The Journal of Open Source Software

Calzone: A Geant4 Python wrapper for the simulation of outdoor particle detectors

Valentin Niess [1¶], **Kinson Vernet**[1], **and Luca Terray** [1,2]

1 Université Clermont Auvergne, CNRS, LPCA, F-63000 Clermont-Ferrand, France. **2** Université Clermont Auvergne, CNRS, IRD, OPGC, Laboratoire Magmas et Volcans, F-63000 Clermont-Ferrand, France. ¶ Corresponding author

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Vangelis Kourlitis

Reviewers:

- @jbeirer
- @peremato

Submitted: 17 December 2024
Published: unpublished

Summary

The transport of high-energy particles (e.g. γ-rays) through ordinary matter is an inherently stochastic process, with individual collisions described within the framework of Quantum Field Theory. The resolution of such transport problems is facilitated by the use of Monte Carlo methods, denoted Monte Carlo Particles Transport (MCPT) herein. In particular, the Geant4 software (Agostinelli et al., 2003; Allison et al., 2006, 2016) is an established MCPT C++ library for simulating the passage of high-energy particles through matter.

Calzone (CALorimeter ZONE) is a MCPT Python package built on top of Geant4. It was developed in the context of geosciences with the objective of studying the emission of radioactivity from volcanoes (Terray et al., 2020), and in particular to simulate the response of gamma spectrometers deployed in the field. To this end, Calzone was developed in conjunction with Goupil (Niess et al., 2024), a backward gamma transport engine, and is interoperable with the latter. Yet, both packages can be used entirely independently, if necessary.

Calzone's interface has been designed with simplicity in mind. Source particles are injected into the simulation volume as a NumPY array (Harris et al., 2020), and a NumPY array of collected energy deposits (or particles) is returned. The Monte Carlo geometry is encoded in a Python `dict`, which can be loaded from configuration files, e.g. using JSON, TOML or YAML formats. This basic workflow is illustrated below,

```
simulation = calzone.Simulation("geometry.toml")
particles = calzone.particles(
    10000,
    pid="gamma",
    energy=0.5,       # MeV
    position=(0,0,1)  # cm
)
deposits = simulation.run(particles).deposits
```

Calzone encourages the use of meshes to describe the Monte Carlo geometry. Various mesh formats are supported, such as OBJ, STL, GeoTIFF and Turtle/PNG (Niess et al., 2020). These formats can be used to encode the components of a detector (exported from a CAD scheme) or a Digital Elevation Model (DEM) describing the surrounding terrain. Additionally, Calzone features an interactive display (`calzone-display`) that allows users to navigate through the Monte Carlo geometry and to inspect Monte Carlo tracks (see e.g. Figure 1).

The Journal of Open Source Software

Statement of need

The Geant4 software was designed as a generic toolkit, with the capability of being extended using the C++ inheritance mechanism. The software is provided under an open-source licence and is subjected to rigorous validation including comparisons with experimental data (Allison et al., 2016). As a result, Geant4 is employed in a multitude of applications, including high-energy physics (its initial scope) and radiation studies (e.g. for medical or space sciences).

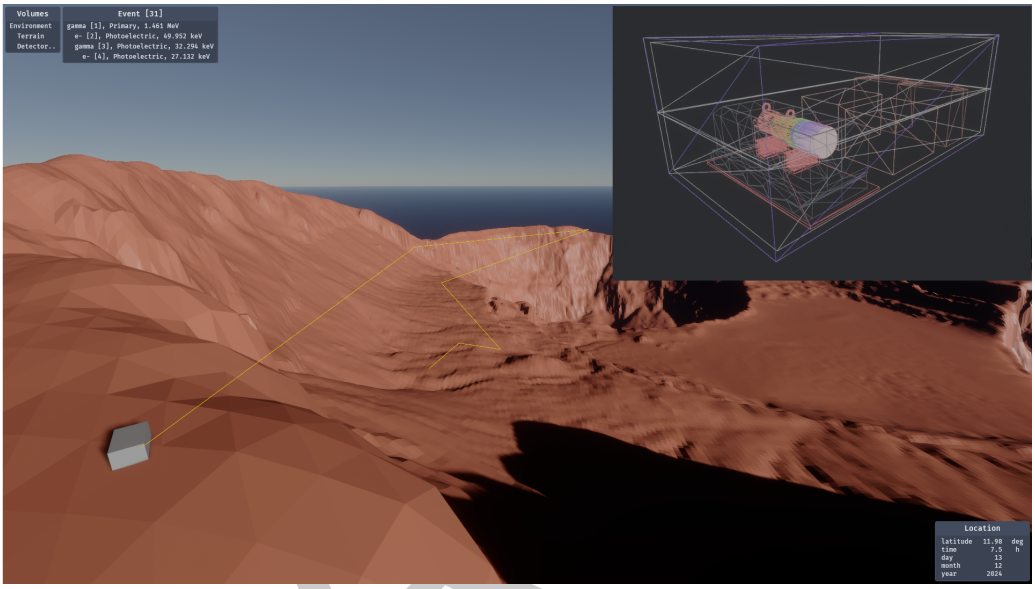

Figure 1: Example of `calzone-display`. The background image comprises a Digital Elevation Model (DEM) of the Masaya volcano, derived from photogrammetry measurements. The grey box on the volcano ridge (bottom-left) corresponds to a gamma-spectrometer (located at $11.983056°$N, $86.172815°$W), the details of which are displayed in the top-right insert (using wireframe mode). The superimposed yellow segments illustrate the trajectory of a photon, originating from the 1.46 MeV emission line of ^{40}K, simulated with Calzone and Goupil in conjunction.

However, the generic nature of Geant4 implies a relatively low-level C++ user interface. Thus, a number of software solutions have been developed on top of Geant4, providing a higher-level user interface and extending its functionalities. This is exemplified by, but not limited to, Gamos (Arce et al., 2014), Gate (Jan et al., 2004; Sarrut et al., 2022), Geant4Py, Gras (Santin et al., 2005) and Topas (Faddegon et al., 2020; Perl et al., 2012).

In the context of geosciences, we encountered specific issues that were not addressed by Geant4, and only partially addressed by some of its derivatives. Some of these issues, which motivated the development of Calzone, are discussed hereafter.

Selected Calzone features

This section outlines a number of key features of the Calzone package, along with the specific issues that these features address.

Native mesh support

The precision of MCPT computations is contingent upon the accuracy of the geometry description. In the context of geosciences, the aforementioned geometry includes the particle detector, which is depicted in a mechanical diagram (using a CAD software), as well as the

study site, which is usually represented by a Digital Elevation Model (DEM). These data are not natively understood by Geant4, requiring transcription. A generic approach is to delineate volumes of the same material (terrain, sensor, mechanical support, etc.) using surfaces approximated by triangular meshes. For instance, the FreeCAD software is able to export the detector parts as STL files, which could then be re-read and transcribed into `G4TessellatedSolids` (e.g., using CADMesh (Poole et al., 2012)). Calzone streamlines this process by defining a geometry format that serves as an intermediary. This format uses standard objects, including `dict`, `float`, `list`, and `str`, and integrates various mesh formats, such as OBJ, STL, GeoTiff and Turtle/PNG (Niess et al., 2020). Calzone then translates this data into Geant4 objects.

Mesh specialisation

The process of meshing a DEM with triangular facets introduces specific issues. To optimise the geometry traversal, the Geant4 software uses a voxelisation algorithm. This method scales poorly for DEMs that typically comprises millions of nodes (see e.g., (Niess et al., 2020)), and is inefficient for long-range particles (such as γ and μ). Thus, Calzone defines a dedicated `Mesh` object that includes a Bounding Volume Hierarchy (BVH) algorithm (partitioning the surface of the mesh, rather than its volume). The user may then select the desired algorithm for each mesh. The default approach is to use a surface BVH for DEMs, while voxelisation is used otherwise (i.e. a `G4TessellatedSolid`).

Interoperability with Goupil

A further distinctive feature of MCPT applications in geosciences (such as gamma-spectrometry and muography) is that the source largely encompasses the detector, which renders analogue simulations ineffective. In a typical use case, only a few dozen out of a million of simulated particles leave a signal in the detector. It is therefore often necessary to rely on Importance Sampling (IS) methods. One effective method in this context is to backward simulate the transport in the detector's far environment (see e.g. (Niess et al., 2018; Niess, 2022)). To this end, Calzone is interoperable with Goupil (Niess et al., 2024).

Particles generator

Another point of interest for MCPT applications is the modelling of particle sources. For this purpose, Calzone provides a geometry-aware `ParticlesGenerator` object, which can, for instance, generate particles entering a specific geometry volume. Moreover, Calzone's `ParticlesGenerator` consistently provides generation weights, which are essential for IS methods.

Software architecture

The Calzone application was developed in Rust, with a Python 3 user interface (using the PyO3 crate). Interfacing with Geant4 was facilitated by the Cxx crate. The interactive visualisation was implemented using the Bevy game engine.

Author contributions

An initial C++ prototype of Calzone was developed by K.V. and V.N. Subsequently, V.N. ported Calzone to Rust and extended its functionalities. L.T. was instrumental in initiating, advising and supervising this project. All authors contributed to the preparation of this manuscript.

Acknowledgements

This is contribution no. 726 of the ClerVolc program of the International Research Center for Disaster Sciences and Sustainable Development of the University of Clermont Auvergne. In addition, We gratefully acknowledge support from the Mésocentre Clermont-Auvergne of the Université Clermont Auvergne for providing computing resources needed for validating this work.

References

Agostinelli, S., Allison, J., Amako, K., Apostolakis, J., Araujo, H., Arce, P., Asai, M., Axen, D., Banerjee, S., Barrand, G., Behner, F., Bellagamba, L., Boudreau, J., Broglia, L., Brunengo, A., Burkhardt, H., Chauvie, S., Chuma, J., Chytracek, R., … Zschiesche, D. (2003). Geant4–a simulation toolkit. *Nucl. Instrum. Methods. Phys. Res. A*, *506*(3), 250–303. https://doi.org/10.1016/S0168-9002(03)01368-8

Allison, J., Amako, K., Apostolakis, J., Araujo, H., Arce Dubois, P., Asai, M., Barrand, G., Capra, R., Chauvie, S., Chytracek, R., Cirrone, G. A. P., Cooperman, G., Cosmo, G., Cuttone, G., Daquino, G. G., Donszelmann, M., Dressel, M., Folger, G., Foppiano, F., … Yoshida, H. (2006). Geant4 developments and applications. *IEEE Trans. Nucl. Sci.*, *53*(1), 270–278. https://doi.org/10.1109/TNS.2006.869826

Allison, J., Amako, K., Apostolakis, J., Arce, P., Asai, M., Aso, T., Bagli, E., Bagulya, A., Banerjee, S., Barrand, G., Beck, B. R., Bogdanov, A. G., Brandt, D., Brown, J. M. C., Burkhardt, H., Canal, Ph., Cano-Ott, D., Chauvie, S., Cho, K., … Yoshida, H. (2016). Recent developments in Geant4. *Nucl. Instrum. Methods. Phys. Res. A*, *835*, 186–225. https://doi.org/10.1016/j.nima.2016.06.125

Arce, P., Ignacio Lagares, J., Harkness, L., Pérez-Astudillo, D., Cañadas, M., Rato, P., de Prado, M., Abreu, Y., de Lorenzo, G., Kolstein, M., & Díaz, A. (2014). Gamos: A framework to do Geant4 simulations in different physics fields with an user-friendly interface. *Nucl. Instrum. Methods. Phys. Res. A*, *735*, 304–313. https://doi.org/10.1016/j.nima.2013.09.036

Faddegon, B., Ramos-Méndez, J., Schümann, J., McNamara, A., Shin, J., Perl, J., & Paganetti, H. (2020). The TOPAS tool for particle simulation, a monte carlo simulation tool for physics, biology and clinical research. *Phys. Med.*, *72*, 114–121. https://doi.org/10.1016/j.ejmp.2020.03.019

Harris, C. R., Millman, K. J., Walt, S. J. van der, Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., Kerkwijk, M. H. van, Brett, M., Haldane, A., Río, J. F. del, Wiebe, M., Peterson, P., … Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, *585*(7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2

Jan, S., Santin, G., Strul, D., Staelens, S., Assié, K., Autret, D., Avner, S., Barbier, R., Bardiès, M., Bloomfield, P. M., Brasse, D., Breton, V., Bruyndonckx, P., Buvat, I., Chatziioannou, A. F., Choi, Y., Chung, Y. H., Comtat, C., Donnarieix, D., … Morel, C. (2004). GATE: A simulation toolkit for PET and SPECT. *Phys. Med. Biol.*, *49*(19), 4543–4561. https://doi.org/10.1088/0031-9155/49/19/007

Niess, V. (2022). The PUMAS library. *Comput. Phys. Commun.*, *279*, 108438. https://doi.org/10.1016/j.cpc.2022.108438

Niess, V., Barnoud, A., Cârloganu, C., & Le Ménédeu, E. (2018). Backward Monte Carlo applied to muon transport. *Comput. Phys. Commun.*, *229*, 54–67. https://doi.org/10.1016/j.cpc.2018.04.001

Niess, V., Barnoud, A., Cârloganu, C., & Martineau-Huynh, O. (2020). TURTLE: A C library

for an optimistic stepping through a topography. *Comput. Phys. Commun.*, *247*, 106952. https://doi.org/10.1016/j.cpc.2019.106952

Niess, V., Vernet, K., & Terray, L. (2024). *Goupil: A monte carlo engine for the backward transport of low-energy gamma-rays*. https://doi.org/10.48550/arXiv.2412.02414

Perl, J., Shin, J., Schümann, J., Faddegon, B., & Paganetti, H. (2012). TOPAS: An innovative proton monte carlo platform for research and clinical applications. *Med Phys.*, *39*(11), 6818–6837. https://doi.org/10.1118/1.4758060

Poole, C. M., Cornelius, I., Trapp, J. V., & Langton, C. M. (2012). A CAD Interface for GEANT4. *Australasian Physical & Engineering Science in Medicine*. https://doi.org/10.1007/s13246-012-0159-8

Santin, G., Ivanchenko, V., Evans, H., Nieminen, P., & Daly, E. (2005). GRAS: A general-purpose 3-d modular simulation tool for space environment effects analysis. *IEEE Transactions on Nuclear Science*, *52*(6), 2294–2299. https://doi.org/10.1109/TNS.2005.860749

Sarrut, D., Arbor, N., Baudier, T., Borys, D., Etxebeste, A., Fuchs, H., Gajewski, J., Grevillot, L., Jan, S., Kagadis, G. C., Kang, H. G., Kirov, A., Kochebina, O., Krzemien, W., Lomax, A., Papadimitroulas, P., Pommranz, C., Roncali, E., Rucinski, A., … Maigne, L. (2022). The OpenGATE ecosystem for monte carlo simulation in medical physics. *Phys. Med. Biol.*, *67*(18), 184001. https://doi.org/10.1088/1361-6560/ac8c83

Terray, L., Gauthier, P.-J., Breton, V., Giammanco, S., Sigmarsson, O., Salerno, G., Caltabiano, T., & Falvard, A. (2020). *J. Geophys. Res. Solid*, *125*(9), e2019JB019149. https://doi.org/10.1029/2019JB019149