BDSIM - BEAMLINE SIMULATION TOOLKIT BASED ON GEANT4*

I. Agapov, J. Carter, G. A. Blair (John Adams Institute/RHUL, London, UK),
O. Dadoun (LAL, Orsay, France)

Abstract

BDSIM is a code that combines accelerator-style particle tracking with traditional Geant-style tracking based on Runge-Kutta techniques. This approach means that particle beams can be tracked efficiently when inside the beampipe, while also enabling full Geant4 processes when beam-particles interact with beamline apertures. Tracking of the resulting secondary particles is automatic. The code is described, including a new MAD-style interface and new geometry description, and key performance parameters are listed.

INTRODUCTION

BDSIM is a Geant4 [1] extension toolkit for simulation of particle transport in accelerator beamlines. It provides a collection of classes representing typical accelerator components, a collection of physics processes for fast tracking, procedures of "on the fly" geometry construction and interfacing to ROOT analysis. BDSIM is available for download under http://flc.pp.rhul.ac.uk/bdsim.html. A more detailed description of the code can be found in [2];

The motivation for such a code was the study of backgrounds, collimation system performance and similar problems for the beam delivery system of the Compact Linear Collider (CLIC), and later of the International Linear Collider (ILC) [3], [4]. Such a study requires simulation of beam interaction with material and tracking the primary beam and the secondary particles produced in interactions down the beamline. There are several codes that are capable of performing such a task. However, there are several key issues that make direct application of such codes practically impossible:

- Particle transport over large distances is required. With high energy beams producing a large amount of interactions and secondary particles the performance becomes a critical issue and extensive optimization is required.
- Long beamlines and machine-detector interface require a standardized geometry description procedure specifically designed for this application area.

* Work supported in part by the British Council Alliance programme, the PPARC LC-ABD Collaboration, and by the Commission of European Communities under the 6th Framework Programme Structuring the European Research Area, contract number RIDS-011899

Geant4 is an ideal basis for dealing with such a problem, since it provides an "open-source framework" in the sense that geometry construction, the physics processes used in simulation, sampling and so on can be implemented as user C++ code. In the next section we describe how this idea is implemented in BDSIM.

THE BDSIM CODE

Overall structure

BDSIM is a single-particle tracking code available on Unix and MacOS operating systems with GNU C compilers. It has been tested and adapted to run in PBS and GRID parallel environments. The architecture of BDSIM is sketched in Fig. 1. Geant4 provides general run management framework, a set of physics processes and geometry construction scheme. BDSIM provides a GMAD parser and additional geometry drivers (Mokka, LCDD) that are used to build internal Geant4 geometry representation. This is done with a help of a set of C++ classes that implement accelerator components - quadrupoles, dipoles and so on. Each element has a "stepper" associated with it which implements the particle transportation inside this element. C++ classes that implement various non-Geant4 physics processes are provided.

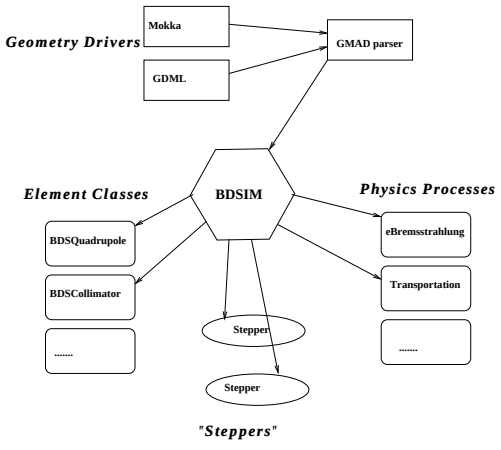

Figure 1: Chart of BDSIM architecture

Lattice description

The beamline, beam properties and physics processes are specified in the input file written in the GMAD [5] language which is a variation of MAD-X [6] language extended to handle sophisticated geometry and parameters relevant to radiation transport. A GMAD program consists of a sequence of element definitions and control commands. For example, tracking a 1 GeV electron beam through a FODO cell will require a file like this:

```
qf: quadrupole, l=0.5*m, k1=0.1;
qd: quadrupole, l=0.5*m, k1=-0.1;
d: drift, l=0.5*m;
fodo : line=(qf,d,qd,d);
use,period=fodo;
beam, particle="e-",energy=1*GeV;
```

GMAD implements almost all the standard MAD elements, but also allows to define arbitrary geometric entities and magnetic field configurations. Another difference from MAD-X is that the beamline need not be a sequence of elements but can be an arbitrary 3d arrangement of components (like in Fig. 3).

The geometry description capabilities are extended by using "drivers" to other geometry description formats which makes interfacing and standardization easier. The most general driver used at the moment is the interface to the Mokka detector description based on SQL. The Mokka driver allows to insert complex geometries such as detector components into the beamline. Fig. 2 shows an example of such a geometry. The lcdd geometry driver [7] is also available.

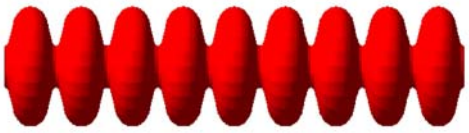

Figure 2: An example of Mokka geometry - an RF cavity

However, standardization of accelerator and detector description formats and a common simulation repository for ILC accelerator studies are currently an open problem requiring further efforts. This means that the BDSIM geometry drivers are expected to evolve extensively in the future.

BDSIM can generate several sorts of beam and beam halo distributions as well as read in the primary particles from a file in a rather wide variety of formats.

Physics processes

BDSIM can exploit all physics processes that come with Geant4. Depending on what sort of problem BDSIM is used for, different sorts of physics processes should be turned on. This processes are grouped into so called "physics lists". The following processes are generally available:

- multiple scattering, ionization, bremsstrahlung, positron annihilation
- gamma conversion
- Compton scattering, photoelectric effect
- synchrotron radiation
- muon production and transport
- neutron and proton elastic and inelastic scattering
- neutron capture
- fission
- radioactive decay

Certain techniques for calculation speed-up are provided:

- biasing schemes are introduced for processes with small cross-sections such as muon pair production.
- controlling production cuts for various reactions in different regions of the geometry is possible
- "fast" physics processes are provided with BDSIM to replace Geant4 ones

Transportation

The accelerator geometry is composed of "components". These could be standard entities such as quadrupoles, bending magnets and so on, and user-defined, such as detector components, collimators with nonstandard apertures etc. Every element is made of several "logical volumes". For instance, a quadrupole can have a beampipe, inner vacuum and iron yoke volumes. Each logical volume is associated with a certain material, a set of physics processes, particle production cuts and so on. The way a particle is transported depends on the logical volume it is in and on the physics processes invoked. When no additional processes are present, a particle is transported in steps that end at boundaries between logical volumes. One step can be produced through a series of smaller substeps to assure the required accuracy. If an element is a quadrupole, drift, or a bending magnet, a step for a primary particle with energy close to nominal inside the beampipe is a mapping given by an appropriate transfer matrix (see f.e. [8])

$$z = M z_0$$

For sextupoles and octupoles a step is accomplished in a series of helical steps which assures the required precision. For multipoles of higher order as well as for fields specified by value maps and for highly off-momentum particles

a Runge-Kutta method of 4-th order is used. This procedure makes tracking of the primary beam faster which is essential for accelerator applications.

Additional processes such as synchrotron radiation may be present. They can be continuous and descrete. For a continuous process the step length is determined in the same way as before, but at the end of the step the particle energy is altered. For a descrete process the step length is determined additionally by the random Monte-Carlo trial which computes the free path. At the end of the step the momentum change is computed and secondary particles produced. The processes present in beam-matter interaction (bremsstrahlung etc.) are invoked only if the logical volume has an associated material other from vacuum.

Validation

Most of the electromagnetic processes in Geant4 and hence in BDSIM are rather accurate, mostly because most of the QED processes are well understood. However, for hadronic physics the situation is quite different and benchmarking with experimental data needs to be performed before a code can be used for application such as shielding and dosimetry. Such activities are now underway withing the Geant4 collaboration and elsewhere.

BDSIM has also been benchmarked against several tracking codes such as MAD [6] for the case of electron transfer lines (see also [9], [10]).

Visualization and output analysis

BDSIM uses standard OpenGL visualization framework provided with Geant4 (an example is shown in Fig. 3). The output of a simulation consists of energy deposition n-tuples and of particle distributions sampled at specified locations in the beamline. This information can be provided either as an ASCII file or in the ROOT [11] format.

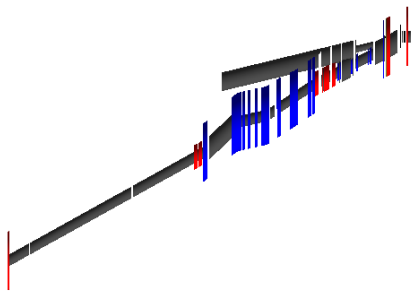

Figure 3: An example of BDSIM visualization - the 2 mrad ILC extraction line with photon and electron dump lines

A set of ROOT scripts for result analysis is provided. An example of a ROOT histogram of power deposition along the accelerator beamline is shown is Fig. 4.

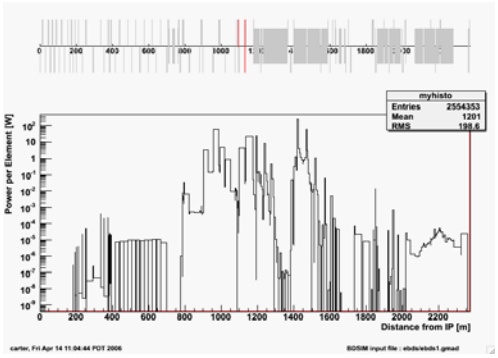

Figure 4: Power deposition along the ILC beam delivery system

CONCLUSIONS

BDSIM has been a useful tool for simulating electron transport in transfer lines. It is constantly improving and an extensive benchmarking program is underway. BDSIM has been succesfully used within the ILC GRID infrastructure [12]. We hope that the the scope of applications and the number of users will continue to grow. The developments which are planned for the future include further benchmarking, extending the geometry drivers necessary for machine-detector interface studies and integrating BDSIM with a CAD tool.

REFERENCES

[1] Geant4 User's guide, http://wwwasd.web.cern.ch/wwwasd/geant4

[2] I. Agapov et al., The BDSIM Toolkit, 2006, EUROTeV-Report-2006-014 (In Review)

[3] J. Carter et. al., Simulation of the ILC collimation system using BDSIM, MARS15 and STRUCT,these proceedings

[4] G. Blair, Simulation of the CLIC Beam Delivery System using BDSIM, CLIC Note 509, 2001.

[5] I. Agapov, GMAD accelerator description language, EUROTeV-Memo-2006-002-1

[6] MAD-X User's Guide, http://mad.home.cern.ch/mad/uguide.html

[7] http://www.lcsim.org/software/lcdd/

[8] A. Chao and M. Tigner, Handbook of Accelerator Physics and Engineering, Word Scientific 1999

[9] R. Appleby et. al., Benchmarking and Tracking of BDSIM/DIMAD Using the ILC Extraction Lines, these proceedings

[10] R. Amerikas et. al., Investigation of Multipole Errors on the Collimation Efficiency for the ILC Beam Delivery System, these proceedings

[11] ROOT User's guide, http://root.cern.ch/root/doc/RootDoc.html

[12] http://grid.desy.de/ilc/