ISBN: 978-3-95450-214-1 ISSN: 2673-5490 doi:10.18429/JACoW-IPAC2021-THPAB214

RECENT BDSIM RELATED DEVELOPMENTS AND MODELLING OF ACCELERATORS*

L. J. Nevay†, S. E. Alden, S. T. Boogert, G. L. D'Alessandro,
S. M. Gibson, H. Lefebvre, W. Shields
John Adams Institute at Royal Holloway, University of London, Egham, UK
E. Gnacadja, E. Ramoisiaux, R. Tesse, Université Libre de Bruxelles, Brussels, Belgium
A. Abramov, C. Hernalsteens, CERN, Geneva, Switzerland
S. L. Liu, 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 frame with great flexibility. It also uses ROOT and CLHEP to create a single simulation model that can accurately track all particle species in an accelerator to predict and understand beam losses, secondary radiation, dosimetric quantities and their origin. BDSIM provides a library of scalable generic geometry for a variety of applications. Our Python package, Pyg4ometry, allows rapid preparation and conversion of geometries for BDSIM and other radiation transport simulations including FLUKA. We present a broad overview of BDSIM developments related to a variety of experiments at several facilities. We present a model of the forward experiment FASER at the LHC, CERN where the geometry is composited from multiple sources using Pyg4ometry. The analysis of particle history is presented as well as production mechanisms. We also present the application of recently introduced laser interactions in Geant4 to Compton photons from a laserwire diagnostic at the ATF2.

INTRODUCTION

To understand the origin of beam losses and experimental backgrounds in the environment of a particle accelerator, it is common to use Monte Carlo radiation transport simulations. Observed energy deposition, radioactiviation, or experimental backgrounds are all caused by the interaction of particles with the material of the accelerator and its surroundings as well as the subsequent secondary particles. To predictively simulate these, a physics library such as Geant4 [1] or FLUKA [2] can be used with a 3D model. BDSIM [3, 4], is a program based on the open-source Geant4 physics library that programmatically builds a 3D Geant4 model of an accelerator including accurate accelerator tracking. Geant4 provides a rich catalogue of physics processes and particles extensively used by the experimental particle physics community and is highly validated and regularly maintained.

BDSIM provides a library of scalable generic geometry for common accelerator components that allows an "optical" description of an accelerator (i.e. a list of magnets with lengths and strengths) to create a detailed 3D model in minimal time. Additionally, custom numerical integrators are used to ensure accurate tracking in common accelerator magnets with improved accuracy over conventional numerical integration algorithms.

For accurate results, a 3D radiation transport model must correctly describe the geometry of the environment as material both blocks and produces radiation. Whilst the generic geometry of BDSIM provides an excellent starting point, specific geometry is almost always required for points of interest. For this purpose, a Python library, Pyg4ometry [5], has been developed to rapidly create Geant4 (GDML [6]) and FLUKA geometry as well as convert between them.

Recent developments in BDSIM are presented with example applications and use of Pyg4ometry for complex model preparation.

GEOMETRY

When creating detailed radiation transport models, the geometries required often come from varied sources including other models and software; e.g. GDML (Geant4 XML geometry export format), FLUKA input, CAD, or STL meshes. Pyg4ometry provides the ability to load, convert and composite several of these formats. GDML was designed as part of Geant as a persistency format for a complete model. Pyg4ometry's registry resolves name collision when compositing multiple sources together as well as providing a function to convert "logical volumes" into "assembly volumes". Each piece of geometry is typically contained in large *world* container volume (e.g. a box of air) that would overlap with other geometry if placed together. Conversion to an assembly removes this outer volume whilst preserving relative placements of daughter volumes inside it.

A key feature of Pyg4ometry is the ability to thoroughly check for overlaps in geometry that could cause unseen tracking problems in simulations. Pyg4ometry identifies collisions, coplanar faces and daughter-protrusion as well as provide colour-coded visualisation of these faults. This permits rapid amendment of errors during geometry preparation.

Example Composite Model

A recent example utilising many of these features is that of the Geant4 geometry developed for the FASER experiment [7]. FASER is a forward experiment at the LHC at CERN looking for undiscovered lightly-interacting particles.

* 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@rhul.ac.uk

ISBN: 978-3-95450-214-1 ISSN: 2673-5490 doi:10.18429/JACoW-IPAC2021-THPAB214

It looks along the line of sight from the IP1 collision point through the surrounding tunnel and rock after the LHC accelerator has bent out of the line of sight. The geometry is composed of 3 main sections:

1. A world volume including tunnel and rock;
2. A BDSIM generated beam line of the accelerator;
3. Placements of GDML shielding blocks.

The world volume is externally prepared in Pyg4ometry and provided to BDSIM into which it places the accelerator and other geometry. The world contains the tunnel complex, surrounding soil and the first ~20 m of vacuum pipe with surrounding shielding from the ATLAS experimental cavern. The geometry is composed of a FLUKA model for this part, a BDSIM-generated generic section of tunnel, and a Y-shaped tunnel written in Python using Pyg4ometry classes. The three parts are prepared individually as GDML files and then composited into a single file, again using Pyg4ometry.

Python scripts for four individual concrete shielding blocks were written and the resultant GDML files are included as 'placements' in BDSIM. The complete model is shown in Fig. 1.

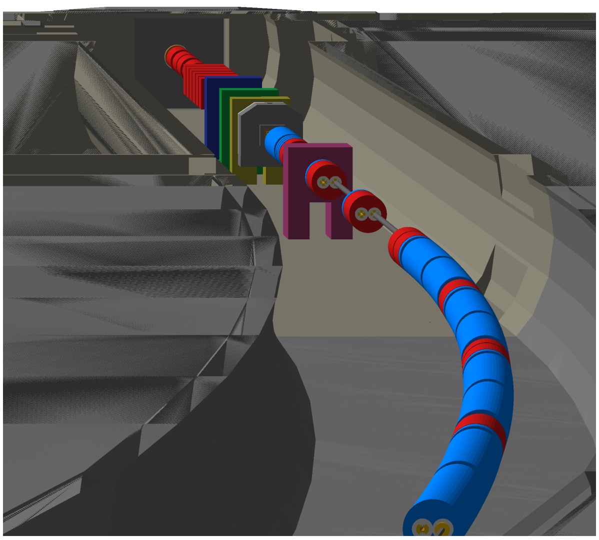

Figure 1: Model of ~500 m of the LHC from IP1 outwards including tunnel complex, shielding and detailed accelerator model in BDSIM. The collision point is hidden behind shielding at the top left and the accelerator comes out of the page. The top half of the geometry is removed to make the accelerator visible.

BDSIM DEVELOPMENTS

Since BDSIM V1.0, development has actively continued and the functionality extended. Several notable developments are described in the following sections. Further developments for medical applications are described in [8].

Process Biasing

BDSIM utilises the generic biasing in Geant4 to allow cross-section biasing and geometric importance biasing. Cross-section biasing allows the cross-section for a list of physics processes for a specific particle to be scaled by unique factors. This can be applied to primary, secondary, or all instances of that particle. Additionally, the biasing only applies in specific volumes that it is attached to. BDSIM now provides several categories of volumes including the accelerator vacuum, the surrounding magnet material, the world volume (e.g. air) and the world contents (e.g. shielding or tunnel volumes).

Recently added is the ability to label volumes from externally provided geometry files as one of these categories permitting highly flexible cross-section biasing. The combinations of Geant4 classes per volume are automatically handled internally in BDSIM before being attached.

Trajectory Handling

Aside from predetermined summary information, the ability to store 'trajectories' of particle tracks of interest is important to understand the interactions of specific particles in the model. The full *tree* of particles is not usually stored as the quantity of information too great. BDSIM now provides several filters to allow selection of specific trajectories of interest. Furthermore, a connection option will store the individual trajectories of parent particles leading back to the parent primary particle. Each trajectory contains a 1D vector of information at each step along the track and each trajectory is stored in a 1D vector in the order it was produced. A map of track ID to storage index is provided to allow any track to be easily accessed. Recent additions include an `std::bitset` describing each filter matched to permit easy filtering of trajectories in analysis as well over the Boolean logic used for the filter result (AND or OR).

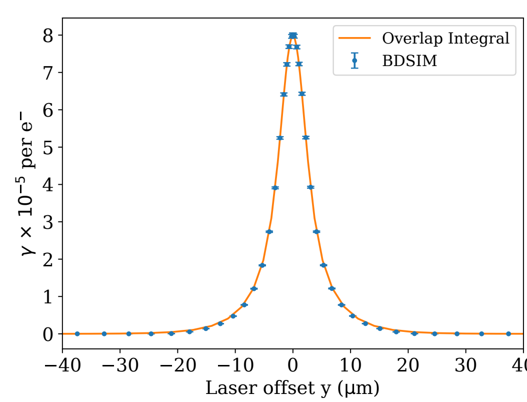

Figure 2: Comparison between simulated Compton-photon rate for various laser beam offsets in BDSIM and a theoretical prediction. The = 532 nm laser beam is focussed to w_0 = 2.3 µm, and the 1.3 GeV e$^-$ beam is 1 × 100 µm (x, y), resulting in a non-Gaussian laserwire scan.

12th Int. Particle Acc. Conf. IPAC2021, Campinas, SP, Brazil JACoW Publishing
ISBN: 978-3-95450-214-1 ISSN: 2673-5490 doi:10.18429/JACoW-IPAC2021-THPAB214

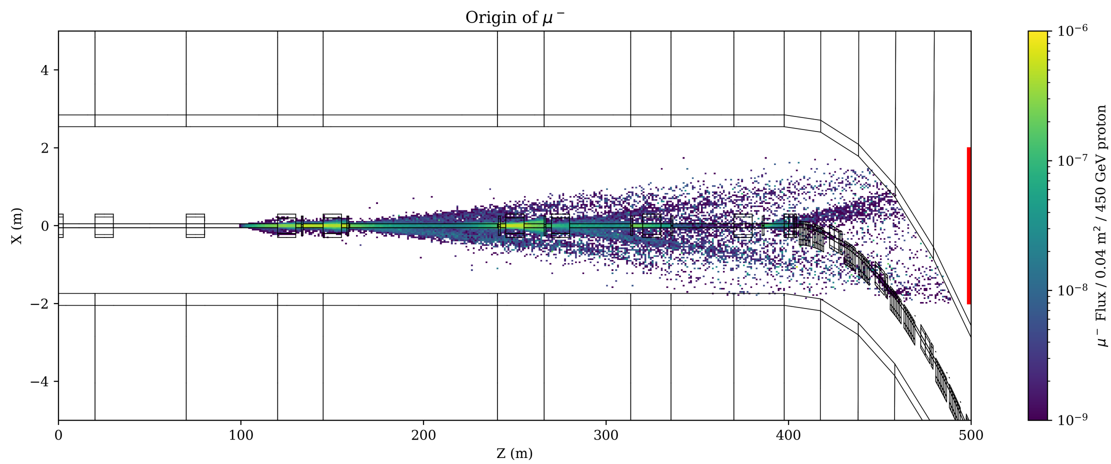

Figure 3: Plan view of the origin of μ^- reaching the (red) sample plane of potential detector location. The flux is normalised per initial 450 GeV proton in a halo distribution impacting collimators (left). The proton beam travels from left to right. An outline (black) of the accelerator plane is shown.

Data Reduction

For detailed simulations, the data volume can be invariably large (GB to TB) with for example, 10^9 events potentially split over thousands of ROOT files. BDSIM now includes two tools for skimming and recombination of data files. `bdskim` uses a *selection* string to filter a raw BDSIM output file's set of events to a smaller one matching some criteria. `bdsimCombine` allows raw bdsim output files to be merged together. Used together, a small, easy to handle, single file for rare events can be produced by skimming the original data (e.g. 1% of original) and then merging these small files together.

Laser Processes

For modelling interactions between in-flight particles and a laser, a specific laser volume was developed for implementation in Geant4 [9]. This allows modelling of the non-uniform laser flux with a multi-step, probabilistic method. In this model, the particle trajectory is discretised into random small steps through the laser volume. Each step is used to calculate an instantaneous probability of interaction used in a Monte Carlo rejection method for initiating the process.

Recent developments in the laser processes introduce an additional approach to modelling the laser interaction for low interaction probabilities. In this approach, a secondary particle is always generated for each particle that enters the laser volume and a weight assigned according to the physical probability. The path of the particle through the laser is used to sample the laser intensity and calculate the cumulative probability. Both model approaches are available for the photodetachment and inverse-Compton scattering processes.

The geometry of the laser and particle beam distribution has significant impact on resulting scan profiles. With strongly focussed laser beams and asymmetric particle beams, the scan can easily be non-Gaussian. These have been accurately reproduced in a setup similar to the ATF2 laserwire experiment at KEK [10] as shown in Fig. 2.

EXAMPLE MODEL

Many of these features are demonstrated in an example model included with BDSIM. "Model-model" is a fictional ~4 km circumference proton synchrotron with two insertion points. A new single pass model is demonstrated here from the collimation section into an arc. A halo of 450 GeV protons in an annulus of 8–12 σ of the beam envelope is simulated hitting the collimation system (apertures at 7 σ). The cross-section of in-flight decay is increased by a factor of 100 for $\pi^\pm$ and $K^\pm$ in material and by a factor 10 in vacuum. The model is designed to estimate the muon flux and spectrum along the line of sight in the rock for a fictional forward location. Such biasing permits predicting rates below the number of events simulated.

A simulation of 60 M events was run and trajectories stored for muons reaching a 2 × 2 m sampling plane 500 m from the start of the collimation insertion. The trajectory information was used to show the origin of muons reaching this plane as shown in Fig. 3.

With the event-by-event ROOT output from BDSIM, it becomes relatively trivial to further select muons originating from a given region and view for example their spectra and production mechanism.

CONCLUSION

Recent developments in BDSIM and PYG4OMETRY are presented permitting an increased level of detail to be achieved in 3D radiation transport models in Geant4 and BDSIM. Flexible process biasing and data handling tools have been introduced to allow efficient handling of large scale simulations and multi-terabyte data sets. Alternative laser-particle methods have also been developed for efficient simulation of low interaction probability models.

ISBN: 978-3-95450-214-1 ISSN: 2673-5490 doi:10.18429/JACoW-IPAC2021-THPAB214

REFERENCES

[1] S. Agostinelli *et al.*, "Geant4 - a simulation toolkit", *Nucl. Instrum. Methods A*, vol. 506, pp. 250-303, 2003. doi:10.1016/S0168-9002(03)01368-8

[2] G. Battistoni *et al.*, "Overview of the FLUKA code", *Annals of Nuclear Energy*, vol. 82, pp. 10-18, 2015. doi:10.1016/j.anucene.2014.11.007

[3] BDSIM, http://www.pp.rhul.ac.uk/bdsim

[4] L. J. Nevay, *et al.*, "BDSIM: An Accelerator Tracking Code with Particle-Matter Interactions", *Comput. Phys. Commun.*, vol. 252, p. 107200, 2020. doi:10.1016/j.cpc.2020.107200

[5] S. T. Boogert, *et al.*, "PYG4OMETRY: a Python library for the creation of Monte Carlo radiation transport physical geometries", 2021. arXiv:2010.01109.

[6] GDML, http://gdml.web.cern.ch/GDML

[7] J. L. Feng *et al.*, "ForwArd Search ExpeRiment at the LHC", *Phys. Rev. D*, vol. 97, p. 035001, 2018. doi:10.1103/physrevd.97.035001

[8] E. Ramoisiaux *et al.*, "BDSIM Developments for Hadron Therapy Centre Applications", presented at the 12th Int. Particle Accelerator Conf. (IPAC'21), Campinas, Brazil, May 2021, paper MOPAB416.

[9] S. E. Alden, S. M. Gibson, and L. J. Nevay, "A Simulation Framework for Photon-Particle Interactions for Laserwires and Further Applications", in *Proc. IPAC'19*, Melbourne, Australia, May 2019, paper THPRB095 pp. 4045–4048.

[10] L. J. Nevay *et al.*, "Laserwire at the Accelerator Test Facility 2 with submicrometer resolution", *Phys. Rev. S.T. Accel. & Beams*, vol. 17, p. 072802, 2014. doi:10.1103/physrevstab.17.072802