ISBN: 978-3-95450-231-8 ISSN: 2673-5490 doi: 10.18429/JACoW-IPAC2023-WEPA074

BDSIM V1.7.0 DEVELOPMENTS FOR THE MODELLING OF ACCELERATORS AND THEIR ENVIRONMENT *

L. J. Nevay†, A. Abramov, C. Hernalsteens, CERN, Geneva, Switzerland
W. Shields‡, S. Alden, S. M. Gibson, H. Lefebvre, Royal Holloway, University of London, Egham, U.K.
S. T. Boogert, Cockcroft Institute, Daresbury, U.K.
E. Gnacadja, E. Ramoisiaux, R. Tesse, Université Libre de Bruxelles, Bruxelles, Belgium
S. D. Walker, DESY, Hamburg, Germany

Abstract

Beam Delivery Simulation (BDSIM) is a program based on Geant4 that creates 3D radiation transport models of accelerators from a simple optical description in a vastly reduced time with great flexibility. It also uses ROOT and CLHEP to create a single simulation model that can accurately track all particles species in an accelerator to predict and understand beam losses, secondary radiation, dosimetric quantities and their origins. We present a broad overview of new features added to BDSIM in version v1.7.0. In particular, the ability to transform and reflect field maps as well as visualise the fields in Geant4 are presented. A new CT object is introduced to allow DICOM images to be used for simulations of Phantoms in proximity to a beamline. For experiments such as FASER, SHADOWS and NA62, a muon production biasing scheme has been added and is presented.

INTRODUCTION

For any particle accelerator there is a need to understand and model the transport of particles, their possible interaction with matter, any beam losses and radiometric quantities such as dose and activation. The importance may be trivial or crucial to the operation of the machine depending on the application, quantity and species of particle and their energy.

Whilst there are many codes to model the motion of particles through magnetic and electric fields, there are relatively few codes to handle particle-matter interactions. This is due to the sheer complexity in modelling all relevant subatomic particles and physics processes. FLUKA [1], Geant4 [2–4] and MARS [5] are some of the most commonly used codes for this purpose, but users are required to construct a detailed model from primitive shapes. BDSIM is a code that uses the open-source Geant4 C++ toolkit to make 3D models of accelerators and their surroundings [6, 7]. BDSIM provides a library of scalable generic geometry in a variety of styles that can used to quickly build a complex 3D model from an 'optical' description of a beamline (type of object, length, angle, strength, etc.).

BDSIM has been applied successfully to many accelerators [8–11], and thanks to the active development of Geant4 by the community, accurately predicts physical quantities in beamlines. It has been recently applied to medical accelerators with great success [12–14]. Since the release of V1.0, several updated versions have been released with new features as required by the developers and users. v1.7.0 has recently been release and marks a significant step forward with many new features that are described herein.

GENERAL DEVELOPMENTS

Specific developments relating to field maps and muon biasing are described in later sections, but several key general developments and new features are described here.

Spectra

A common task is preparing a particle spectra after a given element in a beamline where the different particle species as a function of kinetic energy are shown. A "sampler" is used in BDSIM to place a plane after a beamline element that records the coordinates of particles as they pass through. Previously, to make a spectra, the user has to make a histogram in kinetic energy for each particle type with a "selection" (i.e. filter) for each individual type desired. A new feature has been introduced where this can be declared in a single line for a given sampler by listing the Particle Data Group identification number (PDG ID) for each species:

```
Spectra samplerName 100 {0:100} {2212,
    ↪ p2212,s2212,2112,211,13,22,11} 1
Spectra samplerName 100 {0:100} {top5} 1
```

The prefix "p" or "s" can be used to optionally filter primary or secondary particles of that type only. In place of a PDG ID, the text `topN` can be used where "N" is an integer for the most common particles observed, e.g. `top5`. This dynamically defines new histograms as new particles are encountered when analysing the data and keeps the N histograms with the highest integrals.

A ROOT TTree selection can also be used (here shown as just "1") to filter the particles included and the full selection string will be automatically built for each histogram understanding multiplicative variables and Boolean filters.

CT Beamline Element

A new computed tomography (CT) beamline element has been introduced in order to use X-ray scan data to build a realistic voxelised phantom that can be used in a BDSIM model to predict dosimetric quantities. Data in the DICOM format [15] is loaded as well as a required Hounsfield Units

* This work is supported by the STFC (UK) grants: JAI ST/P00203X/1, HL-LHC-UK1 ST/N001583/1, HL-LHC-UK2 ST/T001925/1, and ST/P003028/1.
† laurie.nevay@cern.ch
‡ presenting

ISBN: 978-3-95450-231-8 ISSN: 2673-5490 doi: 10.18429/JACoW-IPAC2023-WEPA074

(HU)-to-density and HU-to-material tables. This information is used to create a regular mesh grid of unique materials based on the analysis of the DICOM data.

Samplers and Scorers

As already described, samplers in BDSIM are planes that record the coordinates of individual particles as they pass through. New cylindrical and spherical sampler shapes have been added to allow studies of targets with more useful cylindrical or spherical coordinates. As well as being placed after an individual beamline element, samplers can be placed at any location and orientation as they are implemented in a parallel world. Therefore, they can passively record distributions without affecting the geometry and can conceptually overlap with the regular geometry. For each sampler the shape can be controlled making only part of a cylinder or sphere.

As in any radiation transport Monte Carlo, output data must be carefully chosen as the sheer quantity of information available is difficult to handle. For samplers, the ability to add a particle filter such that only certain particle types are recorded was recently added. This allows efficient storage of, for example, only muons, vastly reducing output data file size.

As well as samplers, 3D scoring meshes now provide a cylindrical scoring mesh as well as the conventional regular cuboidal mesh. The scorers were extended to allow 4D scoring where the 4^{th} dimension is kinetic energy. Custom binning for the kinetic energy can be used to provide differential fluences that can be used in PHITS software. ROOT does not have a 4D histogram, so the Boost histogram library is used [16].

FIELD MAP TRANSFORMS

A key feature of BDSIM is the ability to forgo the provided generic geometry in favour of the user's own custom geometry for a given beamline element. Alongside this, a user-provided magnetic, electric or electro-magnetic field can be overlaid on the geometry (including also over generic geometry). User-provided fields are given by “field maps” that specify field 3-vectors (e.g. B_x, B_y, B_z) in a rectilinear grid. BDSIM allows 1-4D field maps and a variety of interpolation algorithms for 1-4D. For 2D or higher field maps, the memory requirements increase geometrically with the number of points in each dimension. For a highly accurate model, many field maps may be used and the field map density should be carefully chosen to stay within the typical 2 Gb memory limit per CPU of most high energy physics computer clusters. For the most common accelerator magnets, geometrical symmetry can be exploited with reflections to reduce the size of the grid required.

New reflections have been added to BDSIM for this purpose. Such reflections must be compatible with interpolation. The implementation of the reflection is split into two operations. Firstly, an index operator maps spatial coordinates to the reflected grid coordinates in a N-dimensional array in memory. Secondly, a value operator applies a flip to each component of the field depending on the reflection chosen. Although the term “reflection” is generally used, several specific types are used as described with examples in Table 1.

Table 1: Example reflections available in BDSIM with description of the coordinate transforms.

BDSIM Name	Description
flipx	$\pm x \mapsto \mp x$
reflectx	$x \mapsto \lvert x \rvert$
reflectxydipole	Reflect a positive x and y quadrant to all four quadrants

‘reflectxydipole‘ is the BDSIM name for a reflection for a quadrant of an H-shaped dipole. It operates as follows:

- $x \mapsto |x|$, $y \mapsto |y|$
- if $x < 0 \wedge y \geqslant 0$, $B_x \mapsto -B_x$
- if $x \geqslant 0 \wedge y < 0$, $B_x \mapsto -B_x$
- if $y < 0$, $B_z \mapsto -B_z$

where $\wedge$ is the logical `AND`. This is shown visually in Fig. 1.

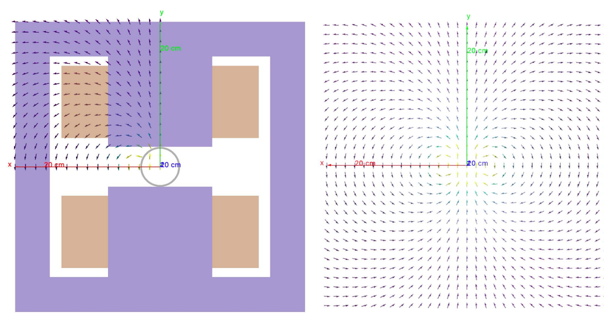

Figure 1: Original dipole field from positive x-y quadrant (left), reflected using “reflectxydipole” (right). The view is with the z axis going into the page and the the coordinate system is right-handed.

To validate field maps, a new feature has been introduced where a user can declare a `query` in the input that defines a grid of points to queried in the model. These can be either in global Cartesian coordinates, or in the local frame of a chosen beamline element. The output of the query can been seen in the visualiser as a field of arrows, but is also written to a BDSIM-format field map file that can be subsequently plotted with the Python utility `pybdsim`, or indeed, reused in a simulation.

MUON SPLITTING

In the environment of a high energy particle accelerator such as the SPS and LHC at CERN as well as fixed-target experiments such as those at the CERN North Area, muons are often produced. High energy muons, despite their short

ISBN: 978-3-95450-231-8 ISSN: 2673-5490 doi: 10.18429/JACoW-IPAC2023-WEPA074

lifetime, can travel far and penetrate large amounts of dense material. They are produced often through the decay of pions. At high momenta, the mean free path of inelastic interactions is significantly shorter than that of decay alone. Despite this, with a high flux of pions found in the above areas, the muon flux cannot be ignored and can be a problematic background for detectors.

Whilst the generation of muons is accurately modelled in Geant4, in such scenarios, it may occur only at 10^{-6} to 10^{-12} per proton collision or per proton on target (i.e. per event). This makes it problematic to accurately estimate the muon flux and spectra when each individual event may take ~0.05-2 s to simulate. Additionally, muons can pass through yokes of magnets and their fields meaning that the estimation of their distribution requires a full Monte Carlo simulation such as that of BDSIM.

Cross-section biasing is available in BDSIM which can be used to make decay more frequent for pions than inelastic hadronic interactions. However, a new feature was introduced called "muon splitting" where when any physics process in Geant4 produces a muon, the process action is resampled N times. Each muon is then weighted by $1/N$. An alternative would be to simply copy the same muon track N times, but this results only in variation seen from the propagation of the muon due to scattering or interactions. By resampling the process a muon with a different momentum and direction may be produced, which more efficiently samples the spatial distribution and spectra at a given location downstream.

The factor N can be chosen to be uniform for all kinetic energies of muons or only apply above a certain kinetic energy. Furthermore, it can be chosen to change linearly between two factors for two given kinetic energies. The value is linearly interpolated then quantised to give an integer number of muons.

Practically, the splitting is implement as a wrapper process. After a physics list is constructed and all processes have been attached to their particle definitions, a specific list of particles and corresponding processes is searched for. If they exist, they are wrapped in the muon splitting class. If the muon splitting class doesn't observe a muon as a secondary particle from the action of the process, it continues passively and returns the secondary particles. Only if a muon is detected in the proposed list of secondaries does it resample the process.

An example comparison is shown in Fig. 2 and Fig. 3 where the transverse distribution of μ^- is shown for the included example with BDSIM called "model-model". Specifically, there is a version of this fictional accelerator called "single-pass" where a small section of straight accelerator with a collimation system for 450 GeV/c primary protons. A sample plane is placed after a set of dipoles and buried in the surrounding earth where a proposed fictional site for a neutrino experiment would be. The muon distribution shows the simulation with splitting achieved a much fuller result for the same number of events simulated the rates are compatible with the unbiased simulation. In both cases, the cross-section biasing feature was used to increase the rate of decays of pions and kaons.

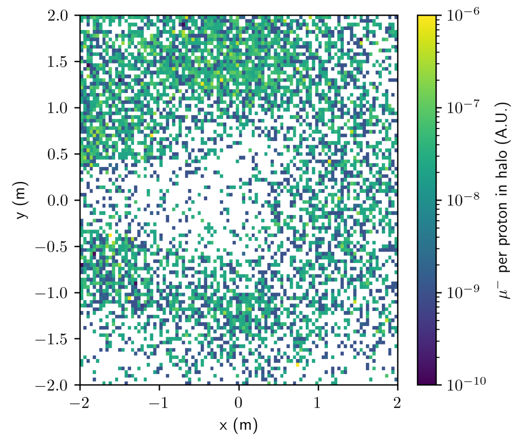

Figure 2: μ^- distribution in model-model example without muon splitting used.

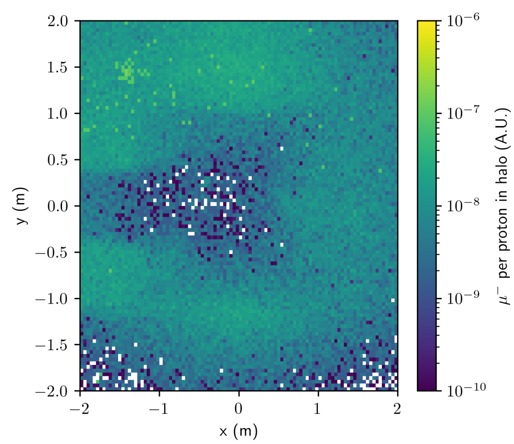

Figure 3: μ^- distribution in model-model example with muon splitting used with a factor of 30 above $E_k = 20$ GeV.

OUTLOOK AND AVAILABILITY

The latest developments released in BDSIM v1.7.0 have been presented including several features that greatly extend the usefulness of the code. Continued development is foreseen to improve aperture descriptions as well as handling multiple beamlines simultaneously.

The software is open-source and available freely, but as source code. A full environment with BDSIM, the associated Python utilities, ROOT and IPython can be found on CVMFS for computers using Centos7. A Dockerfile, and therefore also a Singularity/Apptainer image, are being prepared for a one click solution to using BDSIM. v1.7.0 has been released approximately 2 years after v1.6.0, but a much quicker release cycle is foreseen for subsequent versions.

ISBN: 978-3-95450-231-8 ISSN: 2673-5490 doi: 10.18429/JACoW-IPAC2023-WEPA074

REFERENCES

[1] G. Battistoni *et al.*, Overview of the FLUKA code, Annals of Nuclear Energy 82 (2015) 10-18.

[2] S. Agostinelli *et al.*, Geant4 - A Simulation Toolkit, Nucl. Instrum. Meth. A 506 (2003) 250-303.

[3] J. Allison *et al.*, Geant4 Developments and Applications, IEEE Trans. Nucl. Sci. 53 (2006) 270-278.

[4] J. Allison *et al.*, Recent Developments in Geant4, Nucl. Instrum. Meth. A 835 (2016) 186-225.

[5] N. Mokhov, MARS15. Computer Software. Vers. 00. USDOE Office of Science (SC), High Energy Physics (HEP). 15 Jul. 2016.

[6] L. J. Nevay, *et al.*, BDSIM: An Accelerator Tracking Code with Particle-Matter Interactions, Computer Physics Communications, 252 (2020) 107200.

[7] BDSIM website: `http://www.pp.rhul.ac.uk/bdsim`

[8] J. L. Feng *et al.*, The forward physics facility at the high-luminosity LHC, J. Phys. G: Nucl. Part. Phys. 50 (2023) 030501.

[9] H. Abreu *et al.*, (FASER Collaboration), First neutrino interaction candidates at the LHC, Phys. Rev. D 104, (2021) L091101.

[10] G. Aymar *et al.*, (LhARA Collaboration), LhARA: The Laser-hybrid Accelerator for Radiobiological Applications, Front. in Phys. 8 (2020) 567738,

[11] K. D J. André *et al.*, An experiment for electron-hadron scattering at the LHC, Eur. Phys. J. C 82 (2022) 40.

[12] C. Hernalsteens *et al.*, A novel approach to seamless simulations of compact hadron therapy systems for self-consistent evaluation of dosimetric and radiation protection quantities, EPL 132 (2020) 50004.

[13] E. Gnacadja *et al.*, Optimization of proton therapy eye-treatment systems toward improved clinical performances, Phys. Rev. Research 4 (2022) 013114.

[14] V. Maradia *et al.*, Increase of the transmission and emittance acceptance through a cyclotron-based proton therapy gantry, Med. Phys. 49 (2022) 4, pg 2183.

[15] W. D. Bidgood, S. C. Horii, F. W. Prior, D. E. Van Syckle, Understanding and using DICOM, the data interchange standard for biomedical imaging, J Am Med Inform Assoc. 4(3):199-212 (1997) PMC61235.

[16] `https://www.boost.org/doc/libs/1_82_0/libs/histogram/doc/html/index.html`