nrCascadeSim - A simulation tool for nuclear recoil cascades resulting from neutron capture

A.N. Villano[1], Kitty Harris[1], and Staci Brown[2]

1 Department of Physics, University of Colorado Denver, Denver CO 80217, USA **2** Department of Applied Mathematics & Statistics, University of New Mexico, Albuquerque NM 87131, USA

Summary

Neutron capture-induced nuclear recoils have emerged as an important tool for detector calibrations in direct dark matter detection and coherent elastic neutrino-nucleus scattering (CEνNS).

`nrCascadeSim` is a command-line tool for generating simulation data for energy deposits resulting from neutron capture on pure materials. Presently, silicon, germanium, neon, and argon are supported. While the software was developed for solid state detector calibration, it can be used for any application which requires simulated neutron capture-induced nuclear recoil data.

A "cascade" occurs when a neutron becomes part of a nucleus. The neutron can be captured to one of many discrete energy levels, or states; if the energy level is nonzero (not the ground state), then the state will eventually change so that it is zero. This can happen either all at once or in multiple steps — that is, the captured neutron may go from its state to the ground state, or it may go to another state with lower energy that is not the ground state (provided that one exists). The cascade refers to the particular "path" of energy levels that a captured neutron takes to get to the ground state from the neutron separation energy. Currently the code assumes that the neutrons that enter the nuclear system have negligible (zero) kinetic energy; this is a good approximation for thermal neutrons because 0.0254 eV (the average kinetic energy of a thermal neutron) is small compared to most nuclear recoil energy scales.

`nrCascadeSim` models many of these cascades at once and saves the energies along with other useful data to a single file, the structure of which is outlined in Figure 1.
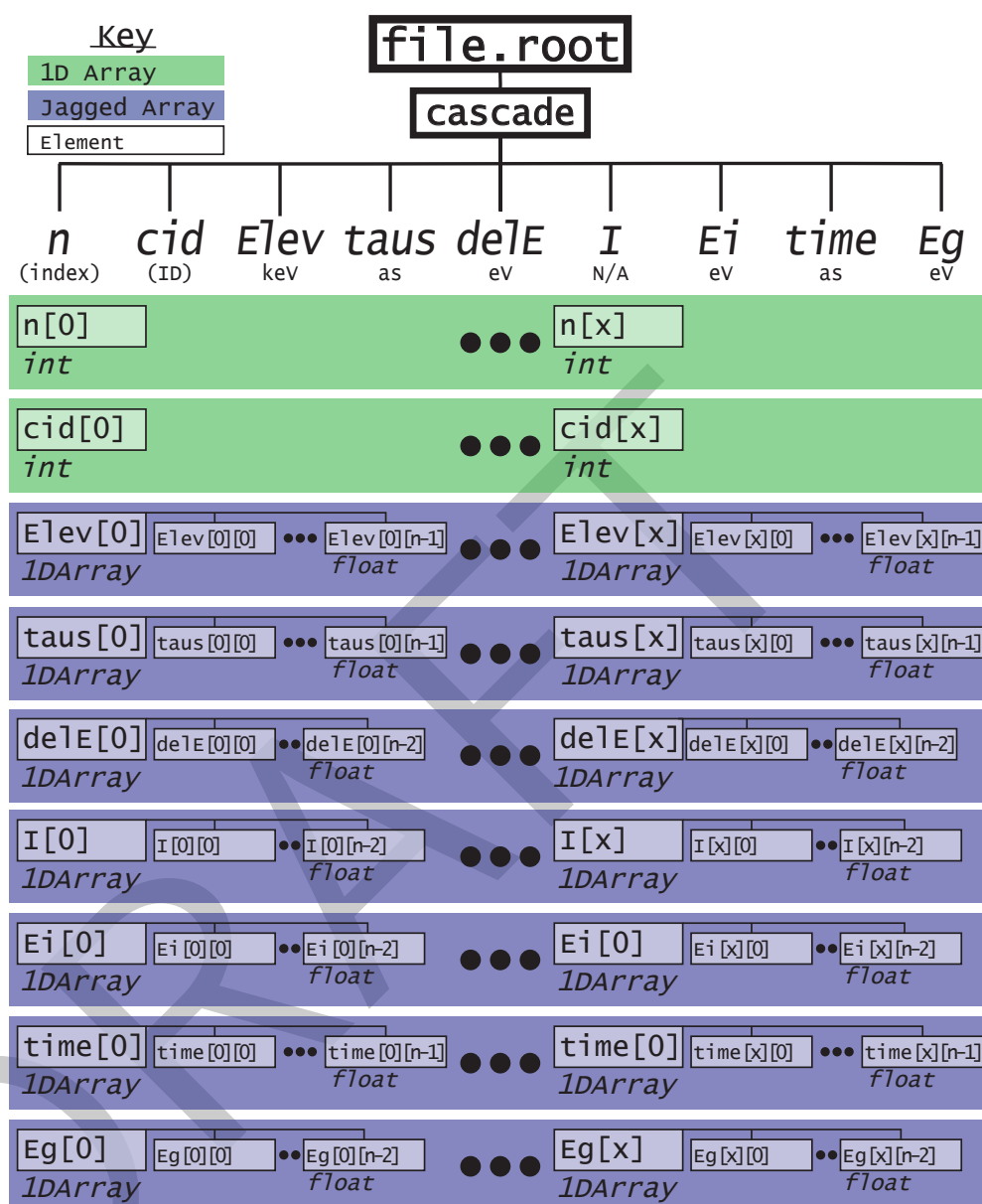
Figure 1: An outline of the structure of a ROOT (CERN, 2021) output file named `file.root`. Everything is contained within a top-level key called `cascade`. Beneath `cascade` are several other keys, each pointing to an array. Each array element corresponds to one cascade; the same index will point to the same cascade across arrays. `n` notes the number of energy levels in the cascade. `cid` is short for "cascade ID" and refers to the row number of the levelfile which was used to generate the cascade, starting from zero. Each element of `Elev` is an array noting the energy levels used, given in eV. Similarly, `taus` notes the lifetimes of these states used, given in attoseconds. Both `Elev` and `taus` will have entries with a length of the corresponding value of n, so if `n[3]` is four then the lengths of `Elev[3]` and `taus[3]` will both be four. `delE` lists the energies deposited during the cascade in eV, and will always a length of one less than n. I calculates the ionization in terms of a number of charges, and `Ei` combines `I` with `delE` to list the ionization energy in eV. `time` describes the simulation-generated time that the neutron spent at each energy level, in attoseconds, and has a length corresponding to n. Eg provides gamma energies associated with each decay, in MeV, and has a length corresponding to one less than n. The gamma energies are not included in any of the other energy arrays.

Models Used

When modeling deposits from neutron capture events, we want to look at the recoil of the nucleus as a result of these cascades. To determine how much energy is deposited, we must track how much the nucleus slows down between steps of the cascade as well as *how* each state change affects the nucleus' travel. nrCascadeSim assumes a constant deceleration that results from the nucleus colliding with other nearby nuclei. This means that it must simulate, along with the steps of the cascade, the time between each state — to calculate how much the nucleus slows down — and the angle between the nucleus' momentum before a decay and the momentum boost (gamma ray) resulting from the decay — to calculate the resulting momentum. The time between steps is simulated as an exponentially-decaying random variable based on the state's half-life[1], and the angle is simulated as having a uniform distribution on the surface of a sphere. Cascade selection is weighted by isotope abundance and cross-section as well as the probability of the energy level. In+ existing levelfiles, energy levels are derived from (Islam et al., 1991) for germanium and from (Raman et al., 1992) for Silicon.

The above process models the recoil energies, and the output gives both the total recoil energy for a cascade as well as the energy per step. For some applications, this may be the desired output, or the user may already have a particular process they will use for converting this energy to what they wish to measure. However, we also include, for convenience, the ionization yield and ionization energy of these recoils. This ionization yield assumes the Lindhard model(Lindhard et al., 1963):

$$
\begin{aligned}
Y &= \frac{kg_{(\epsilon)}}{1+kg_{(\epsilon)}} \\
g_{(\epsilon)} &= a\epsilon^{\gamma} + b\epsilon^{w} + \epsilon \\
\epsilon_{(E_r)} &= 11.5 E_r[keV]Z^{-7/3}
\end{aligned}
$$

Using the accepted value for Silicon ($k = 0.143$) or Germanium ($k = 0.159$), whichever is appropriate; $a = 3$; and $b = 0.7$.

Statement of Need

nrCascadeSim is a C/C++ package for generating a specified number of energy deposits resulting from nuetron capture-induced nuclear recoils. The energy levels and their lifetimes are customizable, and multiple isotopes of the same element can be present within the simulation. Pre-defined energy level files exist for silicon and germanium, which are constructed from the data in (Rosman & Taylor, 1998) and (Sonzogni & Shu, 2020). Outputs include energy deposits at each step, total kinetic energy deposits, and ionization energy deposits, making them useful for a variety of applications, including nuclear recoil calibrations for dark matter direct detection or coherent neutrino detection (CEνNS).

Example Use Case

Included in the repository is an example `test-example/Yields_and_Resolutions.ipynb` which users can follow to ensure the code is running correctly. This example both applies a yield model to the individual energy deposits and applies variation intended to simulate the resolution of the detector. The yield and resolution models are described in more detail in the

---

[1] It is most correct to use the half-life for the state given the state it will decay to. However, these are not generally well-known unless the branching ratios are well-known. If the ratios are well-known, then a correction can be made and incorporated into the input file.

example notebook. Figure 2 shows overlaid histograms of different combinations of analysis on the same data file.
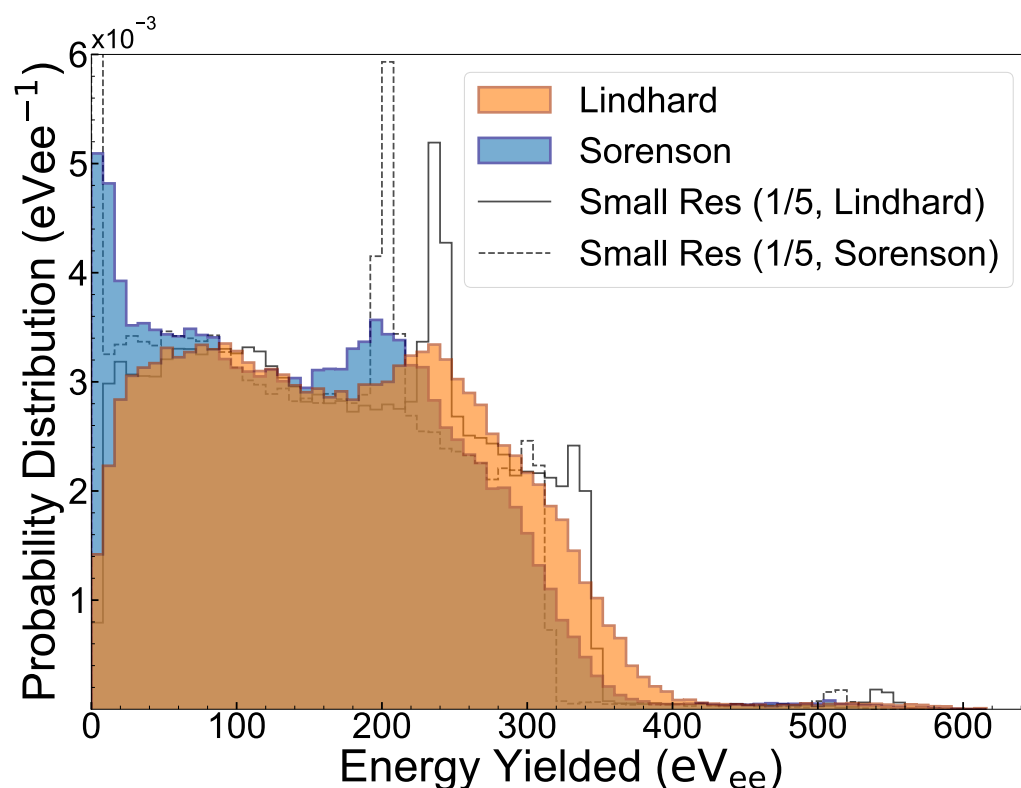

Figure 2: An overlaid histogram showing an example use case in which points are generated and then multiple yield models and resolutions are applied. In this example, the x-axis represents the ionization energy "yielded" by the cascade; this is effectively a way of noting what the detector reads out as opposed to what the pure kinetic energy of the cascade is. The Lindhard yield (Lindhard et al., 1963) is output by `nrCascadeSim` as Ei; the Sorenson yield (Sorensen, 2015) is applied to the values from `delE`. Resolutions are applied by adding random values generated from a Gaussian distribution of fixed width to the energy yield. The "Small Res (1/5)" histograms have Gaussians with 1/5 of the width of their counterparts. The y-axis represents the normalized frequency of energy yields.

References

CERN. (2021). *ROOT*. CERN. https://root.cern/install/

Islam, M. A., Kennett, T. J., & Prestwich, W. V. (1991). Radiative strength functions of germanium from thermal neutron capture. *Physical Review C, 43*, 1086–1098. https://doi.org/10.1103/physrevc.43.1086

Lindhard, J., Nielsen, V., & Scharff, M. (1963). Integral equations governing radiation effects. *Mat. Fys. Medd. Dan. Vid. Selsk., 33*, 1–41.

Raman, S., Jurney, E. T., & Lynn, J. E. (1992). Thermal-neutron capture by silicon isotopes. *Physical Review C, 46*, 972–983. https://doi.org/10.1103/PhysRevC.46.972

Rosman, K. J. R., & Taylor, P. D. P. (1998). Isotopic compositions of the elements 1997. In *Pure & Appl. Chem.* (Vol. 70, pp. 217–235). International Union of Pure; Applied Chemistry. https://doi.org/10.1515/iupac.70.0026

78 Sonzogni, A., & Shu, B. (2020). *Nudat 2*. Brookhaven National Laboratory. https://www.
79 nndc.bnl.gov/nudat2/

80 Sorensen, P. (2015). Atomic limits in the search for galactic dark matter. *Phys. Rev. D, 91*.
81 https://doi.org/10.1103/PhysRevD.91.083509