LA-UR-17-23432
Approved for public release; distribution is unlimited.

Title: Importance Sampling Variance Reduction in GRESS ATMOSIM

Author(s): Wakeford, Daniel Tyler

Intended for: Report

Issued: 2017-04-26

Disclaimer:
Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by the Los Alamos National Security, LLC for the National Nuclear Security Administration of the U.S. Department of Energy under contract DE-AC52-06NA25396. By approving this article, the publisher recognizes that the U.S. Government retains nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.

Importance Sampling Variance Reduction in GRESS ATMOSIM

Daniel T. Wakeford

April 21, 2017

This document is intended to introduce the importance sampling method of variance reduction to a Geant4 user for application to neutral particle Monte Carlo transport through the atmosphere, as implemented in GRESS ATMOSIM.

1 The problem

Monte Carlo simulation is limited by statistical accuracy. To achieve an accurate solution in all regions of phase space of a problem, a sufficient number of source particles must reach each region. This proves difficult in dense media which present a high macroscopic cross section along the particle flight path. Although the Earth's atmosphere is not locally dense, particles must travel many mean free paths to fully traverse it. In this way the atmospheric transport problem shares a similar difficulty with deep-shielding problems. One method of solution to deep-shielding problems with Monte Carlo is to employ variance reduction methods. These methods improve the convergence time of the simulation and can take various forms. This document focuses only on the method known as importance sampling (also called splitting).

The General Response Simulation System (GRESS) [1] software includes a package called ATMOSIM that is used for simulation of particle transport in the atmosphere. ATMOSIM is an application built using the Geant4 toolkit [2] and consisting of a model of the Earth's atmosphere (based on the NRL MSISE00 model [3]) and various radiation source terms and analysis scripts. ATMOSIM has been used to evaluate the flux of neutrons and photons as a function of altitudes, both within and above the atmosphere. The challenge is that at low altitudes (<40 km) the atmosphere presents many interaction mean free paths to an escaping particle flux. The atmospheric attenuation is problematic even along the outward radial path at low altitude, and severely attenuating at high angles off normal and at low energies. To obtain a reasonably accurate estimate of particle flux under these conditions, many weeks and months of computational simulation may be required.

Owing to these issues, neutron transport through the atmosphere at long distances requires variance reduction (VR) techniques to achieve better computational efficiency. The efficiency of the computation is strongly dependent on the choice and configuration of VR technique. With VR methods, the goal is to avoid wasting time tracking particles that never make it to the tally region. Put another way, we want to extract some useful tally information from every source particle, without biasing the end result. Since every particle provides useful physical information it is possible to reduce the variance of the tally without increasing computational time. This can be accomplished with importance sampling as described in this document. An excellent general review of variance reduction considerations and implementation in Monte Carlo is given in [4].

2 Importance sampling in ATMOSIM

The goal of Monte Carlo variance reduction is to reduce the computational time required to obtain results to a desired accuracy. The reduction in time equates to an improvement in simulation efficiency, where efficiency can be expressed as:

$$\epsilon = \frac{1}{\sigma^2 t} \tag{1}$$

where σ^2 is the sample variance (the square to the standard deviation of the mean of the quantity of interest) and t is the real world computational time (in minutes or hours). If n is the number of events thrown, then $\sigma^2 \propto n^{-1}$ and $t \propto n$, and so for a purely analog simulation the efficiency is a constant factor regardless of the simulation time. With importance sampling, n is altered by creating and destroying particles is a controlled way at all regions of the model. By altering n and re-weighting the track with a bookkeeping weight w_n, σ^2 can be reduced without increasing t. Essentially this translates to creating additional particles at lower weight in difficult to reach (more important) regions. To keep the simulation unbiased, particles are also destroyed with equivalent probability (roulette) if they return to regions of lower importance. Geant4 includes an importance algorithm that the user can apply to a model [5]. The results herein use the provided Geant4 importance algorithm in `G4ImportanceAlgorithm.cc`, unmodified.

The Geant4 algorithm is illustrated in Figure 1. The source is at the center of the innermost ring and the tally region is the outer surface of the outermost ring. Importance boundaries are represented by dotted rings. At the source, a single particle is emitted. In the source region the importance value is set to 1 and the particle weight begins as 1. As the particle crosses an importance boundary from cell i to cell $i+1$ the following calculation is made:

$$\begin{aligned}
&\text{if } \frac{I_i}{I_{i+1}} = 1, \text{ continue track without modification,}\\
&\text{if } \frac{I_i}{I_{i+1}} < 1, \text{ split into} \frac{I_{i+1}}{I_i} \text{identical tracks, each with weight} \frac{I_i}{I_{i+1}}\\
&\text{if } \frac{I_i}{I_{i+1}} > 1, \text{ roulette, kill particle with probability } 1 - \frac{I_{i+1}}{I_i}, \text{ set weight to i+1 cell weight}
\end{aligned} \tag{2}$$

The calculation is blind to the current weight of the track, and all tracks in a particular cell will in fact have the same weight with this technique. In Figure 1, the importance ratio between neighboring cells is 2, which leads to duplicating tracks at boundaries of increasing importance, and killing half the tracks that cross boundaries of decreasing importance.

2.1 Implementation code

The basic importance algorithm described above is included with Geant4 but it is not included in application code by default (as it is in MCNP). ATMOSIM includes a user implementation of importance sampling which enables a user to get started with the method. Tuning the method involves setting appropriate importance cell dimensions and importance values, and is not an automated process. The method is implemented in a class called `atmoParallelConstruction`, and called by the main application function `atmosim` as shown in Listing 1.

In Listing 1 a class atmoParallelConstruction returns a defined World Volume that includes the importance cell geometries and specified importances. Users can refer to

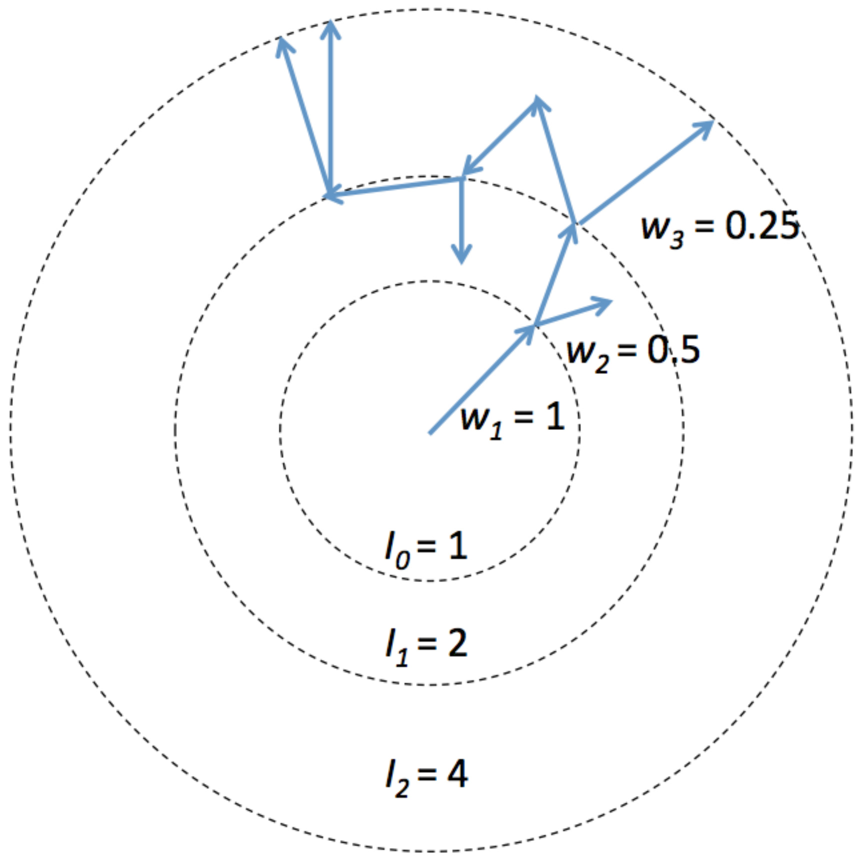

Figure 1: An importance sampled geometry showing a number of track histories resultant from a single primary event. The importances per cell are denoted by I_i and the track weights by w_j.

Listing 1: atmosim.cpp (main) implementation

```
...
  G4String parallelName("parallelWorld");
  //generate a new parallel world for sampling
  atmoParallelConstruction* theParallelConstruction =
      new atmoParallelConstruction(parallelName);
  G4GeometrySampler pgs(theParallelConstruction->GetWorldVolume(),"neutron");

  if (theatmoCommand->UseVR())  {
    theDetectorConstruction->RegisterParallelWorld(theParallelConstruction);
    G4cout<<"Setting up importance sampling"<<G4endl;
    pgs.SetParallel(true);
    physicsList->RegisterPhysics(new G4ImportanceBiasing(&pgs,parallelName));
    physicsList->RegisterPhysics(new G4ParallelWorldPhysics(parallelName));

  }

...

  std::ifstream macfile(argv[1]);
  std::string macstr;
  while (std::getline(macfile, macstr)) {
    UI->ApplyCommand(macstr);
    if (macstr == "/run/initialize") {
      if (theatmoCommand->UseVR())  {
        G4cout<<"Building Importance Store..."<<G4endl;
        theParallelConstruction->CreateImportanceStore();
      }
    }
  }

...

  if (theatmoCommand->UseVR())  {
    pgs.ClearSampling();
  }
...
```

`src/atmoParallelConstruction.cpp` for an example of constructing concentric sphere importance shells of variable radius and importance value. Importance sampling parameters (cell size and importance) are problem-specific, and will differ based on event altitude and source energy. The user will have to verify that the importance cells are correctly constructed for their simulation.

Steps for running an ATMOSIM simulation with importance sampling:

1. Ensure the macro flag is set: `/atmo/vr/ enabled`
2. Analog run:
   (a) Are the cells set up properly and centered properly in the parallel world?
   (b) Are the importances set properly (=1) for each cell, ensuring that the parallel world itself has the proper importance (typically this is set to be equal to the importance of the last cell, =1)
   (c) Run analog (importance=1) and inspect the cell population column - does it look reasonable based on the macroscopic cross section of the cell?
3. Now using the cell populations in the analog run, establish what the importances need to be and plug these in. For this example this means raising the importance of each successive radial cell by an appropriate amount to achieve a constant track population per cell.
4. Biased (VR) run(s):
   (a) Run and inspect the cell population column.
   (b) Is the tack population constant (within a factor of 2 to the number of initial event particles thrown)?
   (c) If not, tweak the importances as needed and iterate.

3 Case study using the example included in ATMOSIM

The tally region for this problem is a ∼42000 km radius from the center of the Earth. The Earth's atmosphere is the sole attenuating medium, extending from the Earth's surface to roughly 1000 km altitude where the density of the atmosphere is ∼15 orders of magnitude thinner than at sea level, and beyond which is considered negligible and is ignored in this simulation (see Figure 2). The example utilizes a point source of neutrons emitted isotropically from a fixed altitude within this region. The energy range of primary neutrons is from 100 eV to 20 MeV. At sea level the mean free paths of these neutrons varies from about ten meters at low energy to 150 meters at the high energy range. The atmospheric density falls off with roughly 1/e over every seven kilometers, and the mean free paths are correspondingly increased.

Unlike a conventional deep-shielding problem, the attenuating medium varies with angle and altitude, and the distances are vast. For this initial implementation of VR, spherical importance regions were chosen, and defined in a parallel (massless) geometry in `atmoParallelConstruction.cpp`. A parallel geometry is used so that the importance regions can stretch across layers of atmosphere without affecting the mass model. Spherical cells have the advantage of isotropy about the source emission, but the disadvantage of varying density along their volume, which leads to an average importance for a cell in which mean free paths can vary by orders of magnitude. Figure 3 illustrates how these spherical importance cells cut across a wide range of atmospheric density.

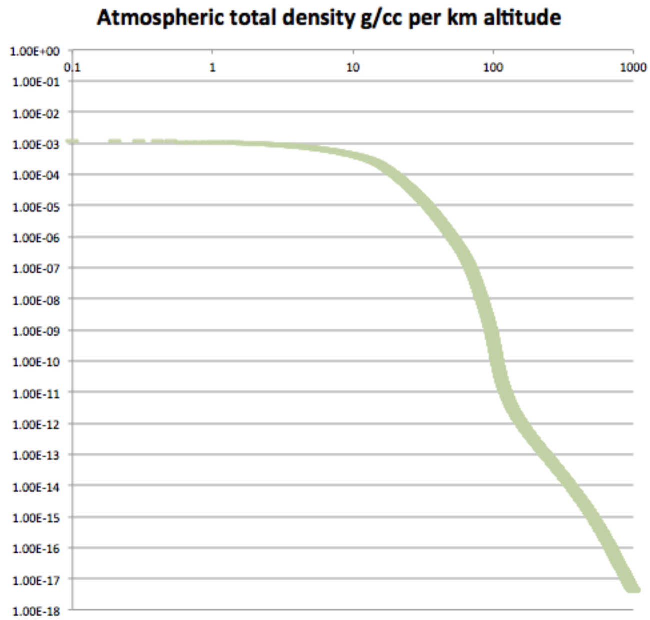

Figure 2: Total atmospheric density (g/cc) versus altitude, from NRL MSISE00 model.

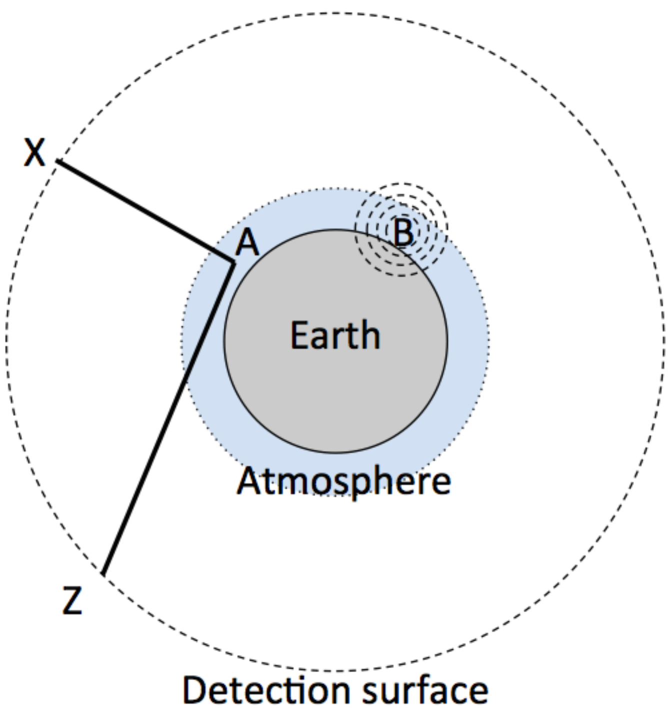

Figure 3: Illustration of the problem.

3.1 Comparison of Analog and VR results

For this example we examine the case of neutron emission at 10 km altitude, which is considered a fairly low altitude and a challenging simulation considering the atmospheric density at that altitude. The source is purely neutrons with a flat energy distribution between 100 eV and 20 MeV. Geometry cells are set up as spherical shells with radii: 100m, 200m, 500m, 1km, 2km, 3km, 4km, 5km, 6km, 7km, 8km, 9km, 10km, 20km, 30km, 40km, 50km, 100km, 200km, 500km, 1000km. In importance sampling it can be useful to set the importance cell range to equal the mean free path of the primary particles as this implies sucessive importances factors of 2, however this is not possible with such a wide range of energies and densities and long distances. Instead these radii were chosen because they provide sufficient sampling for the 10 km altitude problem. First, setting the importances to 1 for all cells, we run an analog simulation to determine the track population per cell. This is used to make a first pass at the importance factors. The importances are set to the inverse of the ratio of the track population in the cell to that of the origin cell. In the next run, this method should provide a reasonably flat track population. It is not necessary or practical to have a perfectly flat track population, however a decreasing population with distance from the source indicates more splitting is required, and an increasing population indicates excess splitting which will cost in extra time and provide diminishing returns to the reduction of the variance. A guideline is to keep the population within a factor of 2 of the number of thrown particles. The resulting track population per cell and the implied importance is shown in Table 1. Setting the importance per cell to the implied importance value, a VR run was made with the results shown in the rightmost column. For the VR result, a reasonably flat track population (within a factor of 2) is achieved. On average, each primary particle contributes one (low weight $\sim 10^{-6}$) particle to the tally.

Table 2 shows the results for 10 repeated analog and VR runs with different seeds. These runs are used to calculate the mean and variance of the number of neutrons tallying at the tally region. Each analog run and each VR run took 256 minutes. The real world time is held constant at 256 minutes in order to estimate the computational speedup. With this fixed time set, the number of events thrown for each analog run was 4,377,600 and the number for each VR run was 684,000. The analog results have quite poor statistics, which illustrates our problem. By comparison about one neutron track arrives at the tally region per neutron thrown for the VR runs. This is exactly what we want. The analog runs have tally statistics governed by the Poisson distribution, and so the variance is equal to the mean. For the VR runs the distribution is normal and the population variance is calculated for the normal distribution.

The average results over the ten runs are shown in Table 3. Two things are important to note: in the column marked N_{ev} norm, the normalized mean neutrons tallied agrees between the analog and VR run (6.4 versus 7.9) which indicates that tally bias was not introduced (although statistics are poor in this case); and the ratio of the analog to the VR variance is about 1000, indicating a computational speedup of 1000x. By Equation 1, this speedup indicates that if the VR result presented here took (for example) one hour to calculate, then to reach the same accuracy an analog simulation would require about 50 days of computation.

The variance of the number of events tallied is the simplest representation of the improvements brought by VR. We can also inspect the positional and energy information contained in those events. Figure 4 shows the distribution of arrival times at a radius of ∼42000 km as a function of neutron initial energy (the energy of the neutron at the event position). The color scale plots neutrons tallying per neutron emitted, per cm^2. The VR result is shown at left and the analog result at right, and both runs took the same time to run. The average tally flux over all times and energies is equal between the two plots, but clearly, the pixel-to-pixel variance in the VR data is smaller,

Table 1: Analog run population per cell, implied importance per cell, VR run population per cell

Cell	Outer radius	Analog population	Importance	VR population
0	100 m	604522	1	10000
1	200 m	600031	1	10218
2	500 m	599686	1	11116
3	1 km	543423	1.1	13593
4	2 km	300989	2.0	15902
5	3 km	69397	8.7	19056
6	4 km	17888	33.8	19475
7	5 km	4775	126	19640
8	6 km	1449	417	18756
9	7 km	504	1199	18140
10	8 km	202	2992	18128
11	9 km	94	6431	16441
12	10 km	55	10991	13589
13	20 km	35	17272	10668
14	30 km	5	120904	8748
15	40 km	4	151130	4609
16	50 km	3	201507	4945
17	100 km	3	201507	4505
18	200 km	2	302261	6265
19	500 km	1	604522	12384
20	1000 km	1	604522	12320

Table 2: Statistics for 10 repeated runs at 10 km for each modality (each run 256 minutes) to establish variance

Run	Analog	VR	VR Norm
1	1	737370	1.220
2	11	767789	1.270
3	7	755064	1.249
4	6	749832	1.240
5	7	737741	1.220
6	5	724179	1.198
7	11	708216	1.171
8	5	758151	1.254
9	7	755249	1.249
10	4	770728	1.275
Mean	6.4	-	1.235
Variance	6.4	-	1.052e-3
Var/Mean	100%	-	0.085%

Table 3: Comparison of Analog and VR results at 10 km - Average over 10 repeated runs

Sampling	Time (min)	Evt thrown	Evt tallied	Weight	Weight norm	N_{ev} norm	Variance
Analog	256	4377600	6.4	1	6.4	6.4	100%
VR	45	684000	746431	1.65e-6	1.235	7.9	0.085%

leaving the result much better defined than in the analog data.

Likewise in Figure 5, which compares the energies (log MeV) of neutrons tallied for the same composite runs. The left panel shows the VR result (solid line with no visible errors) and analog result (points with errors) binned to a minimum reasonable binning of the analog data. The mean energies of these spectra are roughly equal and the spectral shape is similar. The right panel shows the same data but with the finer binning available with the VR data, illustrating the greater accuracy provided by VR for the same run time.

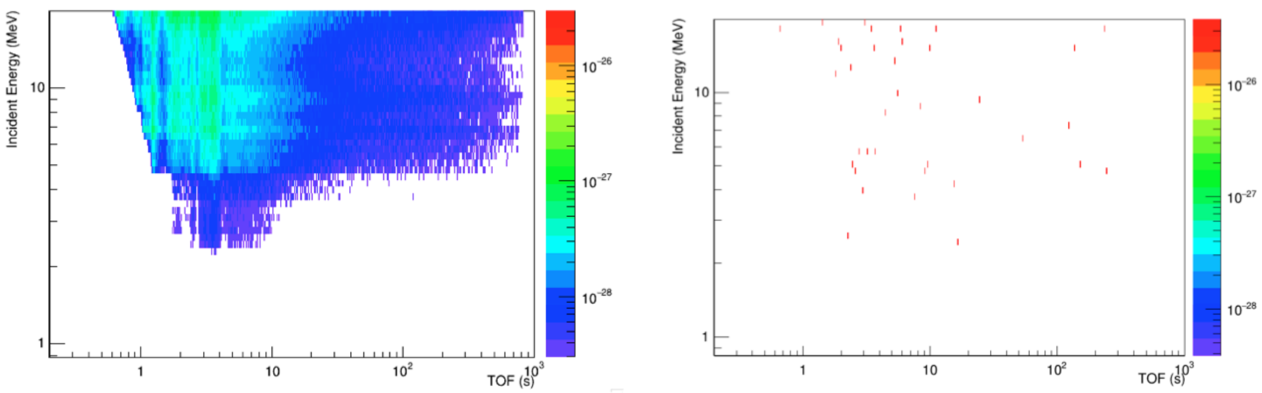

Figure 4: Initial neutron energy vs time of flight, for VR (left panel) and analog (right panel) runs of the same duration. The color scale is of units neutrons tallying per neutron thrown, per cm^2.

4 Summary

Importance sampling has been implemented in ATMOSIM. The equivalence between analog and VR techniques has been shown, and useful flux results were obtained for isotropic point source neutron emission at 10km altitude. A computational speedup factor of approximately 1000x was achieved for this problem. The technique of radial importances is not ideal for this problem geometry because there is angular dependence to the material properties that limits the effectiveness of the importance sampling. Future work could break up radial cells into angular wedges, for example.

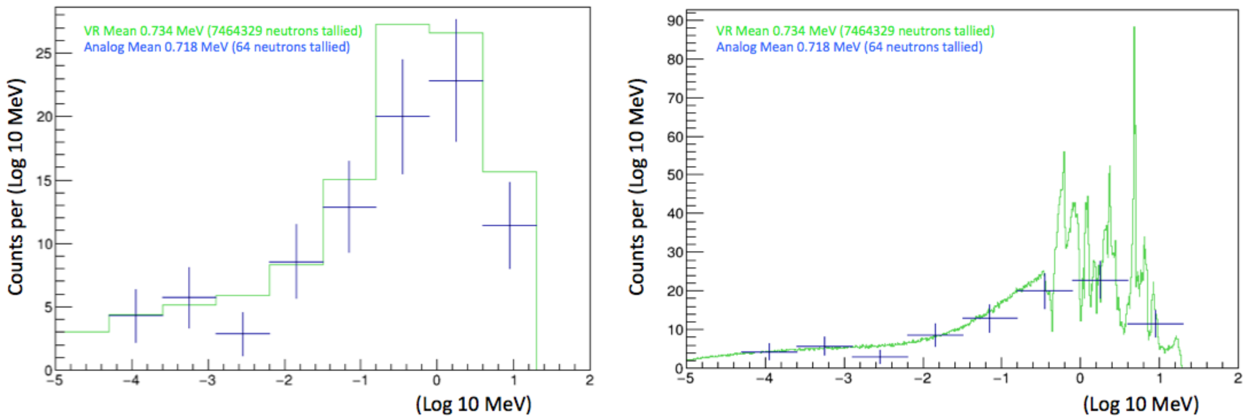

Figure 5: Tallied energy for analog and VR runs of the same duration. Left panel: VR (solid line) and analog (points) binned to minimum analog binning. Right panel: same data with finer VR binning.

References

[1] Hoover, Andrew, Kippen, Marc and Wallace, Mark. *GRESS User's Guide*, GRESS-SUG-001-24, Jan. 17, 2014.

[2] Agostinelli, S. *et al.* *GEANT4: A Simulation Toolkit*, NIM A506 (2003) 250-303.

[3] `https://ccmc.gsfc.nasa.gov/modelweb/models/nrlmsise00.php`

[4] Booth, Thomas E. *A Sample Problem for Variance Reduction in MCNP*, Los Alamos National Laboratory Report LA-10363-MS, Oct 1985.

[5] Dressel, Michael. *Geometrical importance sampling in Geant4: from design to verification*, CERN-OPEN-2003-048, Sep 18, 2003.