SIMULATION TOOLS FOR THE MUON COLLIDER DESIGN FEASIBILITY STUDY*

Thomas J. Roberts#, Muons, Inc. Batavia IL 60510, U.S.A.
Richard Fernow, Brookhaven National Laboratory, Upton, NY 11973, U.S.A.

Abstract

The U.S. muon collider community is mobilizing itself to produce a "Design Feasibility Study" (DFS) for a muon collider. This is happening on an aggressive schedule and must include the best possible simulations to support and validate the technical design. The DFS for a muon collider will require innovative new approaches to many aspects of accelerator design, and the simulations to support it will require tools with features and capabilities that are equally innovative and new. Two computer programs have emerged as the preferred and most commonly used simulation tools within the muon collider community: ICOOL (primary author: Dr. Fernow), and G4beamline (primary author: Dr. Roberts). We describe the ongoing development and testing of both tools for the DFS, including a common suite of tests to ensure that both tools give accurate and realistic results, as well as innovative user-friendly interfaces with emphasis on graphical user interfaces and windows.

THE DESIGN FEASIBILITY STUDY

This study will involve extensive simulations of new accelerator designs, including the interactions of protons in a target, the decay of pions to muons in a magnetic field, the capture and phase rotation of muon beams in magnetic fields and RF cavities, the passage of muon beams through matter in order to cool them, the acceleration of muon beams to very high energy, and the circulation of muon beams in storage rings and their intersection at the center of a modern particle detector. The understanding of muon decay backgrounds will be an important aspect of these simulations, including the power deposited in accelerator structures, and tracks in the intersection-region detectors. Many of those tasks are beyond the capabilities of traditional accelerator-physics codes, so programs have been developed specifically to perform such simulations. The community feels strongly that at least two such programs should be available, so they can crosscheck each other, increasing the confidence that the results are correct and that facilities designed using the programs will operate as intended.

At present this Design Feasibility Study (DFS) is just starting up. The details of its budget and scope are still being negotiated, and the design of the actual muon collider facility to be considered is essentially undetermined – the first few years of the study will be devoted to evaluating and selecting the components of the final design. It is clear that extensive simulations will be required to do that, and the current plans are to use both ICOOL [1] and G4beamline [2]. The recent history of the muon collider community shows that new ideas for muon cooling are still being invented, and it is likely that during the period of the DFS innovative new ideas will be discovered, evaluated, and perhaps incorporated into the final design of the facility. This implies that the simulation programs used for such evaluations may need to be modified to accommodate the new ideas and concepts. So specific tool development will be necessary to support the DFS, as well as routine bug fixes and program enhancements. As it is important to retain two independent simulation programs (to crosscheck each other), no attempt will be made to merge them; we do intend to implement a common test suite to verify their simulation accuracy with each other and with real experiments (discussed below).

In short, this project is an *enabler* for the much larger DoE project that is the muon collider Design Feasibility Study (DFS). The software support that is the focus of this project will enable all of the participants in the DFS to perform their required simulations more effectively and with greater confidence in their accuracy. As the muon collider itself includes completely new types of accelerator structures (e.g. muon cooling), innovative solutions will be required throughout both projects.

BACKGROUND: ICOOL

ICOOL [1] is a particle beam simulation code that has played a major role in the design of ionization cooling channels for muon colliders and neutrino factories since 1996. It provides the flexibility to quickly examine widely different ideas for configuring muon collider front ends. For example, setting up many preliminary field configurations is accomplished in ICOOL using predefined, analytic field models. This simplifies the adjustment of parameters for the field to obtain some desired result. Comparisons of results from ICOOL and Geant4-based codes have generally shown good agreement. One program is frequently used to check results from the other.

ICOOL uses a command input file consisting of five parts: simulation control variables, beam definition, control of physics interactions, diagnostics, and region definition. The program tracks the particles to the end of a given region and generates any desired diagnostics. It then continues to track the surviving particles to the end of the next region. This program structure was adopted to make it possible to include space charge interactions. There is no practical limit on the number of regions in a problem.

* Work supported in part by USDOE STTR Grant DE-FG02-08ER86281 and DE-AC02-98CH10886.
# tjrob@muonsinc.com

The user can choose to generate Gaussian or uniform initial particle distributions inside the program. Available particles are electrons, muons, pions, kaons and protons of either sign. It is possible to generate a number of important correlations in the initial beam distribution. The input beam information may also be read in from an existing set of particle data on an external file. There is no practical limit on the number of particles. Currently the first 300,000 particles are stored in memory and particles in excess of this are stored on a direct access disk file. The program can save the particle state after any region in the same format required for input, so problems can be run in stages

Muon and pion decays can be simulated. Energy loss, straggling and scattering in matter are simulated as continuous processes. The code currently uses a set of 17 internally defined elements and compounds that are of greatest interest for ionization cooling. The mean value of the energy loss is computed using the Bethe-Bloch formalism including the density effect. Fluctuations in the energy loss may be sampled from Gaussian, Landau or Vavilov distributions. Multiple Coulomb scattering may be sampled from Gaussian distributions, from Rutherford single scattering or from the Moliere distribution.

ICOOL has been used to model some very complicated muon beam channels. These include the 550 m long front end for the Feasibility Study 2 Neutrino Factory design [3] and the 295 m long Study 2a Neutrino Factory design [4]. Modeling the Feasibility Study 2 channel used 1466 separate regions in ICOOL, consisting of liquid hydrogen absorbers with aluminum windows and pillbox RF cavities with beryllium windows, all immersed in a tapered, alternating direction, periodic solenoid field lattice. The code was also used to design the RFOFO ring [5], which was one of the first successful ring coolers for 6D emittance reduction. It has also been used to do the preliminary designs of all the subsystems for the high-emittance muon collider scenario currently being studied for the muon collider.

BACKGROUND: G4BEAMLINE

Geant4 [6] is an internationally supported particle tracking toolkit that was developed to simulate particle interactions in large detectors for high energy and nuclear physics experiments, and includes most of what is known about the interactions of particles and matter, including time-varying electromagnetic fields. G4beamline [2] is a program that provides a highly flexible and user-friendly interface to the Geant4 capabilities relevant to simulating beamlines, requiring no programming by users.

G4beamline was conceived and designed as a flexible interface to the Geant4 toolkit specifically intended to permit the rapid prototyping and evaluation of different accelerator and beamline designs. It has become a “Swiss Army Knife” for Geant4. Its internal programming infrastructure is fully object-oriented, highly modular, and designed to be easy to extend. No changes to its existing infrastructure are expected for this project.

G4beamline uses an ASCII file to specify all aspects of the simulation. It consists of a series of commands that control the simulation, define elements to be used in the simulation, place elements into the simulated world, and direct the generation of results. The complexity of the input file is comparable to the complexity of the system to be simulated (compared to system-specific simulation programs which are much more complex than the system itself). G4beamline has a rather large repertoire of common elements used in particle accelerators and detectors, such as bending magnets, quadrupole magnets, RF cavities, etc. In addition, G4beamline makes it easy to lay out beamlines that involve bending magnets and the resulting rotations of elements – centerline coordinates are available that automatically handle the details so the user does not need to laboriously compute the locations and rotations of beamline elements.

Once the system to be simulated is specified in the input file, particles are tracked using the full accuracy of the Geant4 toolkit. To permit trade-offs between accuracy and computation time, several different models for hadronic interactions are available, and over 100 megabytes of experimental data are available for use by the more detailed models; much of these data are specific to neutrons and photons. The result is that G4beamline can provide a realistic assessment of how the real system will behave. To assist the user in verifying that the simulated system accurately represents the real system, extensive visualization capabilities are included in G4beamline, so the user can see what is being simulated, and where the simulated particles go.

To assist the user in obtaining answers from a simulation run, a user-friendly histogram program is included with G4beamline. Called HistoRoot, it permits the user to easily obtain histograms and plots from the G4beamline output files.

SOFTWARE DEVELOPMENT METHODOLOGY

Modern software development does not occur independent of applications. The effective implementation of new capabilities for ICOOL and G4beamline requires that we use the programs while developing them. So the software development teams will also directly participate in the simulations required as part of the DFS. This ensures that we do not spend time implementing useless features, and also helps us refine and optimize the user interface to the new features. In the software engineering literature this aspect of our methodology is known as “rapid prototyping”, and we often implement one or more prototypes of a new feature before finalizing its implementation.

Experience shows that while this is not directly in line with a top-down management style, in practice the effort expended in re-work and refinement of prototypes is more than compensated by the improved quality of the result. So we cannot describe in detail the steps required to implement the new features. Similarly, we cannot describe the bug fixes that will be implemented.

All software developed in this project will be freely available to participants in the DFS. All G4beamline code will remain open source and freely available on the web. ICOOL is also freely downloadable from the web. New tools will be open source and available on the web unless they rely on software that prevents it.

DEVELOPMENT PLANS

At present we have identified the following items common to both programs:

- A common test suite for ICOOL and G4beamlne.
- A common repository for subsystem designs.
- Common code for assessing cooling channel performance, perhaps based on the program *ecalc9* [7].
- Converters among ICOOL and G4beamline data files, including Root [8] files.
- A common system description, perhaps an automated translation from G4beamline to ICOOL input decks.

ICOOL Plans

We plan to maintain and upgrade ICOOL as necessary, at least through the completion of the Muon Collider Design Feasibility Study. The code should have the capability of simulating any muon beam channel needed for the front end of the collider. This includes channels for muon decay, bunching, phase rotation, transverse cooling, charge separation, 6D emittance cooling, bunch merging, charge recombination, and final cooling to very small muon beam emittances. The DFS calls for a decision on the baseline design of the accelerator and its technical specifications to be made around the end of 2011. The ultimate goal is to use the code to make an end-to-end simulation of the entire collider front end. The results of the front end simulations must be carefully compared with the results of the same channels using G4Beamline, and any discrepancies must be understood.

We will prepare a list of extensions to ICOOL that are needed to perform the desired simulations of the DFS. This list will be maintained for the duration of both this project and the DFS, and they will be implemented as resources permit. Note that the DFS is likely to involve new and innovative techniques, and this list will change over time.

G4beamline Plans

We plan to maintain and upgrade G4beamline for the foreseeable future – it has become an important part of the culture of Muons, Inc., and is being used by a diverse and growing user community. We ultimately intend that the program will be able to simulate every aspect of the muon collider, from the initial gas bottle of the proton driver to the detection of particles in the collider detector. A major goal is to use the code to make an end-to-end simulation of the entire collider front end, the muon accelerator, and the intersecting storage rings; perhaps simulations of the proton driver, the pion production target, and the colliding detector will also be performed. The results of the front end simulations must be carefully compared with the results of the same channels using ICOOL, and any discrepancies must be understood.

We will prepare a list of extensions to G4beamline that are needed to perform the desired simulations of the DFS. This list will be maintained for the duration of both this project and the DFS, and they will be implemented as resources permit. Note that the DFS is likely to involve new and innovative techniques, and this list will change over time.

There is a rather large list of new features suggested by users for G4beamline. We will prioritize them according to their effort and relevance to the DFS, and will implement them as resources permit. The current list is always available at http://g4beamline.muonsinc.com (link at the top).

We intend to implement a Graphical User Interface to G4beamline that is a user-friendly editor for its input files. The user will be presented with a view of the system to be simulated, along with other windows containing libraries of elements, plus the usual toolbars and menus. The user can define new libraries of elements, and add new customized or generic elements to libraries. To construct the system, the user simply drags an element from a library into the world, rotating and placing it as necessary. Each element will have individually editable parameters, and the user will be able to customize the menus, toolbars, libraries, and other aspects of the GUI. Interestingly, the initial library of generic elements will be automatically built from the G4beamline source, eliminating any need to keep the GUI and the program in sync. This new editor will be integrated into the existing GUI that runs G4beamline, and it will be coordinated with the item mentioned above for having a common system description for ICOOL and G4beamline.

CONCLUSION

The muon collider Design Feasibility Study will place demands on the tools it uses to be complete, accurate, and up-to-date. The project described in this paper is an *enabler* for the study, providing support for maintaining those tools.

REFERENCES

[1] R. Fernow, Proc. 2005 Part. Acc. Conf. Knoxville, TN, p. 2651; R. Fernow, Proc. 1999 Part. Acc. Conf., New York, NY, p. 3020.
[2] G4beamline: http://g4beamline.muonsinc.com
[3] Technical Report No. BNL-52623, edited by S. Ozaki, R. Palmer, M. Zisman and J. Gallardo, 2001.
[4] J.S. Berg et al., Phys. Rev. ST-AB 9, 011001 (2006).
[5] R. Palmer et al., Phys. Rev. ST-AB 8, 061003 (2005).
[6] Geant4: http://geant4.cern.ch
[7] Ecalc9: http://nfmcc-docdb.fnal.gov/cgi-bin/ShowDocument?docid=280
[8] Root: http://root.cern.ch