

LeptonInjector and LeptonWeighter: A neutrino event generator and weighter for neutrino observatories ☆,☆☆

R. Abbasi^q, M. Ackermann^{be}, J. Adams^r, J.A. Aguilar^l, M. Ahlers^v, M. Ahrens^{av}, C. Alispach^{ab}, A.A. Alves Jr.^{ae}, N.M. Amin^{ao}, R. Anⁿ, K. Andeen^{am}, T. Anderson^{bb}, I. Ansseau^l, G. Anton^z, C. Argüellesⁿ, S. Axani^o, X. Bai^{as}, A. Balagopal V.^{ak}, A. Barbano^{ab}, S.W. Barwick^{ad}, B. Bastian^{be}, V. Basu^{ak}, V. Baum^{al}, S. Baur^l, R. Bay^h, J.J. Beatty^{t,u}, K.-H. Becker^{bd}, J. Becker Tjus^k, C. Bellenghi^{aa}, S. BenZvi^{au}, D. Berley^s, E. Bernardini^{be,1}, D.Z. Besson^{af,2}, G. Binder^{h,i}, D. Bindig^{bd}, E. Blaufuss^s, S. Blot^{be}, S. Böser^{al}, O. Botner^{bc}, J. Böttcher^a, E. Bourbeau^v, J. Bourbeau^{ak}, F. Bradascio^{be}, J. Braun^{ak}, S. Bron^{ab}, J. Brostean-Kaiser^{be}, A. Burgman^{bc}, R.S. Busse^{an}, M.A. Campana^{ar}, C. Chen^f, D. Chirkin^{ak}, S. Choi^{ax}, B.A. Clark^x, K. Clark^{ag}, L. Classen^{an}, A. Coleman^{ao}, G.H. Collin^o, J.M. Conrad^o, P. Coppin^m, P. Correa^m, D.F. Cowen^{ba,bb}, R. Cross^{au}, P. Dave^f, C. De Clercq^m, J.J. DeLaunay^{bb}, H. Dembinski^{ao}, K. Deoskar^{av}, S. De Ridder^{ac}, A. Desai^{ak}, P. Desiati^{ak}, K.D. de Vries^m, G. de Wasseige^m, M. de With^j, T. DeYoung^x, S. Dharani^a, A. Diaz^o, J.C. Díaz-Vélez^{ak}, H. Dujmovic^{ae}, M. Dunkman^{bb}, M.A. DuVernois^{ak}, E. Dvorak^{as}, T. Ehrhardt^{al}, P. Eller^{aa}, R. Engel^{ae}, J. Evans^s, P.A. Evenson^{ao}, S. Fahey^{ak}, A.R. Fazely^g, S. Fiedlschuster^z, A.T. Fienberg^{bb}, K. Filimonov^h, C. Finley^{av}, L. Fischer^{be}, D. Fox^{ba}, A. Franckowiak^{k,be}, E. Friedman^s, A. Fritz^{al}, P. Fürst^a, T.K. Gaisser^{ao}, J. Gallagher^{aj}, E. Ganster^a, S. Garrappa^{be}, L. Gerhardtⁱ, A. Ghadimi^{az}, C. Glaser^{bc}, T. Glauch^{aa}, T. Glüsenskamp^z, A. Goldschmidtⁱ, J.G. Gonzalez^{ao}, S. Goswami^{az}, D. Grant^x, T. Grégoire^{bb}, Z. Griffith^{ak}, S. Griswold^{au}, M. Gündüz^k, C. Haack^{aa}, A. Hallgren^{bc}, R. Halliday^x, L. Halve^a, F. Halzen^{ak}, M. Ha Minh^{aa}, K. Hanson^{ak}, J. Hardin^{ak}, A.A. Harnisch^x, A. Haungs^{ae}, S. Hauser^a, D. Hebecker^j, K. Helbing^{bd}, F. Henningsen^{aa}, E.C. Hettinger^x, S. Hickford^{bd}, J. Hignight^y, C. Hill^p, G.C. Hill^b, K.D. Hoffman^s, R. Hoffmann^{bd}, T. Hoinka^w, B. Hokanson-Fasig^{ak}, K. Hoshina^{ak,3}, F. Huang^{bb}, M. Huber^{aa}, T. Huber^{ae}, K. Hultqvist^{av}, M. Hünnefeld^w, R. Hussain^{ak}, S. In^{ax}, N. Iovine^l, A. Ishihara^p, M. Jansson^{av}, G.S. Japaridze^e, M. Jeong^{ax}, B.J.P. Jones^d, R. Joppe^a, D. Kang^{ae}, W. Kang^{ax}, X. Kang^{ar}, A. Kappes^{an}, D. Kappesser^{al}, T. Karg^{be}, M. Karl^{aa}, A. Karle^{ak}, U. Katz^z, M. Kauer^{ak}, M. Kellermann^a, J.L. Kelley^{ak}, A. Kheirandish^{bb}, J. Kim^{ax}, K. Kin^p, T. Kintscher^{be}, J. Kiryluk^{aw}, S.R. Klein^{h,i}, R. Koirala^{ao}, H. Kolanoski^j, L. Köpke^{al}, C. Kopper^x, S. Kopper^{az}, D.J. Koskinen^v, P. Koundal^{ae}, M. Kovacevich^{ar}, M. Kowalski^{j,be}, K. Krings^{aa}, G. Krückl^{al}, N. Kurahashi^{ar}, A. Kyriacou^b, C. Lagunas Gualda^{be}, J.L. Lanfranchi^{bb}, M.J. Larson^s, F. Lauber^{bd}, J.P. Lazar^{n,ak}, K. Leonard^{ak}, A. Leszczyńska^{ae}, Y. Li^{bb}, Q.R. Liu^{ak}, E. Lohfink^{al}, C.J. Lozano Mariscal^{an}, L. Lu^p, F. Lucarelli^{ab}, A. Ludwig^{x,ah}, W. Luszczak^{ak}, Y. Lyu^{h,i}

☆ The review of this paper was arranged by Prof. Z. Was.

☆☆ This paper and its associated computer program are available via the Computer Physics Communications homepage on ScienceDirect (<http://www.sciencedirect.com/science/journal/00104655>).

E-mail addresses: aschn@mit.edu (A. Schneider), analysis@icecube.wisc.edu (The IceCube Collaboration).

¹ Also at Università di Padova, I-35131 Padova, Italy.

² Also at National Research Nuclear University, Moscow Engineering Physics Institute (MEPhI), Moscow 115409, Russia.

³ Also at Earthquake Research Institute, University of Tokyo, Bunkyo, Tokyo 113-0032, Japan.

W.Y. Ma^{be}, J. Madsen^{ak}, K.B.M. Mahn^x, Y. Makino^{ak}, P. Mallik^a, S. Mancina^{ak}, I.C. Mariş^l, R. Maruyama^{ap}, K. Mase^p, F. McNally^{ai}, K. Meagher^{ak}, A. Medina^u, M. Meier^p, S. Meighen-Berger^{aa}, J. Merz^a, J. Micallef^x, D. Mockler^l, G. Momenté^{al}, T. Montaruli^{ab}, R.W. Moore^y, R. Morse^{ak}, M. Moulai^o, R. Naab^{be}, R. Nagai^p, U. Naumann^{bd}, J. Necker^{be}, L.V. Nguyễn^x, H. Niederhausen^{aa}, M.U. Nisa^x, S.C. Nowicki^x, D.R. Nygrenⁱ, A. Obertacke Pollmann^{bd}, M. Oehler^{ae}, A. Olivas^s, E. O'Sullivan^{bc}, H. Pandya^{ao}, D.V. Pankova^{bb}, N. Park^{ak}, G.K. Parker^d, E.N. Paudel^{ao}, P. Peiffer^{al}, C. Pérez de los Heros^{bc}, S. Philippen^a, D. Pieloth^w, S. Pieper^{bd}, A. Pizzuto^{ak}, M. Plum^{am}, Y. Popovych^a, A. Porcelli^{ac}, M. Prado Rodriguez^{ak}, P.B. Price^h, B. Pries^x, G.T. Przybylski^l, C. Raab^l, A. Raissi^r, M. Rameez^v, K. Rawlins^c, I.C. Rea^{aa}, A. Rehman^{ao}, R. Reimann^a, M. Renschler^{ae}, G. Renzi^l, E. Resconi^{aa}, S. Reusch^{be}, W. Rhode^w, M. Richman^{ar}, B. Riedel^{ak}, S. Robertson^{h,i}, G. Roellinghoff^{ax}, M. Rongen^a, C. Rott^{ax}, T. Ruhe^w, D. Ryckbosch^{ac}, D. Rysewyk Cantu^x, I. Safa^{n,ak}, S.E. Sanchez Herrera^x, A. Sandrock^w, J. Sandroos^{al}, M. Santander^{az}, S. Sarkar^{aq}, S. Sarkar^y, K. Satalecka^{be}, M. Scharf^a, M. Schaufel^a, H. Schieler^{ae}, P. Schlunder^w, T. Schmidt^s, A. Schneider^{ak}, J. Schneider^z, F.G. Schröder^{ae,ao}, L. Schumacher^a, S. Sclafani^{ar}, D. Seckel^{ao}, S. Seunarine^{at}, A. Sharma^{bc}, S. Shefali^a, M. Silva^{ak}, B. Skrzypekⁿ, B. Smithers^d, R. Snihur^{ak}, J. Soedingrekso^w, D. Soldin^{ao}, G.M. Spiczak^{at}, C. Spiering^{be,2}, J. Stachurska^{be}, M. Stamatikos^u, T. Stanev^{ao}, R. Stein^{be}, J. Stettner^a, A. Steuer^{al}, T. Stezelbergerⁱ, R.G. Stokstadⁱ, T. Stuttard^v, G.W. Sullivan^s, I. Taboada^f, F. Tenholt^k, S. Ter-Antonyan^g, S. Tilav^{ao}, F. Tischbein^a, K. Tollefson^x, L. Tomankova^k, C. Tönnis^{ay}, S. Toscano^l, D. Tosi^{ak}, A. Trettin^{be}, M. Tselengidou^z, C.F. Tung^f, A. Turcati^{aa}, R. Turcotte^{ae}, C.F. Turley^{bb}, J.P. Twagirayezu^x, B. Ty^{ak}, M.A. Unland Elorrieta^{an}, N. Valtonen-Mattila^{bc}, J. Vandenbroucke^{ak}, D. van Eijk^{ak}, N. van Eijndhoven^m, D. Vannerom^o, J. van Santen^{be}, S. Verpoest^{ac}, M. Vraeghe^{ac}, C. Walck^{av}, A. Wallace^b, T.B. Watson^d, C. Weaver^x, A. Weindl^{ae}, M.J. Weiss^{bb}, J. Weldert^{al}, C. Wendt^{ak}, J. Werthebach^w, M. Weyrauch^{ae}, B.J. Whelan^b, N. Whitehorn^{x,ah}, K. Wiebe^{al}, C.H. Wiebusch^a, D.R. Williams^{az}, M. Wolf^{aa}, K. Woschnagg^h, G. Wrede^z, J. Wulff^k, X.W. Xu^g, Y. Xu^{aw}, J.P. Yanez^y, S. Yoshida^p, T. Yuan^{ak}, Z. Zhang^{aw}

^a III. Physikalisches Institut, RWTH Aachen University, D-52056 Aachen, Germany^b Department of Physics, University of Adelaide, Adelaide, 5005, Australia^c Dept. of Physics and Astronomy, University of Alaska Anchorage, 3211 Providence Dr., Anchorage, AK 99508, USA^d Dept. of Physics, University of Texas at Arlington, 502 Yates St., Science Hall Rm 108, Box 19059, Arlington, TX 76019, USA^e CTSPS, Clark-Atlanta University, Atlanta, GA 30314, USA^f School of Physics, Center for Relativistic Astrophysics, Georgia Institute of Technology, Atlanta, GA 30332, USA^g Dept. of Physics, Southern University, Baton Rouge, LA 70813, USA^h Dept. of Physics, University of California, Berkeley, CA 94720, USAⁱ Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA^j Institut für Physik, Humboldt-Universität zu Berlin, D-12489 Berlin, Germany^k Fakultät für Physik & Astronomie, Ruhr-Universität Bochum, D-44780 Bochum, Germany^l Université Libre de Bruxelles, Science Faculty CP230, B-1050 Brussels, Belgium^m Vrije Universiteit Brussel (VUB), Dienst ELEM, B-1050 Brussels, Belgiumⁿ Department of Physics, Laboratory for Particle Physics and Cosmology, Harvard University, Cambridge, MA 02138, USA^o Dept. of Physics, Massachusetts Institute of Technology, Cambridge, MA 02139, USA^p Dept. of Physics, Institute for Global Prominent Research, Chiba University, Chiba 263-8522, Japan^q Department of Physics, Loyola University Chicago, Chicago, IL 60660, USA^r Dept. of Physics and Astronomy, University of Canterbury, Private Bag 4800, Christchurch, New Zealand^s Dept. of Physics, University of Maryland, College Park, MD 20742, USA^t Dept. of Astronomy, Ohio State University, Columbus, OH 43210, USA^u Dept. of Physics, Center for Cosmology and Astro-Particle Physics, Ohio State University, Columbus, OH 43210, USA^v Niels Bohr Institute, University of Copenhagen, DK-2100 Copenhagen, Denmark^w Dept. of Physics, TU Dortmund University, D-44221 Dortmund, Germany^x Dept. of Physics and Astronomy, Michigan State University, East Lansing, MI 48824, USA^y Dept. of Physics, University of Alberta, Edmonton, Alberta, T6G 2E1, Canada^z Erlangen Centre for Astroparticle Physics, Friedrich-Alexander-Universität Erlangen-Nürnberg, D-91058 Erlangen, Germany^{aa} Physik-department, Technische Universität München, D-85748 Garching, Germany^{ab} Département de Physique Nucléaire et Corpusculaire, Université de Genève, CH-1211 Genève, Switzerland^{ac} Dept. of Physics and Astronomy, University of Gent, B-9000 Gent, Belgium^{ad} Dept. of Physics and Astronomy, University of California, Irvine, CA 92697, USA^{ae} Karlsruhe Institute of Technology, Institute for Astroparticle Physics, D-76021 Karlsruhe, Germany^{af} Dept. of Physics and Astronomy, University of Kansas, Lawrence, KS 66045, USA^{ag} SNOLAB, 1039 Regional Road 24, Creighton Mine 9, Lively, ON, P3Y 1N2, Canada^{ah} Department of Physics and Astronomy, UCLA, Los Angeles, CA 90095, USA^{ai} Department of Physics, Mercer University, Macon, GA 31207-0001, USA^{aj} Dept. of Astronomy, University of Wisconsin-Madison, Madison, WI 53706, USA^{ak} Dept. of Physics, Wisconsin IceCube Particle Astrophysics Center, University of Wisconsin-Madison, Madison, WI 53706, USA

- ^{al} Institute of Physics, University of Mainz, Staudinger Weg 7, D-55099 Mainz, Germany
^{am} Department of Physics, Marquette University, Milwaukee, WI, 53201, USA
^{an} Institut für Kernphysik, Westfälische Wilhelms-Universität Münster, D-48149 Münster, Germany
^{ao} Bartol Research Institute, Dept. of Physics and Astronomy, University of Delaware, Newark, DE 19716, USA
^{ap} Dept. of Physics, Yale University, New Haven, CT 06520, USA
^{aq} Dept. of Physics, University of Oxford, Parks Road, Oxford OX1 3PU, UK
^{ar} Dept. of Physics, Drexel University, 3141 Chestnut Street, Philadelphia, PA 19104, USA
^{as} Physics Department, South Dakota School of Mines and Technology, Rapid City, SD 57701, USA
^{at} Dept. of Physics, University of Wisconsin, River Falls, WI 54022, USA
^{au} Dept. of Physics and Astronomy, University of Rochester, Rochester, NY 14627, USA
^{av} Oskar Klein Centre, Dept. of Physics, Stockholm University, SE-10691 Stockholm, Sweden
^{aw} Dept. of Physics and Astronomy, Stony Brook University, Stony Brook, NY 11794-3800, USA
^{ax} Dept. of Physics, Sungkyunkwan University, Suwon 16419, Republic of Korea
^{ay} Institute of Basic Science, Sungkyunkwan University, Suwon 16419, Republic of Korea
^{az} Dept. of Physics and Astronomy, University of Alabama, Tuscaloosa, AL 35487, USA
^{ba} Dept. of Astronomy and Astrophysics, Pennsylvania State University, University Park, PA 16802, USA
^{bb} Dept. of Physics, Pennsylvania State University, University Park, PA 16802, USA
^{bc} Dept. of Physics and Astronomy, Uppsala University, Box 516, S-75120 Uppsala, Sweden
^{bd} Dept. of Physics, University of Wuppertal, D-42119 Wuppertal, Germany
^{be} DESY, D-15738 Zeuthen, Germany

ARTICLE INFO

Article history:

Received 12 January 2021

Received in revised form 15 April 2021

Accepted 23 April 2021

Available online 27 May 2021

Keywords:

Neutrino generator
 Event generator
 Neutrino interaction
 Neutrino simulation
 Weighting

ABSTRACT

We present a high-energy neutrino event generator, called *LeptonInjector*, alongside an event weighter, called *LeptonWeighter*. Both are designed for large-volume Cherenkov neutrino telescopes such as IceCube. The neutrino event generator allows for quick and flexible simulation of neutrino events within and around the detector volume, and implements the leading Standard Model neutrino interaction processes relevant for neutrino observatories: neutrino-nucleon deep-inelastic scattering and neutrino-electron annihilation. In this paper, we discuss the event generation algorithm, the weighting algorithm, and the main functions of the publicly available code, with examples.

Program summary

Program Titles: *LeptonInjector* and *LeptonWeighter*

CPC Library link to program files: <https://doi.org/10.17632/662gkpfjd9.1>

Developer's repository links: <https://github.com/icecube/LeptonInjector> and <https://github.com/icecube/LeptonWeighter>

Licensing provisions: GNU Lesser General Public License, version 3.

Programming Language: C++11

External Routines:

- Boost
- HDF5
- nuflux (<https://github.com/icecube/nuflux>)
- nuSquIDS (<https://github.com/arguelles/nuSquIDS>)
- Photospline (<https://github.com/icecube/photospline>)
- SuiteSparse (<https://github.com/DrTimothyAldenDavis/SuiteSparse>)

Nature of problem: *LeptonInjector*: Generate neutrino interaction events of all possible topologies and energies throughout and around a detector volume.

LeptonWeighter: Reweight Monte Carlo events, generated by a set of *LeptonInjector* Generators, to any desired physical neutrino flux or cross section.

Solution method: *LeptonInjector*: Projected ranges of generated leptons and the extent of the detector, in terms of column depth, are used to inject events in and around the detector volume. Event kinematics follow distributions provided in cross section files.

LeptonWeighter: Event generation probabilities are calculated for each Generator, which are then combined into a generation weight and used to calculate an overall event weight.

© 2021 Elsevier B.V. All rights reserved.

1. Introduction

Neutrinos have been measured in a wide energy range from MeV energies in solar and reactor experiments to PeV energies in neutrino telescopes [1]. Different neutrino interaction processes [2] are relevant in this wide energy range, from *e.g.* coherent-neutrino scattering [3] at very small momentum (Q^2) transfer, to very large Q^2 processes which create W bosons [4–8] and heavy quark flavors [9]. However, deep-inelastic scattering [10] is always the dominant process above ~ 10 GeV. This broad energy range has led to the development of various neutrino event generators used

by experiments to simulate neutrino interactions [11–14], most of which have been optimized for GeV neutrino energy ranges and sub-megaton target mass detectors [12]. Such generators are not optimal for gigaton-scale neutrino detectors, often known as neutrino telescopes, such as the currently operating IceCube Neutrino Observatory at the Amundsen-Scott South Pole Station [15] and next-generation observatories such as KM3NeT [16] in the Mediterranean Sea and GVD in Lake Baikal [17].

The first neutrino telescope event generators started their simulation at the Earth's surface [18–21], which required solving two

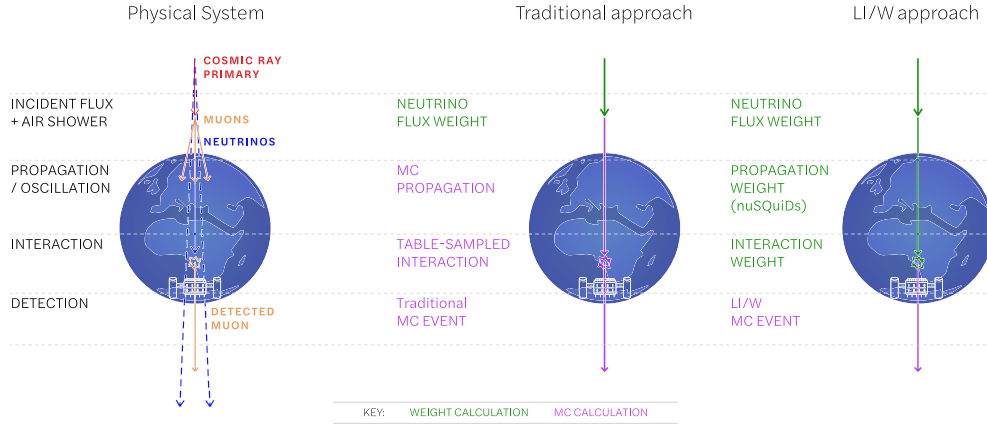


Fig. 1.1. A diagram illustrating the different event generation and weighting steps for traditional methods compared with the *LeptonInjector* and *LeptonWeighter* philosophy.

distinct problems: neutrino transport through the planet and the generation of neutrino events near the sensitive volume. The first such event generator was NUSIM [19], developed in the 1990s for the Antarctic Muon and Neutrino Detector Array (AMANDA); see also [22] for a similar effort for ANTARES. NUSIM established the fundamental concepts of what would later evolve into this project by breaking the problem of event generation into a three-step procedure. First a neutrino energy was randomly drawn from a prior distribution, then forced to interact somewhere near the detector, and finally an event weight would be calculated and applied [19]. This process relied on costly calculations of survival probability of the neutrino through the entirety of the Earth, and tightly coupled the generation and interaction of each neutrino to the calculation of its event weight.

In 2005, NUSIM was ported to C++ and released as the All Neutrino Interaction Simulation (ANIS) [20], and then modified and adopted into the IceCube internal framework [23] as neutrino-generator, or NuGen. The basic simulation scheme remained unchanged, although with each update of the software the scope of features grew and the fundamental features and techniques of the algorithm were further refined and optimized. In the past five years, efficient algorithms to solve the neutrino transport problem have become publicly available [21,24–29], allowing the possibility of simplifying event generation to only consider the problem of event generation in and around a volume near the detector. This allowed the event generation scheme to be separated into two standalone and publicly-available software projects: *LeptonInjector* [30] and *LeptonWeighter* [31]. This separation is not only convenient from software maintenance point of view, but also facilitates optimizations in different energy ranges. For example, IceCube’s analyses focusing in EeV energies [32], where the Earth is opaque to neutrinos, have used the JAVA-based JULeT [21,33] software package for neutrino transport. JULeT, much like the C++-based nuSQuIDS [24,25] package, has the computational advantage of solving Earth propagation using a set of differential equations instead of a Monte Carlo approach. The combination of software presented in this work allows for the user to choose the neutrino transport solution that best suits their needs. This simulation technique, the *LeptonInjector/LeptonWeighter* (LI/W), and traditional techniques are illustrated in Fig. 1.1

In this paper, we will describe the structure and function of the *LeptonInjector* software package, as well as a companion package called *LeptonWeighter*. In Section 2 we describe the basic functionality of *LeptonInjector*, focusing on the structure of the software (Section 2.1), the injection of particles into the detector (Section 2.2), and a comparison between the output of *LeptonInjector* and NuGen. Section 3 contains a description of *LeptonWeighter*

and provides examples of reweighted neutrino samples from various physical sources of neutrinos. We conclude in Section 4. Details of the event and file structures provided by the software packages, as well as example driver scripts, are provided in Supplemental Material.

2. *LeptonInjector*

LeptonInjector is written in C++ with boost-python bindings, and uses Photospline [34] for the cross sections needed for kinematic variable sampling. A standalone version of the code is publicly available from the IceCube GitHub repository [30]. In the description of the software that follows, we use monospace font to refer to libraries and packages, **bold font** to refer to classes, and *italic font* to refer to members of a class.

LeptonInjector is capable of simulating neutrino events of all flavors over a wide range of energies from 10 GeV to 100 PeV and beyond, undergoing neutrino-nucleon interaction in the Deep Inelastic Scattering (DIS) regime and antineutrino-electron scattering producing W in a Glashow Resonance (GR) interaction, ($\bar{\nu}_e + e^- \rightarrow W^-$). The initial event energy is sampled according to a single power-law spectrum at any desired spectral index, and final state kinematics are sampled from spline interpolations of the differential cross sections for the relevant interaction. These splines are saved in FITS files generated by Photospline [34]. For optimum efficiency, the spectrum of generated events would match the physical one. Atmospheric and astrophysical neutrino fluxes, for example, follow a power-law flux. As it is often desirable to maintain large sample size at high energies, events can be generated at one flux and subsequently reweighted to any physical flux using *LeptonInjector*’s sister software package *LeptonWeighter*, available at [31]. Because the event generation starts from near the detector, the primary neutrino energy will be guaranteed to follow the spectral index of event generation; this is not the case for event generators beginning at the Earth’s surface.

To facilitate the reweighting, *LeptonInjector* creates configuration objects complete with a full description of all relevant event generation parameters. *LeptonWeighter* uses these configuration files to reweight events to any desired physical distribution.

Following the event generation, described in this work, the IceCube Monte Carlo proceeds using the following publicly available packages. First, leptons are propagated using PROPOSAL [35], a software package based on MMC [36], while hadronic or electromagnetic showers are propagated by the Cascade Monte Carlo (CMC) package, which implements the physics described in [37] and [38–40]. Next, Cherenkov photons arising from the

Table 2.1

Final state particles. Final state particle types are given, in the right two columns, for various possible desired interactions.

Event type	Interaction	<i>finalType1</i>	<i>finalType2</i>
Nu(E,Mu,Tau)	CC	{E,Mu,Tau}Minus	Hadrons
Nu(E,Mu,Tau)	NC	Nu(E,Mu,Tau)	Hadrons
Nu(E,Mu,Tau)Bar	CC	{E,Mu,Tau}Plus	Hadrons
Nu(E,Mu,Tau)Bar	NC	Nu(E,Mu,Tau)Bar	Hadrons
NuEBar	GR	Hadrons	Hadrons
NuEBar	GR	{E,Mu,Tau}Minus	Nu(E,Mu,Tau)Bar

Table 2.2

Flux properties. Parameter names are given in the left column, their description in the center column, and values on the right column. These parameters are chosen with respect to the desired flux. The allowed energy range is driven by the extent of the provided cross section tables.

Parameter	Description	Allowed ranges
$E_{\nu}^{\min}, E_{\nu}^{\max}$	Neutrino injected energy	[100 GeV, 1 EeV]
γ	Spectral index power law	$(-\infty, \infty)$
$\theta_{\nu}^{\min}, \theta_{\nu}^{\max}$	Injected primary zenith angle	$[0, \pi]$
$\phi_{\nu}^{\min}, \phi_{\nu}^{\max}$	Injected primary azimuth angle	$[0, 2\pi]$

charged particles are simulated by direct photon propagation using CLSIm [41,42] or PPC [43,44]. Finally, a detector response is produced using a proprietary detector simulation.

LeptonInjector requires two kinds of objects: one or more **Injectors** and a **Controller**. An **Injector** object represents one primary neutrino type and one interaction channel, one cross section model to guide interactions, a number of neutrinos to be injected, and one parameter to control the sampling of the interaction vertex: ranged or volume mode, which are described in Section 2.1. In practice, the type of primary neutrino and interaction channel are specified in the **Injector** as a pair of particles that would be generated in such an interaction. These final state particles are called *finalType1* and *finalType2*, and the order of these particles is strictly defined in Table 2.1. A ‘Hadrons’ particle is used to represent the hadronic shower produced by the recoiling nucleus from the DIS interaction and the hadronic decay channel from a W produced in a GR interaction. Ideally the propagation of the hadronic showers would be simulated directly using GEANT4 or a similar particle interaction framework, although this process is far too computationally expensive to be practical. Instead, a parametrization is used for the propagation of the hadronic shower, developed through GEANT4 simulations, as described in [38].

The **Controller** defines energy ranges, azimuth and zenith ranges, and a spectral index of the primary neutrino as shown in Table 2.2. One or more **Injector** objects must be assigned to a **Controller** as well as the destinations for the output files. Once the **Controller** is configured, the simulation is initialized by calling the *Execute* member function. The **Controller** iterates over its member **Injectors**, with each combining the **Controller**’s flux properties with its own injection parameters into a **Generator**, and generating events until reaching their target number as set by the user. This process is illustrated in Fig. 2.1. In both ranged and volume injection modes, **Injectors** generate a ‘primary’ neutrino according to a power-law spectrum with the configured spectral index. This ‘primary’ neutrino is the neutrino as it was in the instant before interaction. The direction of the primary neutrino is sampled uniformly from the allowed ranges in azimuth and cosine of zenith. The event location is selected according to the injection mode, which is described in detail later in Section 2.1.

The event kinematics are defined in terms of the Bjorken x and Bjorken y kinematic variables. As shown in Eq. (2.1), Bjorken y is the fractional energy carried away by the out-going lepton in a DIS interaction and Bjorken x is the fraction of primary-particle mo-

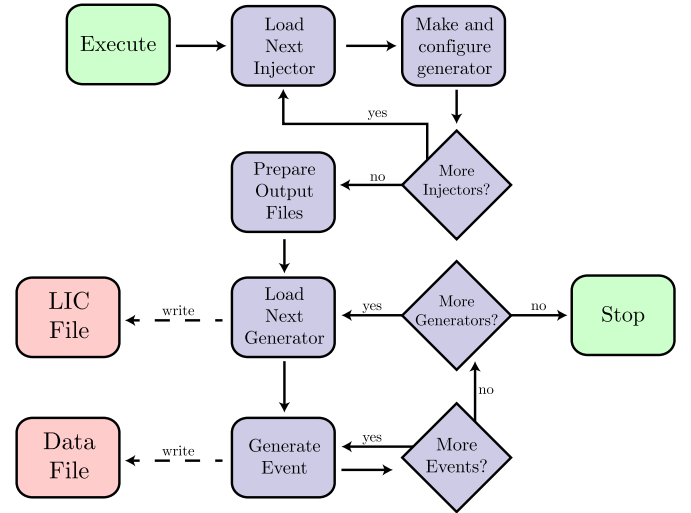


Fig. 2.1. Flowchart displaying the iterative process by a **Controller** prepares to generate, and then generates events. The **Generator** object is used by **LeptonInjector** to store all necessary information to simulate events.

mentum transferred by the weak interaction. These two variables are given by

$$y = 1 - \frac{E_f}{E_i} \quad x = \frac{4E_i E_f \sin^2 \theta}{2m_p (E_i - E_f)}, \quad (2.1)$$

where θ is the angle between the trajectories of the initial- and final-state leptons, m_p the proton mass, and E_i and E_f are the energies of the initial and final state lepton, respectively.

The Bjorken quantities are sampled, using b-splines from a joint 3D probability density function in the logarithm of each of E_i , x , and y space, using the Metropolis-Hastings algorithm [45]. Final state particle energies and deviations from the injected primary direction are then calculated analytically according to these kinematic variables. Events are written to an HDF5 file as they are generated, and a **LeptonInjector** Configuration (LIC) file is written in parallel storing the exact configuration settings used for generation including both differential and total cross sections used. These LIC files are structured binary data files containing a header for meta-data, including a version number, and may evolve over time. The backwards-compatibility of LIC files is of importance, and will be maintained in future versions of **LeptonInjector** and **LeptonWeighter**. The structure of the HDF5 files and exact specifications of the LIC files are described in Appendices A and B, respectively.

2.1. Injection

The two modes for injecting events are ranged mode and volume mode; each accepts and requires the parameters described in Table 2.3. An overview of the injection process is shown in Fig. 2.2.

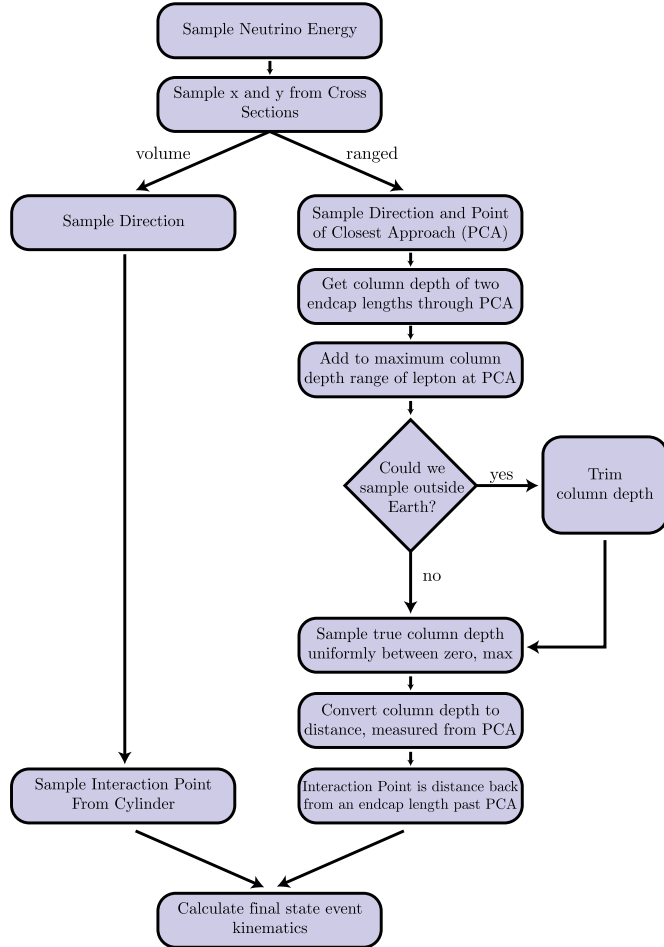
In volume mode, a cylinder, oriented vertically, is constructed around the origin according to specified parameters and an interaction point is selected uniformly within that cylinder’s volume. This injection mode is suitable for simulating events which are approximately point-like for the purposes of detection, such as neutral-current interactions and charged-current ν_e interactions which produce particle showers which are fairly short in dense media compared to the size of the detector.

The ranged mode process is illustrated in Fig. 2.3. This mode is intended as a counterpart of the volume injection mode. Ranged mode is suitable for simulating events where the detection is due to visible daughter particles ($\mu^\pm$, $\tau^\pm$) which travel through

Table 2.3

Detector properties. Parameter names are given in the left column, their description in the middle-left column, and Defaults on the middle-right column. These parameters are chosen in order to cover the detector. Example parameters for the IceCube Neutrino Observatory are provided on the far-right column.

Parameter	Description	Defaults	IceCube example
<i>InjectionRadius</i>	Max distance of closest approach from origin for injection	1200 m	900 m
<i>EndcapLength</i>	Possible longitudinal extent of injection from point of closest approach	1200 m	900 m
<i>CylinderRadius</i>	Radius of injection cylinder	1200 m	700 m
<i>CylinderHeight</i>	Height of injection cylinder	1200 m	1000 m

**Fig. 2.2.** A flowchart demonstrating the process of generating an event.

dense media for distances comparable to or larger than the size of the detector. It ensures sampling of interaction positions, over the whole volume of a target detector, both as far away as possibly visible to the detector due to daughter particles leaving the interaction, and proportional to local material density. A typical example of an interaction type in this category is the charged-current ν_μ interaction.

During generation in ranged mode, a direction for the primary neutrino is first chosen within the allowed range of azimuth and zenith angles; this is shown in Fig. 2.3a. Then, as in Fig. 2.3b, a point is randomly chosen from a disk of radius *InjectionRadius* centered at the origin and perpendicular to the sampled direction; this point will be the point of closest approach (PCA) of the injected neutrino's projected path. The distance from the sampled

PCA to the origin is called the impact parameter. Next, a range of possible positions along this path is determined in which the interaction position may be sampled. This includes two 'endcaps,' specified as lengths (*EndcapLength*) on either side of the disk containing the PCA, and a maximum lepton 'range.' The endcaps are added to ensure that events are sampled over the entire volume of the detector, and the range is computed to account for the maximum distance that the charged lepton daughter of the interaction may travel. This maximum distance is calculated according to

$$R_\mu(E) = \frac{1}{d_b} \log \left(1 + E \frac{d_a}{d_b} \right), \text{ and} \quad (2.2)$$

$$R_\tau(E) = R_\mu(E) + \left(3.8 \times 10^4 \right) \log \left(1 + \frac{1}{5.6 \times 10^7} \right), \quad (2.3)$$

with $d_a = 0.212/1.2$ [GeV mwe⁻¹] and $d_b = (0.251 \times 10^{-3})/1.2$ [mwe⁻¹] [46]. R_μ and R_τ represent the maximum ranges, in meters water equivalent, for 99.9% of muons and taus of energy E in GeV respectively. The factor of 1.2 is to account for the observed deviations from the fits producing these max range functions. The maximum deviation was less than 20%, and as such d_b is appropriately scaled.

The range of possible positions is converted to common units of column depth by taking into account the density of the material along the line formed by the two endcap lengths, including both local material around the detector and the Earth more generally, using a variation of the Preliminary Reference Earth Model (PREM) [47], which we have extended with three uniform-density layers: a 2.6 km thick clear-ice layer, a 200 m thick firn layer, and a 103 km atmosphere layer; these are demonstrated in Appendix C. These extra layers cover the entire Earth, and are required to accurately distribute the events, with respect to depth, in ranged mode. This gives the preliminary maximum column depth within which the generator should ideally sample the interaction point. The geometry of this calculation is shown in Fig. 2.3c. The model of the surrounding material is then integrated again, to determine whether the amount of column depth desired from the preliminary calculation actually exists along the path; at high energies it may not if the lepton range is sufficient to extend outside the outermost layer of the Earth model. In this case, the maximum column depth is reduced to the physically available value, as in Fig. 2.3d. The resulting column depth is called the total column depth. The amount of column depth the neutrino should traverse before interacting is then sampled uniformly between zero and this total column depth, and then converted to a physical position along the chosen path by a final integration of the material model as shown in Fig. 2.3e.

2.2. Comparisons

Prior to this generator, in IceCube, the primary neutrino event generator has been NuGen. Similar to *LeptonInjector*, NuGen has been used to generate both up- and down-going neutrino events of all flavors and neutrino type; it has been used for the Monte Carlo event generation of numerous studies in IceCube and is thoroughly vetted. As part of the development of *LeptonInjector*, comparisons were made between identical MC samples generated by *LeptonInjector* and NuGen's 'Detector Mode,' which uses an injection scheme roughly analogous to *LeptonInjector*'s ranged mode. Events of all flavor, for both neutrino and anti-neutrino primaries, were generated for each neutrino type at a spectrum of E^{-1} over all azimuth angles and up-going zenith angles. A comparison of the spectra of injected lepton energies is shown in Fig. 2.5, and a comparison of the average inelasticity parameter as a function of primary neutrino energy is shown

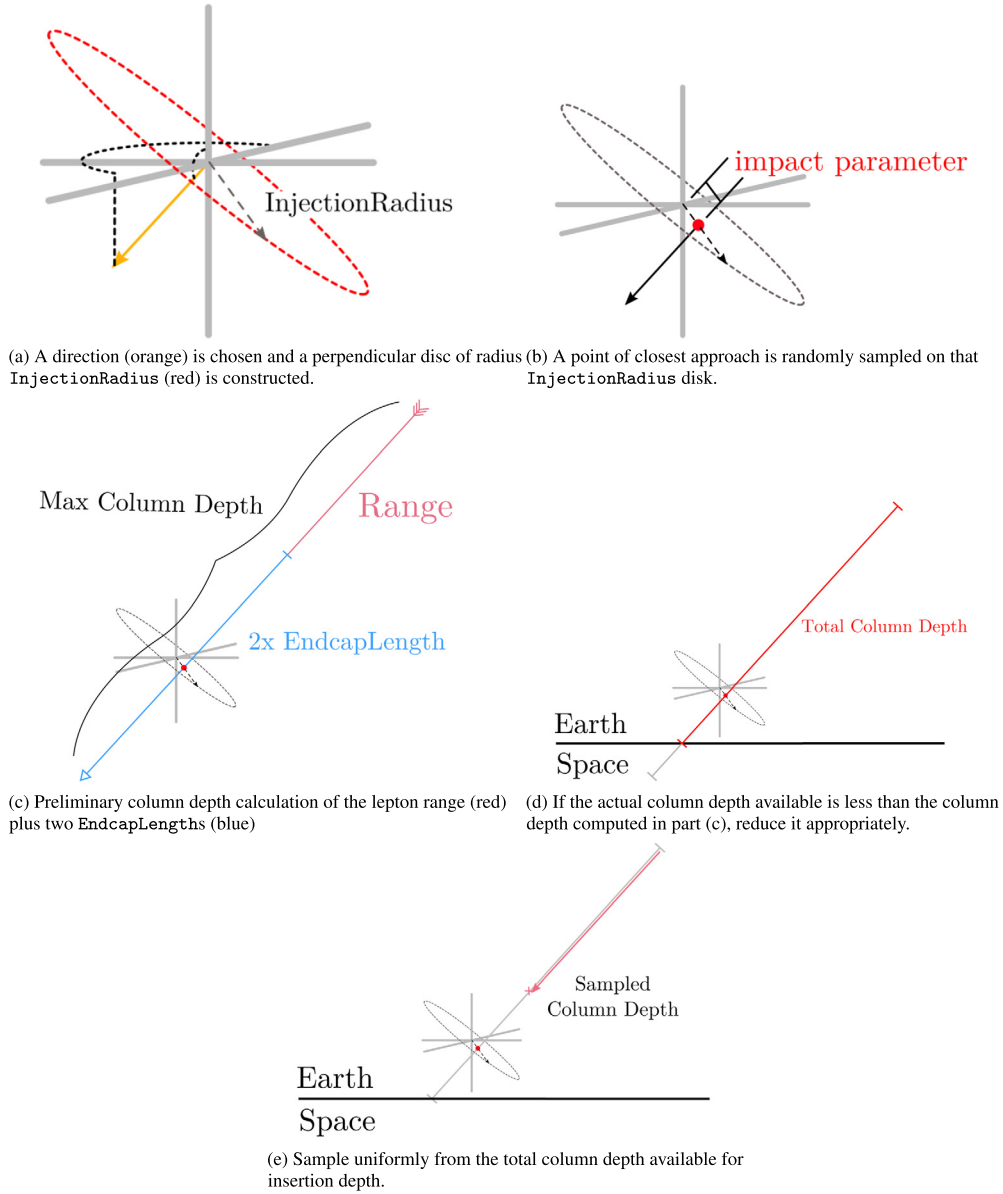


Fig. 2.3. Visualization of the ranged injection process and geometry.

in Fig. 2.4b. Additional comparisons were carried out for distributions of Bjorken x and y , and the opening angle between injected particles, and were all found to be in agreement.

`LeptonInjector` was also verified to produce events with a realistic distribution of final state kinematics. Among other tests, to this end, we compared a large sample of generated events with the theoretical predictions of the final states resulting from CC, NC, and GR interactions, for which we use [48] to model the DIS interactions and the analytical expressions given in [4,10] for the GR. Fig. 2.4a compares the average inelasticity for an interaction of a given energy for neutral- and charged-current interactions with both neutrinos and anti-neutrinos; results from a five-year IceCube study on inelasticity distributions are overlaid [49] along with a flux-averaged `LeptonInjector` sample. This trend closely follows the predictions of [48]; see [50] for an extended discussion.

3. LeptonWeighter

The events produced by the `LeptonInjector` algorithm described in Section 2 are generated at an arbitrary rate chosen by

the user; `LeptonWeighter` allows these events to then be re-weighted to any physical neutrino flux and interaction cross section. Here, we briefly explain the reweighting procedure.

Suppose a sample of events was generated according to some ansatz distribution $\Phi(E)$, e.g. according to

$$\frac{dN}{dE} = \Phi(E), \quad (3.1)$$

where E is the neutrino energy and dN/dE is the expected flux density. To re-weight the sample to a uniform distribution in energy, a weight w_{event} is applied to each event, inversely proportional to the generating probability density:

$$w_{\text{event}}(E_0) = 1/\Phi(E_0), \quad (3.2)$$

where E_0 is the energy of the event. Suppose instead two sub-samples were generated from distributions Φ_a and Φ_b , with the same domain in energy, and were then combined into one. To re-weight events in the combined sample to a uniform distribution, a weight of

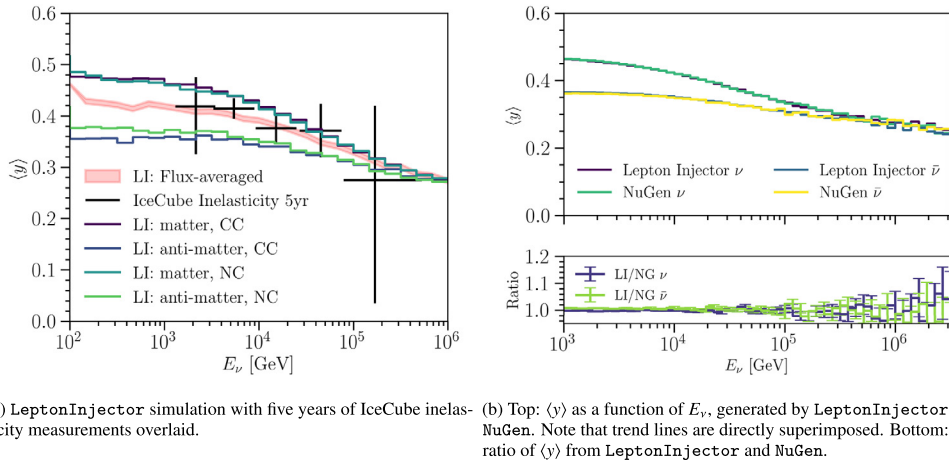


Fig. 2.4. Average event inelasticity, with respect to neutrino energy in GeV, for NC and CC events resulting from neutrinos and anti-neutrinos.

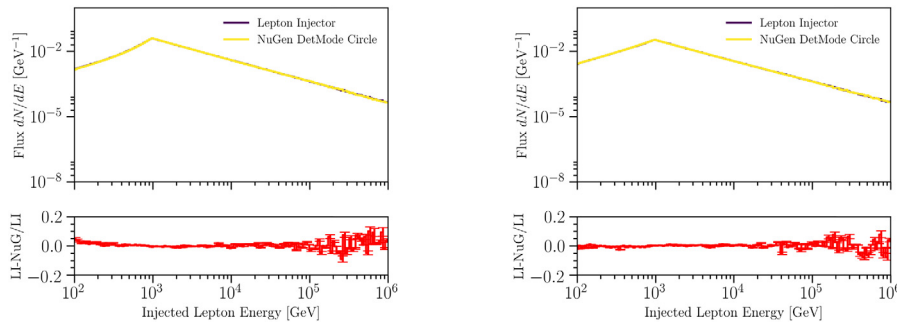


Fig. 2.5. Comparison between **LeptonInjector** and **NuGen**'s Detector Mode for the spectrum of the energies of leptons produced in the interactions. Total event energies were sampled with a 1 TeV minimum. Note that the **LeptonInjector** and **NuGen** lines are directly superimposed.

$$w_{\text{event}}(E_0) = \frac{1}{\Phi_a(E_0) + \Phi_b(E_0)} \quad (3.3)$$

would be applied. This is the generation weight, and accounts for the probability that either distribution could produce any given event. Events can then be re-weighted to a new distribution by evaluating their probability density in the new distribution and dividing by the generation weight.

Extending this to **LeptonInjector**, the probability density that a given **Generator** could have produced an event for each event is

$$p_{\text{MC}} = N_{\text{gen}} \frac{1}{\Omega_{\text{gen}} A_{\text{gen}}} \times \frac{\rho_{\text{gen}}(\ell)}{X_{\text{gen}}^{\text{col}}} \times \frac{1}{\sigma_{\text{tot}}} \frac{\partial^2 \sigma}{\partial x \partial y} \times \frac{\Phi(E)}{\int_{E_{\text{min}}}^{E_{\text{max}}} \Phi(E) dE} \quad (3.4)$$

where Ω_{gen} is the solid angle over which events were generated, $\Phi(E)$ is the power-law flux spectrum of the **Generator**, A_{gen} is the integrated area of the sampling surface, $\rho_{\text{gen}}(\ell)$ is the local mass density of targets, $X_{\text{gen}}^{\text{col}}$ is total column depth of targets in the sampling region, N_{gen} is the total number of generated events, and $\partial_{xy}\sigma$ and σ_{total} are the differential and total cross sections evaluated for the event, respectively. As a result, p_{MC} has units of $\text{sr}^{-1} \text{cm}^{-3} \text{GeV}^{-1}$. For a single MC generator, whose exact definitions are discussed in [51], the generation weight is the inverse of the generation probability density:

$$w_{\text{gen}} = \frac{1}{p_{\text{MC}}}. \quad (3.5)$$

In the regime of small neutrino interaction probability, an event's final weight, in units of s^{-1} , is approximately given by

$$w_{\text{event}} = \sum_{\{\text{gen}\}} \underbrace{\left(\frac{X_{\text{physical}}^{\text{col}} N_A}{M_{\text{target}}} \times \frac{\rho_{\text{physical}}(\ell)}{X_{\text{physical}}^{\text{col}}} \times \frac{\partial^2 \sigma}{\partial x \partial y} \times \Phi_{\text{physical}} \right)}_{\text{physical distribution}} \times \underbrace{w_{\text{gen}}}_{\text{gen weight}}, \quad (3.6)$$

where $\{\text{gen}\}$ indicates the set of generators, M_{target} is the molar mass of the target, Φ_{physical} is the desired physical flux of neutrinos at the detector, N_A is Avogadro's constant, $X_{\text{physical}}^{\text{col}}$ is calculated as

$$X_{\text{physical}}^{\text{col}} = \int_{\ell_i}^{\ell_f} \rho_{\text{physical}}(\ell) d\ell \quad (3.7)$$

along the path ℓ the particle would take to interact, and $\partial_{xy}\sigma$ is the differential cross section evaluated for the event. Note that $X_{\text{physical}}^{\text{col}}$ is the physical column density between the generation boundaries ℓ_f and ℓ_i , which are not necessarily the same across different generators. If the physical and generation density models are the same, then the $\rho_{\text{physical}}(\ell)/X_{\text{physical}}^{\text{col}}$ and $\rho_{\text{gen}}(\ell)/X_{\text{gen}}^{\text{col}}$ terms cancel. For a more complete description of the weighting that covers non-negligible interaction probabilities, see Appendix D.

This weighting calculation procedure is implemented in the **LeptonWeighter** C++ library and Python module; it is available

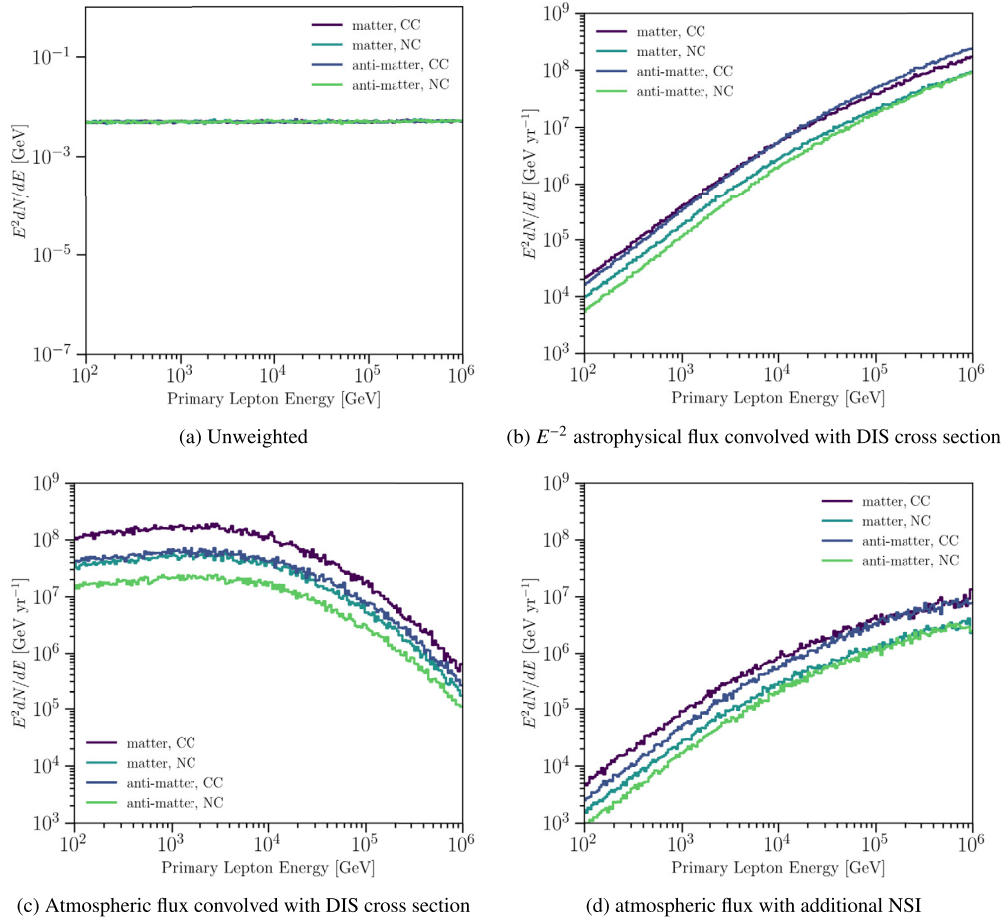


Fig. 3.1. Several re-weightings of the same sample with one year of live-time. Top-left: unweighted, top-right: re-weighted to an E^{-2} astrophysical flux, bottom-left: re-weighted to an atmospheric flux, bottom-right: re-weighted to a sample with non-standard interactions used propagation of atmospheric neutrinos. Here an extra, lepton-number violating, non-diagonal, term has been added to the neutrino propagation Hamiltonian.

in [31]. LeptonWeighter uses the LIC files generated by LeptonInjector to calculate the above generation weights, then a user-specified cross section and flux to calculate event weights.

3.1. Code structure

LeptonWeighter divides its functionality into distinct components: fluxes, cross sections, **Generators**, and **Weighters**. **Flux** objects are constructed to define a flux to which a sample should be weighted. **CrossSection** objects are similarly constructed with paths to locally saved FITS files of the same format as those used by LeptonInjector, and are used to define the cross sections used to weight the sample's events. LeptonWeighter constructs **Generators** by reading LIC files from disk and deserializing them. These **Generators** contain the exact simulation parameters used by LeptonInjector to generate events, and are able to calculate the generation weight for any event following a process illustrated in Fig. 3.2.

LeptonWeighter creates a **Weighter** object by using a list of **Generators**, a **Flux** object, and a **CrossSection** object. The **Weighter** object uses an event's properties, see Fig. A.1 in the Appendix, to calculate a weight as defined by Eq. (3.6). Fig. 3.1 demonstrates an all-flavor Monte Carlo sample composed of equal parts neutral- and charged-current DIS events, generated using an E^{-2} spectrum, and re-weighted to multiple different fluxes. The top-left plot shows the unweighted sample; the top-right left plot shows the sample reweighted to an astrophysical flux and weighted to the CSMS calculation of the DIS cross section [48]; the bottom-

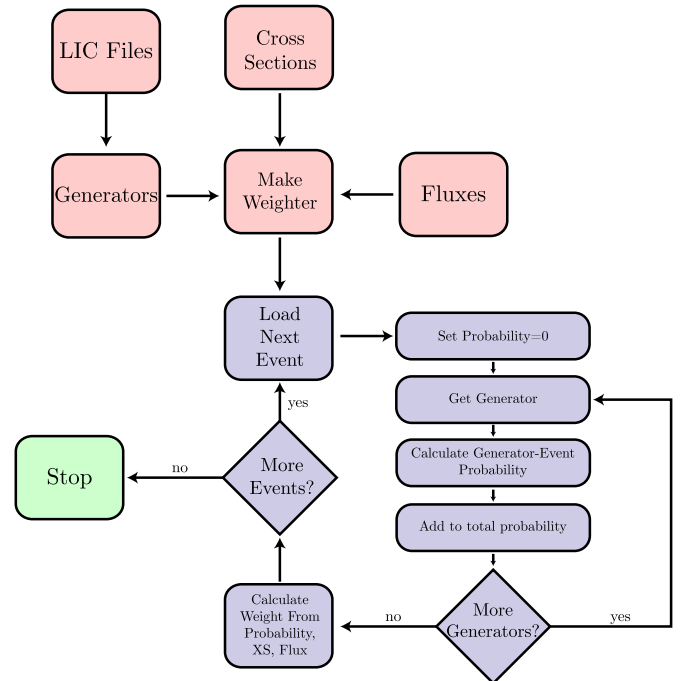


Fig. 3.2. A flowchart illustrating the process for calculating the individual weights of a collection of events.

left is re-weighted to an atmospheric neutrino flux and convolved with the same DIS cross sections [48]; the bottom-right plot shows the sample re-weighted to an atmospheric neutrino flux, with non-standard neutrino interaction (NSI) parameter strength of $\varepsilon_{\mu\tau} = 2 \times 10^{-1}$. See [52] for a precise definition of this parameter and nuSQuIDS [24,25] for the NSI implementation used.

In general, the weighting scheme allows to modify an already generated Monte Carlo set to any cross section that maps onto the same final states. For example, the DIS cross section could be a perturbative QCD calculation such as the CSMS@NLO [48] or the BGR@NNLO calculation given in [53] or a phenomenological estimate using the color-dipole model [25]. Similarly for the Glashow process one could use the original tree-level calculation in [4] or the updated calculation including radiative corrections given in [54].

4. Broader applications

Although `LeptonInjector` and `LeptonWeighter` have principally been developed for use by IceCube, the injection and weighting techniques are broadly applicable for experiments that need to simulate natural sources of neutrinos above 10 GeV. To adapt the software to other experiments we must account for differences in detector geometry and material composition within and around the detector.

The size of the injection region can be easily adjusted by changing the *InjectionRadius*, *EndcapLength*, *CylinderRadius*, and *CylinderHeight* parameters to encompass the extent of the detector. As long as the detector occupies a significant fraction of this cylindrical volume, the event injection remains efficient.

The material model used by `LeptonInjector` and `LeptonWeighter` has a simple implementation that models the Earth as a series of spherical shells with a radially varying polynomial density distribution. The polar ice cap is modeled as an offset spherical shell, also with a radially varying polynomial density. This implementation works well for detectors embedded in media that conform to spherical symmetry such as KM3NeT [16] in the Mediterranean Sea. However, this approach to the material model breaks down when the symmetry is broken, as is the case for GVD in Lake Baikal [17] and the 17 kt liquid Argon modules planned for DUNE [55]. To accommodate these experiments a more detailed software model of the surrounding material would need to be implemented. As long as the injection and weighting procedures query the updated material mode, no other modifications should be necessary.

5. Conclusions

Here we have presented the first publicly available neutrino telescope event generator for GeV-PeV+ energy ranges that factorizes the problem of Earth and atmospheric propagation from the event generation. The valid energy range of the generator is not limited by the software, but by the input cross sections provided by the user. The default CSMS cross section [48] provided with the code has less than 5% uncertainty in the 100 GeV to 100 EeV energy range. The factorization allows for streamlined and efficient production of neutrino events. `LeptonInjector`, along with its sister software `LeptonWeighter`, satisfy the needs of generating events for gigaton-scale neutrino observatories. The current implementation contains the most significant processes relevant to current analyses performed by these observatories, however we expect that this work will be extended as new calculations are made available and newer experimental needs arise. To this end, the code discussed in this paper follows an open-source model. Improvements recently proposed in the literature include: adding

sub-leading neutrino interactions such as interactions with the nuclear coulomb field [5–8], which is expected to be a 10% contribution at 1 PeV; use of updated DIS models such as those given in [29]; inclusion of trident neutrino events [56–59,8]; inclusion of nuclear effects on interactions [53]; inclusion of new physics processes such as production of heavy-neutral leptons [60–62] or dark neutrinos [63–67]; inclusion of new neutrino interactions mediated by light Z-prime [68–70]; among others.

Declaration of competing interest

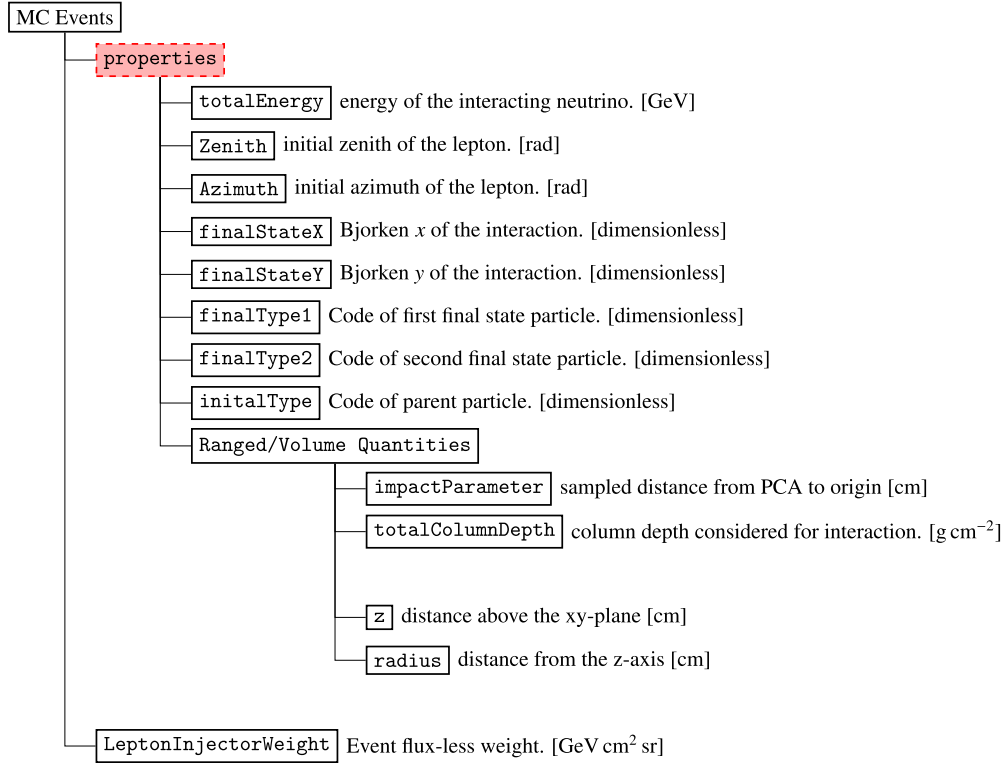
The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The IceCube collaboration acknowledges the significant contributions to this manuscript from Carlos Argüelles, Austin Schneider, and Benjamin Smithers. We acknowledge the support from the following agencies: USA – U.S. National Science Foundation-Office of Polar Programs, U.S. National Science Foundation-Physics Division, Wisconsin Alumni Research Foundation, Center for High Throughput Computing (CHTC) at the University of Wisconsin-Madison, Open Science Grid (OSG), Extreme Science and Engineering Discovery Environment (XSEDE), Frontera computing project at the Texas Advanced Computing Center, U.S. Department of Energy-National Energy Research Scientific Computing Center, Particle astrophysics research computing center at the University of Maryland, Institute for Cyber-Enabled Research at Michigan State University, and Astroparticle physics computational facility at Marquette University; Belgium – Funds for Scientific Research (FRS-FNRS and FWO), FWO Odysseus and Big Science programmes, and Belgian Federal Science Policy Office (Belspo); Germany – Bundesministerium für Bildung und Forschung (BMBF), Deutsche Forschungsgemeinschaft (DFG), Helmholtz Alliance for Astroparticle Physics (HAP), Initiative and Networking Fund of the Helmholtz Association, Deutsches Elektronen Synchrotron (DESY), and High Performance Computing cluster of the RWTH Aachen; Sweden – Swedish Research Council, Swedish Polar Research Secretariat, Swedish National Infrastructure for Computing (SNIC), and Knut and Alice Wallenberg Foundation; Australia – Australian Research Council; Canada – Natural Sciences and Engineering Research Council of Canada, Calcul Québec, Compute Ontario, Canada Foundation for Innovation, WestGrid, and Compute Canada; Denmark – Villum Fonden, Danish National Research Foundation (DNRF), Carlsberg Foundation; New Zealand – Marsden Fund; Japan – Japan Society for Promotion of Science (JSPS) and Institute for Global Prominent Research (IGPR) of Chiba University; Korea – National Research Foundation of Korea (NRF); Switzerland – Swiss National Science Foundation (SNSF); United Kingdom – Department of Physics, University of Oxford. United Kingdom – Science and Technology Facilities Council (STFC), part of UK Research and Innovation.

Appendix A. `LeptonInjector` event structure

All `LeptonInjector` events from a single process are saved to a single HDF5 file. Each **Injector** used in the generation process is given its own dataset inside the HDF5 file with four lists containing an entry for each event generated. Two lists are stored for the two final state particles' parameters, a third list contains the initial state particles' parameters, and a fourth list contains overall parameters for the events. Each of the event-entries in each of the three lists of particles contain, in order: a number differentiating whether it is initial or final state, the particles' PDG ID [71], the particles' positions, the particles' directions in radians,



Suppl. Figure A.1. Lepton Injector Monte Carlo event structure.

and the particles' energies. The overall properties stored for each event are shown in Fig. A.1. The impact parameter and total column depth are defined in Section 2.1 and shown graphically in Figs. 2.3b and 2.3d.

Appendix B. LIC file structure

Data serialized in the LIC file is written little-endian, regardless of machine architecture. When a LIC file is first opened, the **Controller** either overwrites any existing file with the same destination name or begins appending to the end of such an existing file. This behavior follows according to user-specification. If a new file is being written or an existing one overwritten, a block is first written to the file enumerating all **LeptonInjector** particle types. A header is first written specifying the size of the block, the name of the enumeration, and the length of the enumeration. Then the name and number of each entry in the particle enumeration is written.

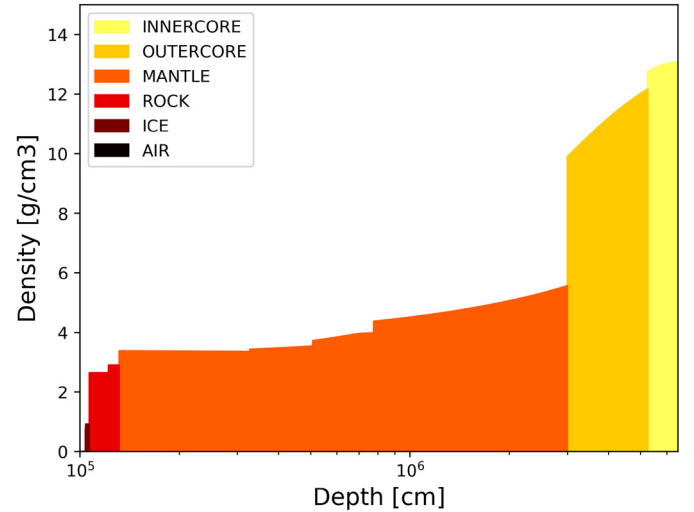
Afterwards, each time a new **Generator** is prepared, the **Controller** writes a new block to the LIC file. Each of these blocks is prefaced with a header specifying the size of the block, the name of the block, and the version of the **LeptonInjector** serialization code used to write the block. Then, all relevant generation parameters are written to the block.

Appendix C. Earth model density

LeptonInjector uses a modified Preliminary Reference Earth Model (PREM) for column depth calculations around the injection region. The density profile of which is shown in Fig. C.1.

Appendix D. Weighting

The generation procedure produces a set of neutrino properties that include the position, direction, energy, neutrino type, interac-

Suppl. Figure C.1. The density of the **LeptonInjector** Earth model as a function of depth from the edge of Earth's atmosphere.

tion type, and final state kinematic properties. The distribution of these properties at generation may differ from those we would expect in a physical scenario, so the weighting procedure is designed to correct for these differences. Beyond the distribution differences, weighting also corrects for differences in dimensionality and raw numbers of events. In our prototypical scenario, the weight of an event is dimensionless so only a correction factor to the total number of events is needed ($N_{\text{physical}}/N_{\text{gen}}$).

In the case of ranged injection we can separate the generation probability density of an event into several independent components

$$p_{\text{gen}} = p_{\text{gen}}^{\text{neutrino type}} \times p_{\text{gen}}^{\text{interaction type}} \times p_{\text{gen}}^{\text{energy}} \times p_{\text{gen}}^{\text{direction}}$$

$$\times p_{\text{gen}}^{\text{impact}} \times p_{\text{gen}}^{\text{depth}} \times p_{\text{gen}}^{\text{kinematics}}. \quad (\text{D.1})$$

The generators in `LeptonInjector` only deal with one neutrino and interaction type each, so $p_{\text{gen}}^{\text{neutrino type}}$ and $p_{\text{gen}}^{\text{interaction type}}$ will either be one or zero depending on if the event matches what can be produced by the generator. Similarly, $p_{\text{gen}}^{\text{energy}}$ is the probability distribution of injected neutrino energies which is zero for events with neutrino energies outside the bounds of the generator. Directions are distributed uniformly in ranged injection and so $p_{\text{gen}}^{\text{direction}} = 1/\Omega_{\text{gen}}$ where Ω_{gen} is the total solid angle available to the generator. The $p_{\text{gen}}^{\text{impact}}$ term is the probability distribution related to the impact parameter and angle; since events are sampled uniformly on a disk, this term is the inverse of the disk area, $p_{\text{gen}}^{\text{impact}} = 1/A_{\text{gen}}$, for events intersecting the disk and zero otherwise. Events are sampled uniformly with respect to column depth along the considered line segment, so the positional distribution of events can be described as $p_{\text{gen}}^{\text{depth}} = \rho_{\text{gen}}(\ell)/X_{\text{gen}}^{\text{col}}$, where $\rho_{\text{gen}}(\ell)$ is the local target mass density where the event is injected and $X_{\text{gen}}^{\text{col}}$ is the total column depth of targets along the considered line segment. Finally, $p_{\text{gen}}^{\text{kinematics}}$ is the probability distribution of the events kinematic variables; for charged current and neutral current events this is $p_{\text{gen}}^{\text{kinematics}} = (\partial_{xy}\sigma_{\text{gen}}^{\text{tot},i})/(\sigma_{\text{gen}}^{\text{tot},i})$.

These terms in the generation probability must then be paired with their physical counterparts. Since our hypothesis can specify the number of neutrinos per type, the neutrino type can be neglected beyond this number correction and the $p_{\text{gen}}^{\text{neutrino type}}$ term from the generator. The energy, direction, and impact terms all have their counterpart in the neutrino flux Φ_{physical} , which specifies the physical neutrino distribution in energy, direction, area, and time. The flux, when paired with a detector livetime L_{physical} also specifies the total number of neutrinos N_{physical} by the relation

$$L_{\text{physical}} \times \Phi_{\text{physical}} = N_{\text{physical}} \times p_{\text{physical}}^{\text{neutrino type}} \times p_{\text{physical}}^{\text{energy}} \times p_{\text{physical}}^{\text{direction}} \times p_{\text{physical}}^{\text{impact}}. \quad (\text{D.2})$$

The remaining terms, $p_{\text{gen}}^{\text{interaction type}}$, $p_{\text{gen}}^{\text{depth}}$, and $p_{\text{gen}}^{\text{kinematics}}$, deal with the neutrino interaction itself, which requires special care. The generation process assumes that the neutrino interacts within a certain region and with a specific interaction type. In reality, neutrinos on a path to the detector are potentially subject to any of several different interactions, and may pass through the Earth entirely unimpeded. Thus, we need to account for the probability that the neutrino in the physical scenario would interact within the region considered by the generator $p_{\text{physical}}^{\text{interaction type}}$, the depth distribution of all neutrino interactions within that region $p_{\text{physical}}^{\text{depth}}$, the probability that a specific interaction occurs once the interaction point has been chosen $p_{\text{physical}}^{\text{interaction type}}$, and finally the kinematic distribution $p_{\text{physical}}^{\text{kinematics}}$. The former two terms, $p_{\text{physical}}^{\text{interaction type}}$ and $p_{\text{physical}}^{\text{depth}}$ depend explicitly on the line segment considered by the generator when choosing the neutrino interaction vertex. The interaction probability can be cast in terms of the “survival” probability, $p_{\text{physical}}^{\text{interaction}} = 1 - p_{\text{physical}}^{\text{survival}}$, the probability that the neutrino will pass through the region without interacting. The survival probability is given by

$$p_{\text{physical}}^{\text{survival}} = \exp\left(-\int_{\ell_i}^{\ell_f} d\ell \sum_{p,i} n_{\text{physical}}^p(\ell) \sigma_{\text{physical}}^{\text{tot},p,i}\right), \quad (\text{D.3})$$

where p iterates over the possible targets (usually nucleons and electrons), i iterates over interaction types, and $n_{\text{physical}}^p(\ell)$ is the

density of target p at a point ℓ along the considered line segment. Thus,

$$p_{\text{physical}}^{\text{interaction}} = 1 - \exp\left(-\int_{\ell_i}^{\ell_f} d\ell \sum_{p,i} n_{\text{physical}}^p(\ell) \sigma_{\text{physical}}^{\text{tot},p,i}\right). \quad (\text{D.4})$$

Another way of writing this is in terms of the total column depth for each target $X_{\text{physical}}^{\text{col},p}$, target molar mass M_p , and Avagadro's number N_A , such that

$$p_{\text{physical}}^{\text{interaction}} = 1 - \exp\left(-N_A \sum_{p,i} (X_{\text{physical}}^{\text{col},p}/M_p) \sigma_{\text{physical}}^{\text{tot},p,i}\right). \quad (\text{D.5})$$

The implementation in `LeptonWeighter` groups protons and neutrons together and assumes that the molar mass of nucleons is 1 g mol^{-1} . The depth distribution $p_{\text{physical}}^{\text{depth}}$ follows a similar form as the survival probability, but is normalized within the generation bounds such that

$$p_{\text{physical}}^{\text{depth}} = \exp\left(-\int_{\ell_i}^{\ell_f} d\ell \sum_{p,i} n_{\text{physical}}^p(\ell) \sigma_{\text{physical}}^{\text{tot},p,i}\right) / \int_{\ell_i}^{\ell_f} d\ell \exp\left(-\int_{\ell_i}^{\ell_f} d\ell \sum_{p,i} n_{\text{physical}}^p(\ell) \sigma_{\text{physical}}^{\text{tot},p,i}\right), \quad (\text{D.6})$$

which can similarly be recast in terms of the density or column depth. The last two terms, $p_{\text{physical}}^{\text{interaction type}}$ and $p_{\text{physical}}^{\text{kinematics}}$, depend only on the position and type of the interaction, and so are independent of the generator. Once we assume that an interaction occurs at a known location, the probability of a specific interaction $p_{\text{physical}}^{\text{interaction type}}$ is the ratio of total cross sections at that location $p_{\text{physical}}^{\text{interaction type}} = (\sigma_{\text{physical}}^{\text{tot},i})/(\sum_j \sigma_{\text{physical}}^{\text{tot},j})$ where the subscript j iterates over all possible interactions. For the chosen interaction type, the kinematic distribution is simply $p_{\text{physical}}^{\text{kinematics}} = (\partial_{xy}\sigma_{\text{physical}}^{\text{tot},i})/(\sigma_{\text{physical}}^{\text{tot},i})$ in the case of charged or neutral current interactions.

By pairing up the terms we can see all the effects that are accounted for in the event weight

$$w_{\text{MC}} = \frac{N_{\text{physical}}}{N_{\text{gen}}} p_{\text{physical}}^{\text{interaction}} \frac{p_{\text{physical}}^{\text{neutrino type}}}{p_{\text{gen}}^{\text{neutrino type}}} \frac{p_{\text{physical}}^{\text{interaction type}}}{p_{\text{gen}}^{\text{interaction type}}} \frac{p_{\text{physical}}^{\text{energy}}}{p_{\text{gen}}^{\text{energy}}} \times \frac{p_{\text{physical}}^{\text{direction}}}{p_{\text{gen}}^{\text{direction}}} \frac{p_{\text{physical}}^{\text{impact}}}{p_{\text{gen}}^{\text{impact}}} \frac{p_{\text{physical}}^{\text{depth}}}{p_{\text{gen}}^{\text{depth}}} \frac{p_{\text{physical}}^{\text{kinematics}}}{p_{\text{gen}}^{\text{kinematics}}}. \quad (\text{D.7})$$

Practical implementations of this replace some of the physical terms with the flux and livetime to obtain the event weight

$$w_{\text{MC}} = \frac{L_{\text{physical}} \Phi_{\text{physical}}}{N_{\text{gen}} p_{\text{gen}}^{\text{neutrino type}} p_{\text{gen}}^{\text{energy}} p_{\text{gen}}^{\text{direction}} p_{\text{gen}}^{\text{impact}}} \times p_{\text{physical}}^{\text{interaction}} \times \frac{p_{\text{physical}}^{\text{interaction type}}}{p_{\text{gen}}^{\text{interaction type}}} \times \frac{p_{\text{physical}}^{\text{depth}}}{p_{\text{gen}}^{\text{depth}}} \times \frac{p_{\text{physical}}^{\text{kinematics}}}{p_{\text{gen}}^{\text{kinematics}}}. \quad (\text{D.8})$$

When simulation is created using multiple generators we must consider the probability that a particular event may be generated in either generator, regardless of which generator it originated from. The behavior we desire is such that events from non-overlapping regions of the parameter space retain their original single-generator weights, but events that reside in the overlap

regions are down-weighted to account for the overlap. The event weight then takes the form

$$w_{MC} = \left[\sum_i \left(p_{\text{physical}}^i \right)^{-1} \times p_{\text{gen}}^i \right]^{-1}, \quad (\text{D.9})$$

where the superscript i denotes the different generators, p_{physical}^i is the physical contribution to the weighting, and p_{gen}^i is the generation contribution to the weighting. Note the superscript i on the physical contribution is there because the terms $p_{\text{physical}}^{\text{interaction}}$ and $p_{\text{physical}}^{\text{depth}}$ depend on the particular generator. If the generation settings governing the line segment along which the event is injected are common to all generators then this dependence can be dropped and p_{physical} can be factored out.

The above description in the weighting starts from the ranged injection procedure, but only minor modifications are needed for this to be applicable to volume injection. The difference arises from the $p_{\text{gen}}^{\text{depth}}$ and $p_{\text{gen}}^{\text{impact}}$ terms on the generation side and $p_{\text{physical}}^{\text{depth}}$ and $p_{\text{physical}}^{\text{interaction}}$ on the physical side. These generation terms are directly analogous to steps in the ranged injection procedure where the position of closest approach and interaction vertex position are chosen. In the volume injection, the interaction vertex is chosen in a single step, so we can replace these two generation terms with the single $p_{\text{gen}}^{\text{position}}$ term which is a uniform probability density along the line segment within the injection cylinder and zero outside. The physical terms only differ in that the line segment considered for the calculation now must come from the volume injection procedure. Specifically, the line segment considered passes through the interaction vertex following the injected neutrino direction, beginning and ending at the boundaries of the injection cylinder.

These differences between the ranged and volume injection mean that the physical contributions to the weighting will differ between the two, and must be calculated separately for each event if both methods are used in the event generation.

Finally, the approximation used in Eq. (3.6) can be obtained by expanding the depth and interaction terms for a vanishing product of the interaction cross section and column depth. For the column depths and cross sections used in `LeptonInjector`, this approximation remains valid for sub ZeV neutrino energies.

Appendix E. Generation example

This Python example is included in the `LeptonInjector` source code. It creates an Injector in Ranged mode for producing CC muon-neutrino events of initial energy between 1000 GeV to 100 000 GeV, and outputs the data to an HDF5 file.

```
import LeptonInjector as LI
from math import pi
import os

xs_folder = os.path.join( os.path.dirname(__file__), '..' )

n_events = 1000
diff_xs = xs_folder + "/test_xs.fits"
total_xs = xs_folder + "/test_xs_total.fits"
# Ranged Mode, described in Section II.A
is_ranged = True
# Particles chosen for NuMu-CC using Table II.1
final_1 = LI.Particle.MuMinus
final_2 = LI.Particle.Hadrons
# Build the Injector object described in Section 2
the_injector = LI.injector( n_events , final_1 , final_2 , diff_xs ,
                           total_xs , is_ranged )

deg = pi/180.
minE = 1000. # [GeV]
```

```
maxE = 100000. # [GeV]
gamma = 2.
minZenith = 80.*deg
maxZenith = 180.*deg
minAzimuth = 0.*deg
maxAzimuth = 180.*deg
```

```
# constructing the Controller object described in Section II
controller = LI.Controller( the_injector , minE , maxE , gamma ,
                           minAzimuth , maxAzimuth , minZenith , maxZenith )
```

```
# specify the output
controller.Output("./data_output.h5")
controller.LICFile("./config.lic")
```

```
# Starts the Process, as illustrated in Figure II.1
controller.Execute()
```

Appendix F. Weighting example

The following Python example is included in `LeptonWeighter`. It reads a set of generated events and computes the weights of for a given neutrino cross sections and fluxes. The result is stored in an HDF5 file for later usage.

```
import LeptonWeighter as LW
import h5py as h5
import numpy as np

"""
This calculates the weight of each event in the LeptonInjector
example script.
"""

# These objects are all defined in Section III.A

# Create generator
# if there were multiple LIC files , you would instead make a
# list of Generators
net_generation = [LW.MakeGeneratorsFromLICFile("config.lic")]

# This cross section object takes four differential cross
# sections (dS/dEdxdy)
# Neutrino CC-DIS xs
# Anti-Neutrino CC-DIS xs
# Neutrino NC-DIS xs
# Anti-Neutrino NC-DIS xs
cross_section_location = "/path/to/cross_sections/"
xs = LW.CrossSectionFromSpline(
    cross_section_location+"/dsdxdy_nu_CC_iso.
fits",
    cross_section_location+"/dsdxdy_nubar_CC_iso.
fits",
    cross_section_location+"/dsdxdy_nu_NC_iso.
fits",
    cross_section_location+"/dsdxdy_nubar_NC_iso.
fits")

# GeV unitless GeV
flux_params={ 'constant': 10**-18, 'index':-2, 'scale':10**5 }
liveTime = 3.1536e7 # [s]

flux = LW.PowerLawFlux( flux_params['constant'] , flux_params['
index'] , flux_params['scale'] )

# build weighter
weight_event = LW.Weighter( flux , xs , net_generation )

def get_weight( props ):
    """
    This function takes the "properties" object from a LI-Event.
    It then calculates and returns the weight
    """
    LWevent = LW.Event()
    LWevent.energy = props[0]
    LWevent.zenith = props[1]
    LWevent.azimuth = props[2]
```

```

LWevent.interaction_x = props[3]
LWevent.interaction_y = props[4]
LWevent.final_state_particle_0 = LW.ParticleType( props[5] )
LWevent.final_state_particle_1 = LW.ParticleType( props[6] )
LWevent.primary_type = LW.ParticleType( props[7] )
LWevent.radius = props[8]
LWevent.total_column_depth = props[9]
LWevent.x = 0
LWevent.y = 0
LWevent.z = 0

weight = weight_event(LWevent)

# this would alert us that something bad is happening
if weight==np.nan:
    raise ValueError("Bad Weight!")

return( weight*time )

data_file = h5.File("data_output.h5")
injector_list = data_file.keys() # Each injector is treated as an
    entry in a dictionary
print("Loaded {} Injectors: {}".format(len(injector_list),
    injector_list))

for injector in data_file.keys():
    print("Evaluating {}".format(injector))
    for event in range(len( data_file[injector][ 'properties' ] )):
        print("      Event {0:06d} Weight: {}".format(event,
            get_weight( data_file[injector][ 'properties' ][event] ) )

data_file.close()

```

References

- [1] E. Vitagliano, I. Tamborra, G. Raffelt, Grand unified neutrino spectrum at Earth, arXiv:1910.11878, 2019.
- [2] J.A. Formaggio, G.P. Zeller, Rev. Mod. Phys. 84 (2012) 1307–1341, <https://doi.org/10.1103/RevModPhys.84.1307>, arXiv:1305.7513.
- [3] D. Akimov, et al., Science 357 (6356) (2017) 1123–1126, <https://doi.org/10.1126/science.aao0990>, <https://science.sciencemag.org/content/357/6356/1123.full.pdf>, <https://science.sciencemag.org/content/357/6356/1123>.
- [4] S.L. Glashow, Phys. Rev. 118 (1960) 316–317, <https://doi.org/10.1103/PhysRev.118.316>.
- [5] D. Seckel, Phys. Rev. Lett. 80 (1998) 900–903, <https://doi.org/10.1103/PhysRevLett.80.900>, arXiv:hep-ph/9709290.
- [6] I. Alikhanov, Phys. Lett. B 741 (2015) 295–300, <https://doi.org/10.1016/j.physletb.2014.12.056>, arXiv:1402.6678.
- [7] B. Zhou, J. Beacom, Phys. Rev. D 101 (02) (2020), <https://doi.org/10.1103/PhysRevD.101.036011>.
- [8] B. Zhou, J.F. Beacom, Phys. Rev. D 101 (3) (2020) 036010, <https://doi.org/10.1103/PhysRevD.101.036010>, arXiv:1910.10720.
- [9] V. Barger, E. Basso, Y. Gao, W.-Y. Keung, Phys. Rev. D 95 (9) (2017) 093002, <https://doi.org/10.1103/PhysRevD.95.093002>, arXiv:1611.00773.
- [10] R. Gandhi, C. Quigg, M.H. Reno, I. Sarcevic, Astropart. Phys. 5 (1996) 81–110, [https://doi.org/10.1016/0927-6505\(96\)00008-4](https://doi.org/10.1016/0927-6505(96)00008-4), arXiv:hep-ph/9512364.
- [11] Y. Hayato, Nucl. Phys. Proc. Suppl. 112 (2002) 171–176, [https://doi.org/10.1016/S0920-5632\(02\)01759-0](https://doi.org/10.1016/S0920-5632(02)01759-0).
- [12] C. Andreopoulos, et al., Nucl. Instrum. Methods A 614 (2010) 87–104, <https://doi.org/10.1016/j.nima.2009.12.009>, arXiv:0905.2517.
- [13] T. Golan, J.T. Sobczyk, J. Zmuda, Nucl. Phys. Proc. Suppl. 229–232 (2012) 499, <https://doi.org/10.1016/j.nuclphysbps.2012.09.136>.
- [14] O. Lalakulich, K. Gallmeister, U. Mosel, J. Phys. Conf. Ser. 408 (012053) (2013), <https://doi.org/10.1088/1742-6596/408/1/012053>, arXiv:1110.0674, 2011.
- [15] M.G. Aartsen, et al., J. Instrum. 12 (03) (2017) P03012, <https://doi.org/10.1088/1748-0221/12/03/P03012>, arXiv:1612.05093.
- [16] S. Adrian-Martinez, et al., J. Phys. G 43 (8) (2016) 084001, <https://doi.org/10.1088/0954-3889/43/8/084001>, arXiv:1601.07459.
- [17] A.D. Avrorin, et al., EPJ Web Conf. 191 (2018) 01006, <https://doi.org/10.1051/epjconf/201819101006>, arXiv:1808.10353.
- [18] A. Roberts, in: DUMAND - Deep Underwater Muon and Neutrino Detection 1978 Summer Workshop, Session 2: Ultra High Energy Interactions and Astrophysical Neutrino Sources, 1978.
- [19] G.C. Hill, Experimental and theoretical aspects of high energy neutrino astrophysics, Ph.D. thesis, Adelaide U., 9, 1996.
- [20] A. Gazizov, M.P. Kowalski, Comput. Phys. Commun. 172 (2005) 203–213, <https://doi.org/10.1016/j.cpc.2005.03.113>, arXiv:astro-ph/0406439.
- [21] S. Yoshida, R. Ishibashi, H. Miyamoto, Phys. Rev. D 69 (2004) 103004, <https://doi.org/10.1103/PhysRevD.69.103004>, arXiv:astro-ph/0312078.
- [22] D.J. Bailey, Monte Carlo tools and analysis methods for understanding the ANTARES experiment and predicting its sensitivity to Dark Matter, Ph.D. thesis, Wolfson College, 2002.
- [23] T.R. De Young, IceTray: a software framework for IceCube, <https://doi.org/10.5170/CERN-2005-002.463>, <http://cds.cern.ch/record/865626>, 2005.
- [24] C.A. Argüelles, J. Salvado, C.N. Weaver, Comput. Phys. Commun. 255 (2020) 107405, <https://doi.org/10.1016/j.cpc.2020.107405>.
- [25] C.A. Argüelles, J. Salvado, C.N. Weaver nuSQLDS, <https://github.com/Arguelles/nuSQLDS>, 2015.
- [26] E. Zas, in: PoS ICRC2017, 2018, p. 972.
- [27] A.C. Vincent, C.A. Argüelles, A. Kheirandish, J. Cosmol. Astropart. Phys. 1711 (11) (2017) 012, <https://doi.org/10.1088/1475-7516/2017/11/012>, arXiv:1706.09895.
- [28] I. Safa, A. Pizzuto, C.A. Argüelles, F. Halzen, R. Hussain, A. Kheirandish, J. Vandenbroucke, J. Cosmol. Astropart. Phys. 2001 (01) (2020) 012, <https://doi.org/10.1088/1475-7516/2020/01/012>, arXiv:1909.10487.
- [29] A. Garcia, R. Gauld, A. Heijboer, J. Rojo, J. Cosmol. Astropart. Phys. 2020 (09) (2020) 025, <https://doi.org/10.1088/1475-7516/2020/09/025>.
- [30] IceCube, LeptonInjector code, <https://github.com/icecube/LeptonInjector>, 2020.
- [31] IceCube, LeptonWeighter code, <https://github.com/icecube/LeptonWeighter>, 2020.
- [32] M.G. Aartsen, et al., Phys. Rev. D 98 (6) (2018) 062003, <https://doi.org/10.1103/PhysRevD.98.062003>, arXiv:1807.01820.
- [33] S. Yoshida, M. Meier, ShigeruYoshida/JULIET: first official release with a DOI, <https://doi.org/10.5281/zenodo.4018117>, Sep. 2020.
- [34] N. Whitehorn, J. van Santen, S. Lafebre, Comput. Phys. Commun. 184 (9) (2013) 2214–2220, <https://doi.org/10.1016/j.cpc.2013.04.008>, <http://www.sciencedirect.com/science/article/pii/S0010465513001434>.
- [35] J. Koehne, K. Frantzen, M. Schmitz, T. Fuchs, W. Rhode, D. Chirkin, J. Becker, Comput. Phys. Commun. 184 (2013) 2070–2090, <https://doi.org/10.1016/j.cpc.2013.04.001>.
- [36] D. Chirkin, W. Rhode, Muon Monte Carlo: a high-precision tool for muon propagation through matter, arXiv:hep-ph/0407075, 7, 2004.
- [37] V. Niess, V. Bertin, Astropart. Phys. 26 (4–5) (2006) 243–256, <https://doi.org/10.1016/j.astropartphys.2006.06.005>.
- [38] C.H.V. Wiebusch, The Detection of Faint Light in Deep Underwater Neutrino Telescopes, Chapter 7, Ph.D. thesis, Physikalische Institute RWTH Aachen, 12, 1995.
- [39] L. Rädcl, C. Wiebusch, Astropart. Phys. 38 (2012) 53–67, <https://doi.org/10.1016/j.astropartphys.2012.09.008>, arXiv:1206.5530.
- [40] L. Rädcl, C. Wiebusch, Astropart. Phys. 44 (2013) 102–113, <https://doi.org/10.1016/j.astropartphys.2013.01.015>, arXiv:1210.5140.
- [41] D. Chirkin, J.C. Díaz-Vélez, C. Kopper, A. Olivas, B. Riedel, M. Rongen, D. Schultz, J. van Santen, in: 2019 15th International Conference on eScience, eScience, 2019, pp. 388–393.
- [42] C. Kopper, CLSim, <https://github.com/claudiok/clsim>, 2019.
- [43] D. Chirkin, in: GPU Computing in High-Energy Physics, 2015, pp. 217–220.
- [44] D. Chirkin, PPC standalone code, <https://icecube.wisc.edu/~dima/work/WISC/ppc/>, 2020.
- [45] W. Hastings, Biometrika 57 (1970) 97–109, <https://doi.org/10.1093/biomet/57.1.97>, <https://inspirehep.net/literature/1114360>.
- [46] D. Chirkin, W. Rhode, Propagating leptons through matter with muon Monte Carlo (mmc), arXiv:hep-ph/0407075, 2004.
- [47] A.M. Dziewonski, D.L. Anderson, Phys. Earth Planet. Inter. 25 (4) (1981) 297–356, [https://doi.org/10.1016/0031-9201\(81\)90046-7](https://doi.org/10.1016/0031-9201(81)90046-7), <http://www.sciencedirect.com/science/article/pii/0031920181900467>.
- [48] A. Cooper-Sarkar, P. Mertsch, S. Sarkar, J. High Energy Phys. 08 (2011) 042, [https://doi.org/10.1007/JHEP08\(2011\)042](https://doi.org/10.1007/JHEP08(2011)042), arXiv:1106.3723.
- [49] M.G. Aartsen, et al., Phys. Rev. D 99 (2019) 032004, <https://doi.org/10.1103/PhysRevD.99.032004>, <https://link.aps.org/doi/10.1103/PhysRevD.99.032004>.
- [50] G.A. Binder, Measurements of the Flavor Composition and Inelasticity Distribution of High-Energy Neutrino Interactions in IceCube, Ph.D. thesis, UC, Berkeley, 2017.
- [51] J.S. Gainer, J. Lykken, K.T. Matchev, S. Mrenna, M. Park, J. High Energy Phys. 10 (2014) 078, [https://doi.org/10.1007/JHEP10\(2014\)078](https://doi.org/10.1007/JHEP10(2014)078), arXiv:1404.7129.
- [52] P.S. Bhupal Dev, et al., SciPost Phys. Proc. 2 (2019) 001, <https://doi.org/10.21468/SciPostPhysProc.2.001>, arXiv:1907.00991.
- [53] V. Bertone, R. Gauld, J. Rojo, J. High Energy Phys. 01 (2019) 217, [https://doi.org/10.1007/JHEP01\(2019\)217](https://doi.org/10.1007/JHEP01(2019)217), arXiv:1808.02034.
- [54] R. Gauld, Phys. Rev. D 100 (9) (2019) 091301, <https://doi.org/10.1103/PhysRevD.100.091301>, arXiv:1905.03792.
- [55] B. Abi, et al., J. Instrum. 15 (08) (2020) T08010, <https://doi.org/10.1088/1748-0221/15/08/T08010>, arXiv:2002.03010.
- [56] W. Altmannshofer, S. Gori, M. Pospelov, I. Yavin, Phys. Rev. Lett. 113 (2014) 091801, <https://doi.org/10.1103/PhysRevLett.113.091801>, arXiv:1406.2332.
- [57] S.-F. Ge, M. Lindner, W. Rodejohann, Phys. Lett. B 772 (2017) 164–168, <https://doi.org/10.1016/j.physletb.2017.06.020>, arXiv:1702.02617.

- [58] P. Ballett, M. Hostert, S. Pascoli, Y.F. Perez-Gonzalez, Z. Tabrizi, R. Zukanovich Funchal, J. High Energy Phys. 01 (2019) 119, [https://doi.org/10.1007/JHEP01\(2019\)119](https://doi.org/10.1007/JHEP01(2019)119), arXiv:1807.10973.
- [59] W. Altmannshofer, S. Gori, J. Martín-Albo, A. Sousa, M. Wallbank, Phys. Rev. D 100 (11) (2019) 115029, <https://doi.org/10.1103/PhysRevD.100.115029>, arXiv:1902.06765.
- [60] P. Coloma, P.A.N. Machado, I. Martinez-Soler, I.M. Shoemaker, Phys. Rev. Lett. 119 (2017) 201804, <https://doi.org/10.1103/PhysRevLett.119.201804>, <https://link.aps.org/doi/10.1103/PhysRevLett.119.201804>.
- [61] G. Magill, R. Plestid, M. Pospelov, Y.-D. Tsai, Phys. Rev. D 98 (11) (2018) 115015, <https://doi.org/10.1103/PhysRevD.98.115015>, arXiv:1803.03262.
- [62] J.M. Cline, M. Puel, T. Toma, J. High Energy Phys. 05 (2020) 039, [https://doi.org/10.1007/JHEP05\(2020\)039](https://doi.org/10.1007/JHEP05(2020)039), arXiv:2001.11505.
- [63] E. Bertuzzo, S. Jana, P.A.N. Machado, R. Zukanovich Funchal, Phys. Rev. Lett. 121 (24) (2018) 241801, <https://doi.org/10.1103/PhysRevLett.121.241801>, arXiv:1807.09877.
- [64] M. Blennow, E. Fernandez-Martinez, A. Olivares-Del Campo, S. Pascoli, S. Rosauero-Alcaraz, A. Titov, Eur. Phys. J. C 79 (7) (2019) 555, <https://doi.org/10.1140/epjc/s10052-019-7060-5>, arXiv:1903.00006.
- [65] P. Ballett, M. Hostert, S. Pascoli, Phys. Rev. D 99 (9) (2019) 091701, <https://doi.org/10.1103/PhysRevD.99.091701>, arXiv:1903.07590.
- [66] P. Coloma, Eur. Phys. J. C 79 (9) (2019) 748, <https://doi.org/10.1140/epjc/s10052-019-7256-8>, arXiv:1906.02106.
- [67] A. Abdullahi, M. Hostert, S. Pascoli, A dark seesaw solution to low energy anomalies: MiniBooNE, the muon ($g-2$), and BaBar, arXiv:2007.11813, 7, 2020.
- [68] John F. Cherry, Alexander Friedland, Ian M. Shoemaker, Short-baseline neutrino oscillations, Planck, and IceCube, arXiv:1605.06506, 5, 2016.
- [69] P. Bakhti, Y. Farzan, M. Rajaei, Phys. Rev. D 99 (5) (2019) 055019, <https://doi.org/10.1103/PhysRevD.99.055019>, arXiv:1810.04441.
- [70] P. Ballett, M. Hostert, S. Pascoli, Y.F. Perez-Gonzalez, Z. Tabrizi, R. Zukanovich Funchal, Phys. Rev. D 100 (5) (2019) 055012, <https://doi.org/10.1103/PhysRevD.100.055012>, arXiv:1902.08579.
- [71] M. Tanabashi, et al., Phys. Rev. D 98 (2018) 030001, <https://doi.org/10.1103/PhysRevD.98.030001>, <https://link.aps.org/doi/10.1103/PhysRevD.98.030001>.