

BEAM DELIVERY SIMULATION: BDSIM AUTOMATIC GEANT4 MODELS OF ACCELERATORS

L. J. Nevay*, S. T. Boogert, H. Garcia-Morales, S. M. Gibson,
R. Kwee-Hinzmann, W. Shields, J. Snuverink

John Adams Institute at Royal Holloway, University of London, Egham, UK
L.C. Deacon, University College London, London, UK

Abstract

Beam Delivery Simulation (BDSIM) is a program that uses a suite of high energy physics software including Geant4, CLHEP & ROOT, that seamlessly tracks particles through accelerators and detectors utilising the full range of particles and physics processes from Geant4. BDSIM has been used to simulate linear colliders such as the International Linear Collider (ILC) and more recently, circular colliders such as the Large Hadron Collider (LHC) [1]. The latest developments including improved geometry modelling; external geometry support; process biasing; and a new event display are presented. A significantly revised and improved accompanying tool chain is presented comprising of a series of Python utilities that allow efficient and automatic preparation of models. Furthermore, a library for both ROOT and Python that provides powerful analysis and event viewing after simulation is demonstrated.

INTRODUCTION

A common task for ensuring the desired operation of an accelerator is to predict the beam losses throughout the machine to avoid damage to the accelerator and surrounding equipment; reduce radioactivation; and minimise background radiation to experiments. Beam loss simulations often make use of multiple different specialist codes resulting in simplifications in each part and the transfer and translation of data. For simplicity and increased accuracy, a combined approach is required that allows both efficient tracking of particles as well as the particle interactions with surrounding material.

BDSIM has been developed for this precise purpose [2, 3]. It is an open source C++ particle tracking code that uses the Geant4 framework [4], which gives access to many electromagnetic and hadronic interaction models as well as powerful geometry description and visualisation tools. For the vacuum volumes, known thick lens transfer maps and integrators are provided rather than general 4th order Runge Kutta integrators of Geant4. BDSIM uses a MADX-like ASCII input syntax, which in combination with the supplied Python tools makes it possible to build models in minutes.

BDSIM was originally developed for the simulation of linear colliders such as the ILC, but in recent years it has seen marked development and modernisation to simulate machines such as the LHC [1]. Presented here are the latest developments.

* laurie.nevay@rhul.ac.uk

MODULAR PHYSICS AND BIASING

BDSIM now uses the modular physics list construction scheme provided by Geant4. The user can now specify the physicsList option with multiple physics lists delimited by a space and these will be built together.

```
option, physicsList="em QGSP_BERT muon"
```

This provides great flexibility and allows future sets of physics processes to be very quickly provided in BDSIM as Geant4 is updated.

In addition, a new interface to Geant4 biasing has been introduced. This allows a user to define a biasing 'object' that defines, for a given particle, which processes should be biased; by how much; and whether for primary, secondary or all instances of that particle. This bias object can then be attached to the vacuum or material volumes of a given beam line element.

With this new ability, the user can more efficiently simulate rare processes such as muon production from e^+ annihilation or even inelastic scattering from residual beam gas. The beam gas interaction would permit beam induced detector background to be studied, or innovatively, the contribution of beam gas interaction to halo generation. By upscaling beam gas interactions, the long term diffusion of a beam in a storage ring could be simulated over a much shorter tracking period.

DIRECT INJECTION

Given recent refactorisation of the construction process in BDSIM, it was possible to easily traverse and interpolate the curvilinear coordinate system of the BDSIM model. Using this capability, a direct injection facility was introduced that allows a particle to be started not only from the beginning but from any s , x , y curvilinear coordinates. BDSIM converts this to the global Cartesian coordinates used in the Geant4 model.

This new feature allows initial particle vertices generated by external software or studies to be used with BDSIM. For the LHC, SixTrack collimation tracking studies are performed and produce maps of where protons meet the mathematical definition of the aperture. These 'hits' can now be transferred to BDSIM and simulated from that point onwards creating an energy deposition map.

EXTENDED GEOMETRY LIBRARY

BDSIM provides a library of parameterised Geant4 geometry for each accelerator component. This has been greatly

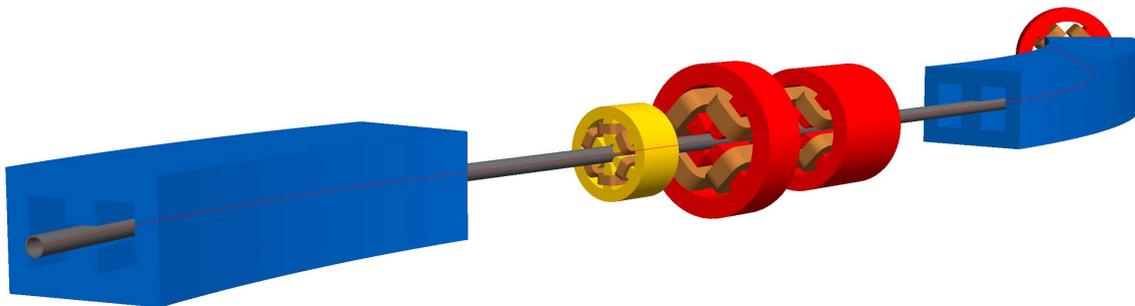


Figure 1: An example beam line built by BDSIM using the *polescircular* geometry style with a single particle passing through (red line).

extended over the past year and now includes 9 aperture formats, 6 magnet geometry formats and 5 tunnel geometry formats. This flexibility is not merely aesthetic, but crucial to achieving correct simulation results as magnet poles present a large depth of material to a particle, where as the gap between poles allows free passage.

With the many different geometry styles for various components, the `BDSGeometryComponent` class that contains geometry objects and manages their extents allows each part of geometry to be constructed independently. Each geometry style is provided by a factory that implements required construction methods based on a simple specification class. BDSIM provides great flexibility in geometry styles while protecting the user from the complexity of Geant4 C++ geometry construction and debugging. It also makes the geometry easily extendable with minimal new code. A small example beam line with the *polescircular* geometry style is shown in Fig. 1.

External Geometry

Should the provided generic parameterised geometry not be sufficient or more accurate geometry is available, external geometry can now be imported via the Geometry Description Markup Language (GDML) file format [6]. Many computer aided design programs can export engineering models to GDML format, or alternatively, GDML can be hand-written relatively quickly. GDML geometry can be inserted into the BDSIM beam line. Figure 2 shows a GDML model of a copper cavity beam position monitor [7] that was hand prepared.

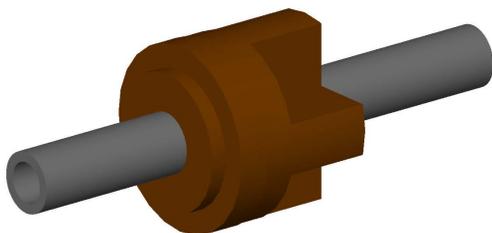


Figure 2: Cavity beam position monitor provided by external geometry in a GDML file in a small example beam line with two drift pieces before and afterwards.

Pole Face Rotations

Although BDSIM provides rectangular and sector bend magnets, many accelerators feature dipole magnets that do not correspond to either of these cases exactly, but instead have a finite end face rotation with respect to the design trajectory. These pole face rotations affect the optical functions of a beam line and therefore it was necessary to provide these in BDSIM. The parameterised geometry allows pole face rotations of up to 45° to be constructed. Unlike most tracking codes, no special tracking algorithms are required for these magnets in BDSIM as the 3D nature provides a physically longer or shorter magnet across the beam width. Figure 3 depicts how the short straight sections that make up a long dipole are modified to accommodate different pole face rotations.

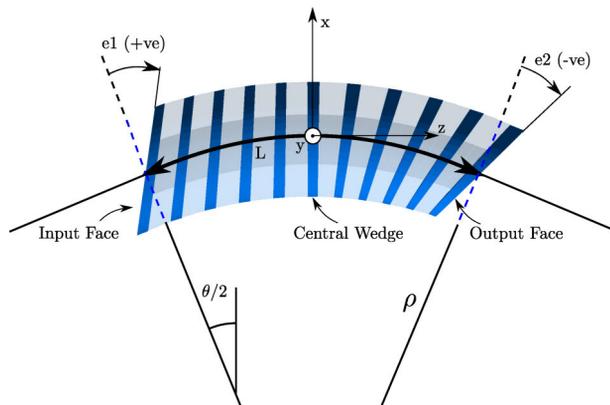


Figure 3: Dipoles in BDSIM are constructed of many small straight sections. Here, the gradual variation from one pole face angle to another is shown.

Tunnel Geometry

Around points where detectors or experiments are located in an accelerator, there is often significant shielding to mitigate any background radiation from the accelerator reaching the sensitive equipment. To accurately simulate detector backgrounds, tunnel geometry has now been introduced with five different styles available as shown in Fig. 4. The tunnel is built by building sections of around the beam line

once it changes by a certain offset or angle. In future, this will be controllable on an element by element basis.

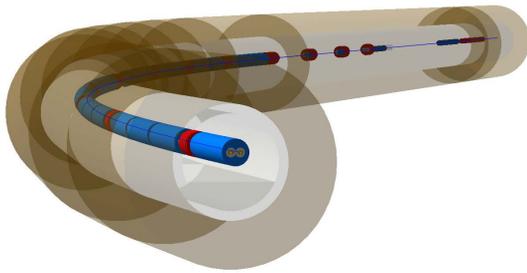


Figure 4: A small section of the Large Hadron Collider modelled in BDSIM showing the capability of producing tunnel geometry that follows the beam line.

The tunnel geometry will allow background radiation from the accelerator to be determined with increased accuracy as the tunnel and surrounding soil will absorb and scatter particles affecting the distribution reaching a sensitive location.

OUTPUT FORMAT & EVENT DISPLAY

BDSIM is already capable of writing output data to the ROOT format, however, recently a new structure has been introduced to allow improved event by event analysis. This provides additional information that wasn't previously stored as well as the complete options used for the simulation for strong reproducibility. Furthermore, placement transforms for each individual beam line element are stored so that only curvilinear coordinates are recorded and global Cartesian coordinates can be calculated during analysis. Despite the additional information, file sizes are reduced due to removal of redundant information. Process information at select points are stored, greatly aiding backtracing losses and their production mechanisms. This will be of great importance when designing new systems as opposed to validating existing ones.

To aid model development and studies, an event display based on the ROOT EVE framework was written as shown in Fig. 5. This allows the new data structure to be displayed event by event and overlaid on top of the geometry generated by BDSIM via a GDML export of the model. Through this visualiser, the user can learn the full history of the event to help them understand the output from the simulations.

TOOLCHAIN

Accompanying BDSIM are several mature C++ and Python packages that support loading of common input sources and data; model preparation; optical analysis; and generalised output analysis. `pymadx` and `pymad8` provide reading and writing capability of MADX and MAD8 lattices respectively. `pybdsim` works in conjunction with these to prepare BDSIM models. `robdsim` is a highly efficient C++ analysis suite that provides interactive or batch analysis of BDSIM ROOT output in both Python and ROOT. In a

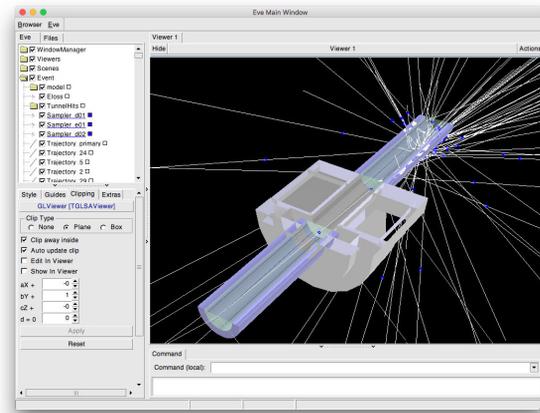


Figure 5: Example event display showing a small lattice in BDSIM that has been exported to a GDML file. The geometry is cut away and the event trajectories are overlaid in the display.

similar fashion, `rebdsim` provides similar functionality for the newer event based BDSIM ROOT output. These are mature pieces of software and are used daily throughout the workflow but continue to be developed.

OUTLOOK

Many aspects of BDSIM have seen significant development, including extended geometry libraries, available physics processes, biasing, output and analysis. These have been implemented in a careful and generalised way such that they are not only applicable to the studies the developers are undertaking, but can be tasked to a wide range of studies. Many of the features described here will be finalised and V1.0 of BDSIM will be released in 2016.

REFERENCES

- [1] L. J. Nevay *et al.*, "LHC Collimation and Energy Deposition Studies using Beam Delivery Simulation (BDSIM)", IPAC'16, Busan, Korea, May 2016, paper MOAB01, this conference.
- [2] I. Agapov *et al.*, "The BDSIM toolkit", EUROTEV-REPORT-2006-014.
- [3] I. Agapov *et al.*, "BDSIM: A particle tracking code for accelerator beam-line simulations including particle-matter interactions", Nucl. Instrum. Methods A 606 (2009) 708.
- [4] S. Agostinelli *et al.*, "Geant4 - a simulation toolkit", Nucl. Instrum. Methods A 506 (2003) 250-303.
- [5] S. B. Boogert *et al.*, "ILC Collimation System Simulation using Beam Delivery Simulation (BDSIM)", IPAC'15, Richmond, Virginia, USA, May 2015, TUPTY065.
- [6] R. Chytracsek *et al.*, "Geometry Description Markup Language for Physics Simulation and Analysis Applications", IEEE Trans. Nucl. Sci., 53, (2006), 5.
- [7] Y. I. Kim *et al.*, "Cavity beam position monitor system for the Accelerator Test Facility 2" Phys. Rev. ST Accel. Beams, 15, (2012), 4.