

# G4BEAMLIN PARTICLE TRACKING IN MATTER DOMINATED BEAM LINES\*

T. J. Roberts<sup>#</sup>, K. B. Beard, Muons, Inc., Batavia, IL, USA  
 S. Ahmed<sup>†</sup>, D. Huang<sup>§</sup>, D. M. Kaplan, 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. It has both graphical and command-line user interfaces. 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. A major new feature is the calculation of space-charge effects. G4beamline 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 designs by physicists without the burden of C++ programming. An important aspect of G4beamline is that its description of the simulated system is far more comprehensible by physicists than the corresponding Geant4 C++ code would be, and unlike the code, it is 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 the input file without reference to the G4beamline documentation, and can learn how to

\* Work supported in part by USDOE STTR Grant DE-FG02-08ER86281.  
<sup>#</sup> [tjrob@muonsinc.com](mailto:tjrob@muonsinc.com)

<sup>†</sup> Current address: Jefferson Lab, Newport News, VA, USA.

<sup>§</sup> Current address: Shanghai Institute of Applied Physics, China.

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 a hyperlinked index and the beginning of its Help text.

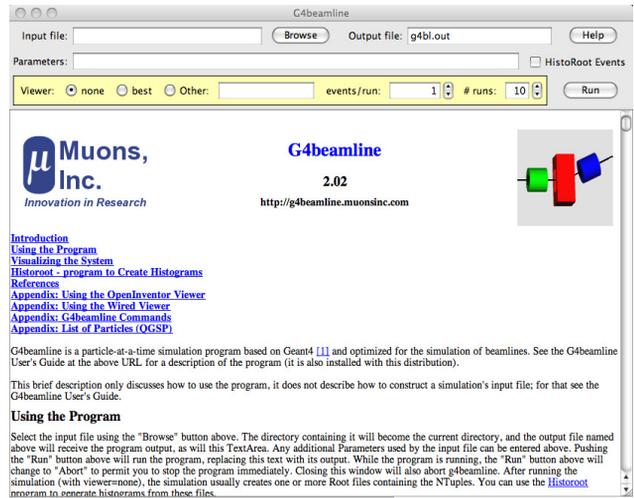


Figure 1: The G4beamline GUI Screen.

To facilitate the generation of histograms and plots, the G4beamline distribution includes the HistoRoot program, which 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 much more usable and efficient.

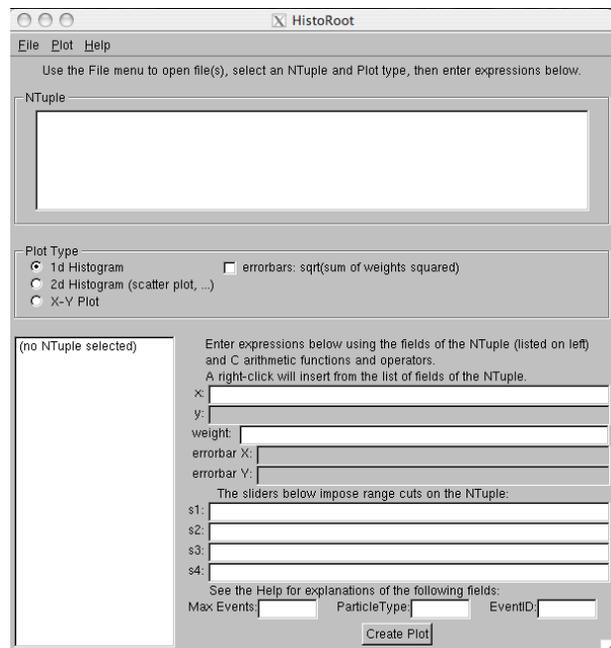


Figure 2: The HistoRoot GUI Screen.

The major aspects of G4beamline are:

- Accurate and realistic simulations using the Geant4 toolkit
- A physicist-readable ASCII input file to specify the simulation, with auxiliary files for field maps, etc.
- Centerline coordinates make it easy to place elements along the beamline
- A rich repertoire of beamline elements suitable for a large class of accelerator, beamline, and detector systems, that can be combined to define new and customized elements
- A general set of initial beam specifications (including the General Particle Source, a cosmic ray muon “beam”, and external files)
- Input and output of beam tracks using several formats including ASCII and ROOT [3] files, supporting easy interfacing to other programs (e.g., for partitioning of complex systems and for 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
- Available for Linux, Windows, and Mac OS systems, as well as source code; it is Open Source
- The HistoRoot program, which makes it easy for non-experts to generate ROOT [3] 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” and “zntuples” can be used to sample the beam at any location.

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 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. 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.

## 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 [4] 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 including numerous tips and techniques to help users apply the program to their problems, get the best simulation performance, and generate results.
2. More Geant4 objects have been implemented, expanding the repertoire of G4beamline.
3. Visualization has been enhanced with the implementation of transparency in objects and the ability to draw E & B field lines
4. Space Charge: current design projections for muon colliders indicate that space charge may be a problem in the final stages of cooling. This computation uses FFTs for the efficient convolution of the Green’s function with the charge distribution on a grid in the beam rest frame; it can use up to about 1 million macro-particles. This includes radical revisions to the code of the Geant4 Run-Manager and EventManager to track particles in parallel for steps in time.
5. Synchrotron radiation is now supported for e and  $\mu$ .
6.  $\mu$ -pair production is now supported.
7. Automated parallelization: Computer farms and multi-CPU systems are now common, and MPI [5] can be used to take advantage of parallelism – a single set of output files contains the computations of multiple processes running on multiple CPUs.

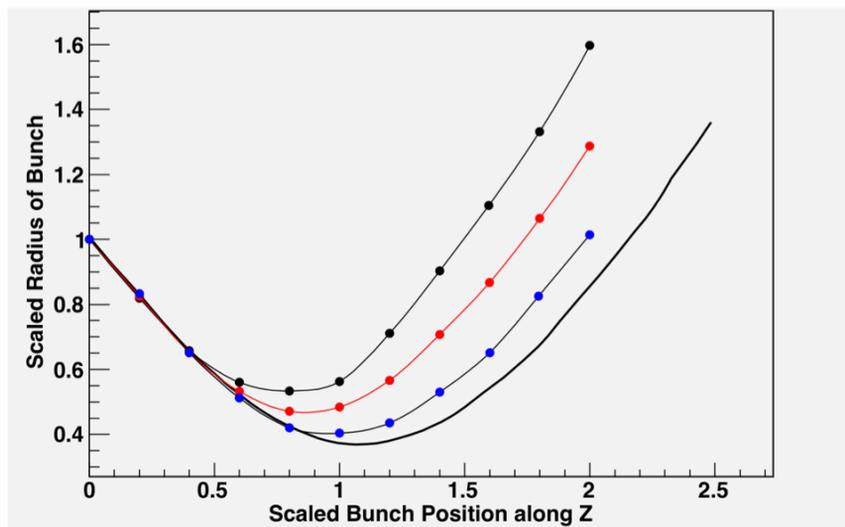


Figure 3: Comparison of G4beamline's space charge computation to a calculation in Reiser's textbook [6], showing scaled transverse beam size as a function of scaled propagation distance  $Z$ . The initial beam is a uniform cylinder which would be focused to a point at  $Z=1$  without space charge. The solid black line is the calculation for a continuous charge density corresponding to  $10^{11} \mu^+$ ; the points are for computations using 1,000 (black), 10,000 (red), and 100,000 (blue) macro-particles, each modeling a bunch also totaling  $10^{11} \mu^+$  (decays disabled).

### SPACE CHARGE VALIDATION

Figure 3 shows a comparison of the G4beamline space charge computation to a calculation from Reiser's textbook [6] of how space charge forces affect the beam size as it propagates. The initial beam is a uniform cylinder with converging trajectories; the space charge defocusing is evident, as without it the beam would be focused to a point at  $Z=1$ . The values plotted are scaled by a function of beam momentum, bunch radius, total charge density, and particle type (see [6]), so the plot is independent of them. Because the macro-particles have huge charges, some macro-particles have experienced an abnormally large interaction with other macro-particles, so the points are above the line, and higher-charge macro-particles are affected more strongly. Overall, the agreement with the calculation is quite good, but requires a large number of macro-particles for accurate agreement with the analytic result.

### "MISSING" PHYSICS PROCESSES

The MAP Collaboration [7] has identified many physics processes that are not in our current tool set; we need to understand each process, and for the important ones either acquire new tools or implement them in our current tools; G4beamline is the prime candidate for the latter.

- Some minor single-particle processes
- Collective effects in vacuum:
  - Wake fields and beam loading
  - Beam-beam interactions
  - Electron cloud effects
  - Decay and interactions of macro-particles
- Collective effects in matter:
  - Space charge screening and dielectric polarization
  - Plasma effects from ionized atoms and their free electrons

- Energy loss and multiple scattering from ionized and excited atoms, and effects from the presence of plasma
  - Intra-beam screening affecting energy loss and multiple scattering
  - Effects of matter on wake fields and electron cloud
- Efforts are underway to analyze and understand these processes; tool acquisition and/or implementation of important ones will follow.

### SUMMARY

G4beamline is a highly flexible and user-friendly program for simulating beamlines both with and without matter. Its 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, current applications include: target hall and test beam design evaluation, feasibility studies for new facilities and experiments, cosmic-ray muon tomography, and detector design, resolution, and background studies. 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] ROOT – <http://root.cern.ch>
- [4] Doxygen – <http://www.doxygen.org>
- [5] MPI – <http://www.open-mpi.org>
- [6] Reiser, M., Theory and Design of Charged Particle Beams, Wiley, 2008.
- [7] The Muon Accelerator Program, <http://map.fnal.gov>