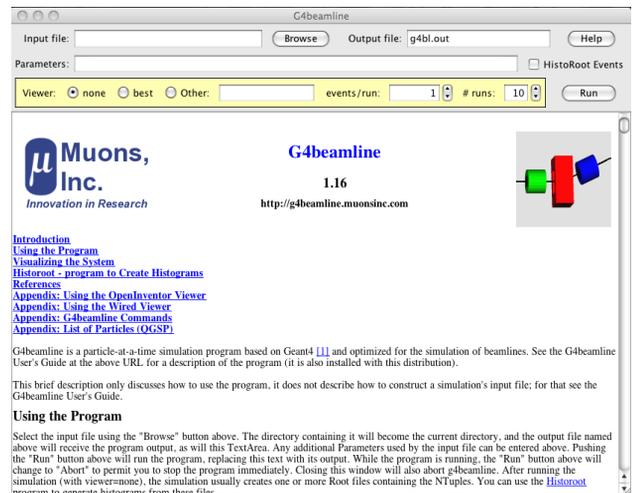


Figure 1. The G4beamline GUI Screen.

To facilitate the generation of histograms and plots, the G4beamline distribution includes the HistoRoot program [3], which provides a user-friendly graphical interface to ROOT [4]. 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.

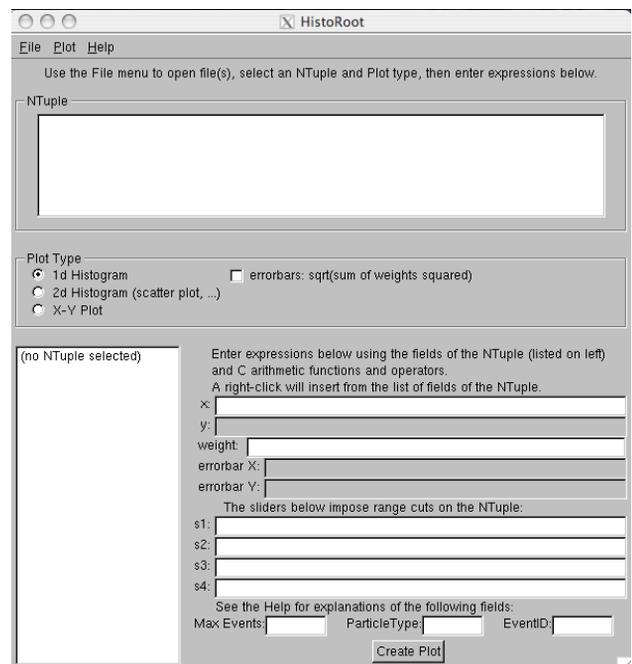


Figure 2. The HistoRoot GUI Screen.

The major aspects of G4beamline are:

- Accurate and realistic simulations using the Geant4 toolkit
- A physicist-readable ASCII file to specify the simulation, with auxiliary files for field maps, etc.
- A rich repertoire of beamline elements that can be combined to define new and customized elements
- A general set of initial beam specifications (including a cosmic ray muon “beam” and external files)
- Input and output of beam tracks using several formats including ASCII and ROOT [4] files – this permits easy interfacing to other programs (e.g., for partitioning of complex systems and verification of results)
- Automatic tuning of many parameters (RF cavity timing and gradients, bending magnet fields, etc.)
- Included visualization of the simulated device or system using many viewers (OpenGL, VRML, Open Inventor, etc.) – provides a powerful visual check that the system implemented is the system desired, and lets you see tracks and their interactions with materials
- Support for parallel jobs on multiple CPUs
- The HistoRoot program, which makes it easy for non-experts to generate ROOT [4] histograms and plots

The basic structure of a simulation is to first define the beamline elements to be used (magnets, RF cavities, etc.), including their geometry, materials, and local fields. Then these elements are placed into the “world”, usually along the nominal beam centerline; each placement can have a position, rotation, and its own field value. Parameters for the element can be defined in the input file or on the command line, so scripting is straightforward. Individual particles can be traced, beam profiles can be generated and displayed, and “virtual detectors” can be used to sample the beam at any point.

The tracking of particles through the simulated system is as accurate and realistic as the Geant4 toolkit implementation. The input file selects any one of the Geant4 physics lists, and can set values for the various Geant4 tracking accuracy parameters. This permits users to make trade-offs between CPU time and simulation accuracy. Similarly, G4beamline permits the specification of magnetic map parameters, permitting a trade-off between memory usage (and the CPU time needed to generate the map) and simulation accuracy.

While G4beamline can make it rather simple to specify a simulation, it cannot substitute for knowledge and experience about the problem domain or about particle tracking simulations in general. Like all computer programs, G4beamline is prone to “garbage in, garbage out,” especially in the hands of unskilled users. It is strongly suggested that visualization be used to verify the geometry of the simulation and the proper tracking of at least a handful of particles. Whenever possible, one should arrange to track through a simple geometry that can be compared to independent results, to make sure that

what one thinks is happening actually does occur in the simulation.

EXAMPLE

An example of using G4beamline is shown in Figure 3. This is the Mu2e experiment at Fermilab [5], searching for the lepton-violating conversion of μ^- to e^- near an atomic nucleus. Many aspects of the design of this experiment have been explored and evaluated using G4beamline and other programs.

SOFTWARE DEVELOPMENT

G4beamline is being developed using modern software development techniques. In particular, our methodology requires that feature documentation be written before the code, and that the documentation is contained within the code, so there is always comprehensive and up-to-date documentation available to users. There are two levels of documentation:

- User documentation describing how to use the code
- Internal documentation describing what the code does and how it works.

The first is intended for users and is contained in Help text within the code to implement always-available online help. The latter is intended only for developers, and is contained in structured comments that the doxygen [6] system converts into hyper-linked HTML. Keeping all documentation within the code makes it easy for developers to keep everything in sync; automated tools format it for presentation to users (e.g., the User’s Guide).

RECENT FEATURES

Several recently added features make G4beamline more powerful and more usable by physicists:

1. The User’s guide has been considerably enhanced and numerous tips and techniques have been discussed for the best performance of simulations and generation of results.
2. An event-list interface between HistoRoot and G4beamline. The user can select an outlier event in a histogram or plot and easily re-run G4beamline with visualization to see what that particular event did.
3. More Geant4 objects have been implemented, expanding the repertoire of G4beamline.
4. A direct interface to the NIST material database [7] gives users the ability to use many common materials without manually looking up their properties.
5. Rudimentary track fitting is implemented, sufficient to evaluate momentum and mass resolutions; combined with the inherent flexibility of G4beamline this becomes a powerful tool for the early design of experiments.
6. Visualization has been enhanced by adding the feature to generate movies of events showing the time evolution of Ntuple values.

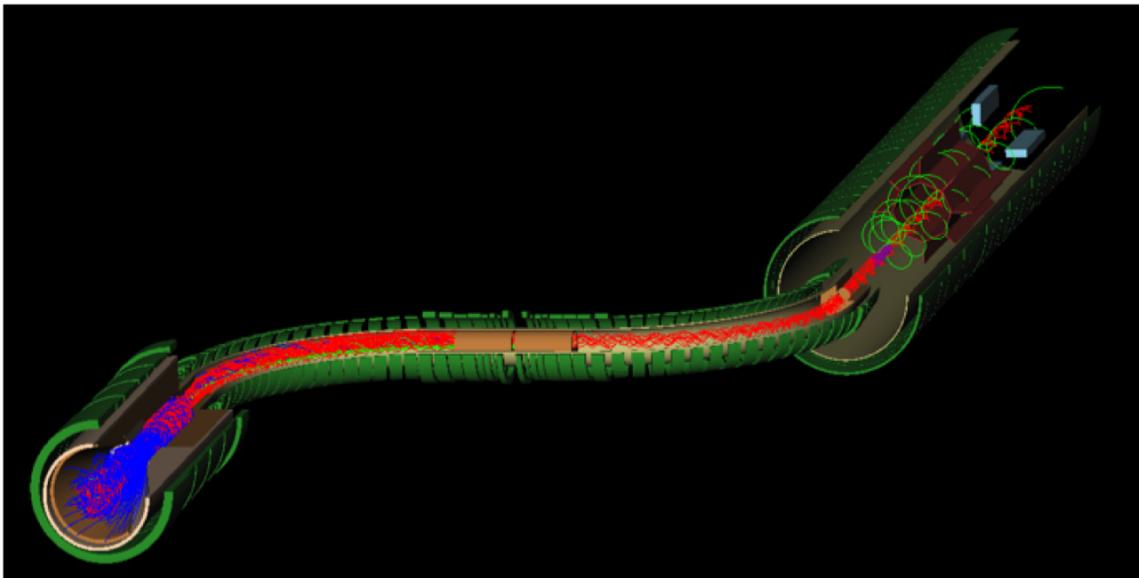


Figure 3: G4beamline visualization of the Mu2e experiment. The production target is at the left, followed by the S-shaped transport solenoid, to the stopping target, with the electron spectrometer and calorimeter behind; π^- are blue, μ^- are red, and e^- are green.

FUTURE PHYSICS INTERESTS

We are currently developing or planning the development of the following new features:

1. Space Charge: Current design projections for muon colliders indicate that space charge may be a problem in the final stage of cooling. The implementation of space charge physics in vacuum is almost completed and validation is progressing. The physics model is similar to the existing programs such as ORBIT [8], TREDI [9] and PARMELA [10]. This includes radical revisions to the Geant4 G4Runmanager and G4EventManager to track particles in parallel.
2. Collective effects in matter: at the end of a muon cooling channel, very high beam densities are expected, and the collective effects between the beam and the material in the absorbers are expected to be important. This requires an interdisciplinary approach that includes atomic, accelerator, and plasma physics.
3. Polarization physics: Many new muon facilities including muon colliders can benefit from using polarized muons. This requires modeling of polarization especially during the interaction with matter in a cooling channel.
4. Very low energy physics: We are interested in simulating the inner workings of advanced detector systems, atomic traps, and some of the details of surface effects. These require accurate modeling at energies of a few tens or hundreds of eV.
5. Automated parallelization: Computer farms and multi-CPU systems are now common, and we will enhance G4beamline to automatically take advantage of parallelism. At present users can do this manually.

SUMMARY

G4beamline is a highly flexible and user-friendly program for simulating beamlines both with and without matter. The learning curve is rather short for a program of this scope, and with sufficient attention to detail, the simulation can be quite realistic. We are continually advancing the applicability of the code. In addition to the primary use of investigating many aspects of muon cooling for a muon collider or neutrino factory, applications have included: target hall and test beam design evaluation, feasibility studies for new facilities and experiments, cosmic-ray muon tomography, and detector design. Our commitment to supporting it, our current funding for new features, and our growing user base ensure that G4beamline will remain useful for the foreseeable future.

REFERENCES

- [1] G4beamline – <http://g4beamline.muonsinc.com>
- [2] Geant4 Toolkit – <http://geant4.cern.ch>
- [3] Inspired by HistoScope – <http://fermitools.fnal.gov/abstracts/histoscope/abstract.html>
- [4] ROOT – <http://root.cern.ch>
- [5] Mu2e – <http://mu2e.fnal.gov>
- [7] Doxygen – <http://www.doxygen.org>
- [7] <http://www.nist.gov/srd/materials.htm>
- [7] Orbit – <http://neutrons.ornl.gov/APGroup/Codes/orbit.htm>
- [7] TREDI – <http://www.tredi.enea.it>
- [10] PARMELA – http://laacg1.lanl.gov/laacg/services/serv_codes.phtml#parmela