ISBN: 978-3-95450-168-7 doi:10.18429/JACoW-IPAC2015-MOPJE078

BEAM DELIVERY SIMULATION - RECENT DEVELOPMENTS AND OPTIMISATION[*]

J. Snuverink[†], S. T. Boogert, H. Garcia-Morales, S. M. Gibson, R. Kwee-Hinzmann, L. J. Nevay,
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 particle tracking code that simulates the passage of particles through both the magnetic accelerator lattice as well as their interaction with the material of the accelerator itself. The Geant4 toolkit is used to give a full range of physics processes needed to simulate both the interaction of primary particles and the production and subsequent propagation of secondaries. BDSIM has already been used to simulate linear accelerators such as the International Linear Collider (ILC) and the Compact Linear Collider (CLIC), but it has recently been adapted to simulate circular accelerators as well, producing loss maps for the Large Hadron Collider (LHC). In this paper the most recent developments, which extend BDSIM's functionality as well as improve its efficiency are presented. Improvement and refactorisation of the tracking algorithms are presented alongside improved automatic geometry construction for increased particle tracking speed.

INTRODUCTION

Many particle accelerators need to predict and minimise beam losses throughout the machine to avoid damage, radioactivation of the beamline elements and background radiation. Beam loss simulation studies often make use of several codes for different features. Typically, a fast tracking code is used to record positions where particles hit a collimator or the beam pipe, and these positions are then input into a detailed radiation transport code. Obviously, this approach is not ideal. For example, the geometry description capabilities of such codes are rarely equivalent as the fast tracking code usually has a very limited description of the geometry, the data must be transferred between the two codes, particles that scatter back into the beam pipe might not be considered, etc. For the greatest accuracy, a combined approach is required that allows both efficient transport of particles in the accelerator as well as particle interaction with surrounding material.

For this purpose BDSIM [1,2] has been developed. BDSIM is a flexible, open source C++ particle tracking code that uses the Geant4 framework [3]. Geant4 gives access to many electromagnetic and hadronic interaction models as well as powerful geometry description and visualisation tools. For fast beam tracking inside the beam pipe efficient tracking routines are implemented.

For its input BDSIM deploys a MAD-like input syntax, which has the advantage of an easy, fast and accurate conversion from existing lattices. Additional keywords allow for a more detailed geometry and material description.

BDSIM was originally developed for the simulation of linear colliders such as the International Linear Collider (ILC) and the Compact Linear Collider (CLIC), see e.g. [4,5]. In recent years it has been adapted to accommodate circular accelerators as well [6,7]. BDSIM is in active development and is currently utilised for several accelerators, e.g. AWAKE [8], CLIC [9], ILC [10] and HL-LHC [11].

This paper gives an overview of the latest code developments. More information and the source code can be found on the BDSIM website [12].

GEOMETRY

For accurate background studies it is important to have the right amount of detail in the description of the geometry and material. The geometry will be of particular relevance in simulation studies for laserwire diagnostics and other Compton-photon based diagnostics, whose specification is highly dependent on their detector background environment. The geometry of poles in comparison to a generic cylinder of material for example, allows the passage of radiation between the poles significantly further along the machine.

A beamline element like a quadrupole can have many different designs and consist of various geometry components. However, it will have a beam pipe geometry (round, rectangular, ellipse, etc.), a magnet geometry (cylindrical, with poles, etc.), and possibly a supporting plinth or girder. Previously, the creation of the simulation world in BDSIM was done in a single pass in the direction of the beam, element by element. This meant that a quadrupole also needed to create beam loss monitors situated in the vicinity, a part of the tunnel, and even surrounding soil. Such a philosophy is suitable if one works with only one a few types of geometry and structures, but becomes unmanageable when one wants to add new geometry types or new structures.

To simplify the already large beamline element class `BDSAcceleratorComponent`, a more basic base class `BDSGeometryComponent` has been created. This improved geometry class hierarchy allows for a plug and play nature of components, which is normally not provided with the Geant4 toolkit where the user must take care to comprehend the geometry description and avoid geometrical overlaps.

[*] Research supported by FP7 HiLumi LHC - grant agreement 284404 and by the STFC via the JAI3 grant.
[†] jochem.snuverink@rhul.ac.uk

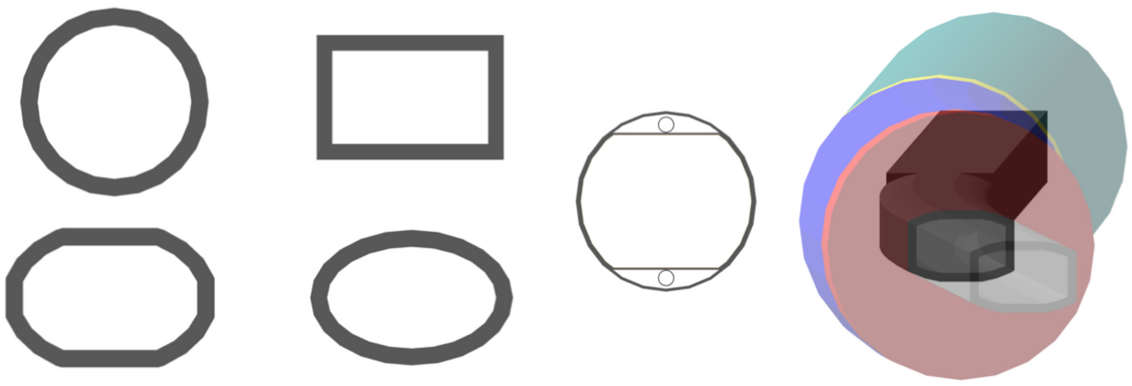

Figure 1: Various new beam pipe models available - (clockwise from *top-left*) circular, rectangular, 'lhcdetailed', an example of various beampipe shapes in a small lattice, elliptical and 'rectellipse'.

Factorisation

Furthermore, the geometry creation code that was present in the `BDSAcceleratorComponent` class has been moved into several so-called 'factories' using the 'Factory Method' and 'Abstract Factory' code patterns. Different factories for the different geometries of beam pipe, magnet geometry, tunnel etc. gives flexibility on how to build up a specific beamline element. It also becomes easy for a user to add a new geometry to the existing code.

Beam Pipe Geometry

An example of the advantages of this code improvement is the beam pipe geometry. Previously, BDSIM allowed for only an elliptical beam pipe shape, with only some components like the collimators that had a different individually specified shape, e.g. rectangular. Also there was a wide range of parameters responsible for the aperture and beam pipe definition.

With the new code structure, every beamline element will use a beam pipe factory to create its beam pipe geometry. The number of aperture parameters has been reduced to four generic parameters analogue to the MADX [13] convention. Beam pipe factories for additional complicated shapes have been implemented without much effort. Available shapes include circular, rectangular, elliptical, 'rectellipse', 'LHC' and 'LHCdetailed', which are shown in Fig. 1. In Fig. 2 a beam pipe with 'LHCdetailed' geometry is shown.

This flexibility in beam pipe and therefore aperture definition is particularly important as the aperture definition throughout a machine strongly affects the beam loss patterns and therefore energy deposition.

Magnet Geometry

Similar to the beam pipe factories, several magnet geometry factories have been implemented. Previously, BDSIM allowed only for cylindrical and magnet with circular poles. Now there are no less than seven different shapes. In Fig. 3 a quadrupole with 'LHCleft' geometry is shown. The suffix 'left' indicates that the magnet body is shifted to the left of the active beamline.

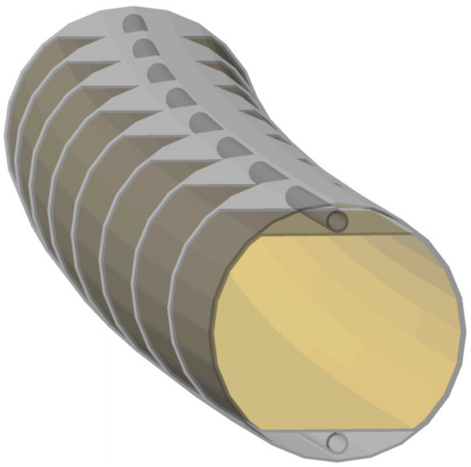

Figure 2: Beam pipe with detailed LHC geometry including 75 μm copper skin and cooling pipes.

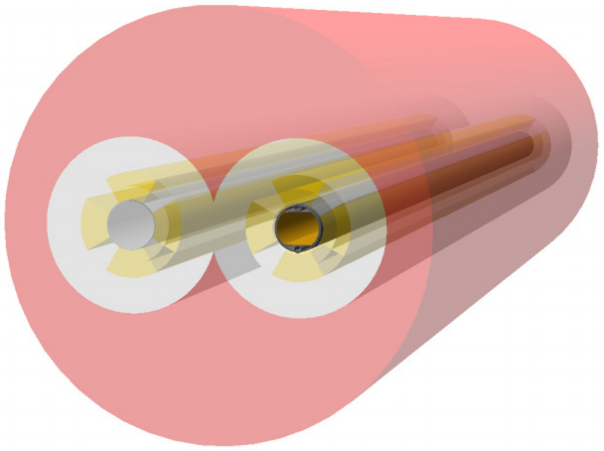

Figure 3: Quadrupole with LHC magnet geometry.

TRACKING

For tracking inside the beam pipe BDSIM uses six dimensional thick tracking routines. These have recently been

ISBN: 978-3-95450-168-7 doi:10.18429/JACoW-IPAC2015-MOPJE078

carefully validated against MADX-PTC [13] with the help of a test suite that was developed.

To improve the tracking speed as well as to facilitate comparisons with other codes, the thick lens, 6D tracking routines are in the process of being factorised into a tracking library. This will allow particles to be tracked solely using the tracking routines without a geometry description and without the navigation overheads in Geant4 as long as the particles are inside of the beam pipe. Symplectic tracking routines, relevant for circular machines, can in this way also be included. Initial tracking studies have shown a reduction in tracking time of approximately an order of magnitude.

Furthermore, the bends in BDSIM are now constructed by an improved method which allows for a, previously impossible, arbitrarily large bending angle by breaking them up into smaller elements. In addition, the new implementation is around 20% faster than the previous implementation.

ANALYSIS

BDSIM has the option to generate output either in ASCII or ROOT format. During the simulation the accumulation of output data was implemented in a way that exposes the output format to the rest of the code. This has been separated by an analysis factory and a uniform output interface, which could allow for easy implementation of new output formats in the future such as HDF5.

Furthermore, an analysis manager has been implemented that accumulates certain distributions in histograms, e.g. the primary and secondary loss maps. While histograms contain less information than raw data, they can provide a substantial reduction in output size. The analysis manager passes these histograms at the end of the run to the output classes to be stored. This allows for greater flexibility on what to store independent of the output format, e.g. the option to store histograms for each beamline element instead of raw data. Addition of further histograms produced during the simulation is now a trivial process. Currently available histograms include primary impact points, primary disintegration points (if hadronic) as well as energy deposition.

MANUAL

With the recent active code development the original manual had become outdated in large parts and was cumbersome to manage. Therefore, a new manual has been written from scratch based on the modern documentation language Sphinx [14] and is close to completion. Sphinx has many user-friendly features, among others many output formats, extensive cross-referencing and a hierarchical structure.

CONCLUSIONS AND OUTLOOK

We report the recent developments of BDSIM, which has seen significant progress in geometry description and many small improvements. The geometry creation has been factorised, which has resulted in several new complicated geometry descriptions and the ability to easily extend this with new designs as may be required by users. In particular, detailed LHC geometry is now available. In future, tunnel and beam loss monitors will be factorised in the same way.

BDSIM is under active development to improve efficiency as required for beam loss simulations of the HL-LHC. The tracking routines have been successfully validated and the tracking is in the process of being factorised, which is expected to reduce computation time significantly. A suite of mature python modules supplied with BDSIM provides easy and fast conversion and preparation of input lattices from established accelerator desicription formats like MAD8 and MADX.

To conclude BDSIM is being actively used, developed and extended providing a flexible, open source code that is used to simulate many different accelerators. This is essential for adoption by users who wish to simulate beam loss and instrumentation sensitive to beam losses.

REFERENCES

[1] I. Agapov *et al.*, “The BDSIM toolkit”, EUROTEV-REPORT-2006-014.

[2] 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.

[3] S. Agostinelli *et al.*, “Geant4 - a simulation toolkit”, Nucl. Instrum. Methods A 506 (2003) 250-303.

[4] H. Burkhardt *et al.*, “Muon backgrounds in CLIC”, IPAC ’10, Kyoto, Japan, May 2010, THPD014.

[5] I. Agapov *et al.*, “Tracking studies of the Compact Linear Collider collimation system”, Phys. Rev. ST Accel. Beams **12** 081001, 2009.

[6] S. T. Boogert *et al.*, “Beam Delivery Simulation (BDSIM): A Geant4 Based Toolkit for Diagnostic and Loss Simulation”, IBIC ’13, Oxford, UK, September 2013, WEPC46.

[7] L. J. Nevay *et al.*, “Beam Delivery Simulation: BDSIM - Development & Optimisation", IPAC ’14, Dresden, Germany, June 2014, MOPRO045.

[8] L. C. Deacon *et al.* “Development of a Spectrometer for Proton Driven Plasma Wakefield Accelerated Electrons at AWAKE", IPAC ’15, Richmond, Virginia, USA, May 2015, WEPWA045, These Proceedings.

[9] F. B. Pilicer *et al.*, “Study of Muon Backgrounds in the CLIC Beam Delivery System", IPAC ’15, Richmond, Virginia, USA, May 2015, TUPTY032, These Proceedings.

[10] S. B. Boogert *et al.*, “ILC Collimation System Simulation using Beam Delivery Simulation (BDSIM)", IPAC ’15, Richmond, Virginia, USA, May 2015, TUPTY065, These Proceedings.

[11] L. J. Nevay *et al.*, “Status of BDSIM developments at RHUL", 4th Joint HiLumi LHC-LARP Annual Meeting, KEK, Japan, November 2014.

[12] BDSIM website: www.pp.rhul.ac.uk/twiki/bin/view/JAI/BdSim

[13] H. Grote and F. Schmidt, “MAD-X: an upgrade from MAD8", PAC 2003, http://mad.web.cern.ch/mad/

[14] http://sphinx-doc.org/