Documentation for sntools 0.7.1[*]

Jost Migenda[†]

January 11, 2021

Contents

[*] See `https://github.com/JostMigenda/sntools` for the most recent version.

[†] King's College London, `jost.migenda@kcl.ac.uk`

1 Introduction

sntools is a Monte Carlo event generator for supernova neutrino interactions.

It is a command line application that uses detailed time- and energy-dependent neutrino fluxes (provided by various supernova models) to generate interactions within the detector volume and write them to event files that can be used as an input for a full detector simulation. sntools was originally developed for Hyper-Kamiokande [1] and later extended to support different detectors and detector materials.

1.1 Installation Instructions

First, make sure you have Python[1] installed on your computer. (If you don't, you can get it from python.org or as part of the Anaconda distribution.)

In a terminal, run `pip install sntools` to install the latest version of sntools and all dependencies. Then, run `python -c 'import sntools; sntools.setup()'` to check whether sntools is working and to download the sample input file used below.

1.2 Quick Start Guide

To generate your first supernova neutrino events with sntools, open a terminal window and type in the following command:

```
sntools fluxes/intp2001.data --format nakazato --endtime 50
```

This generates events for the first 50 ms of a supernova based on the neutrino flux given in the input file `fluxes/intp2001.data`[2] (which is in the `nakazato` format) and writes them to a file named `outfile.kin` in the current directory.

A more realistic usage that demonstrates more of sntools' capabilities looks like this:

```
sntools fluxes/intp2001.data --format nakazato --channel es --detector
SuperK --distance 20 --verbose --output intp2001es.kin
```

This uses the neutrino flux given in the same input file to generate neutrino-electron elastic scattering events in Super-Kamiokande for a supernova at 20 kpc distance. It also produces more verbose output, which lets you see that sntools generates events separately for ν_e, $\bar{\nu}_e$, ν_x and $\bar{\nu}_x$ (which have different fluxes and cross sections), before merging them into an output file named `intp2001es.kin`.

The input file is a required argument. The following optional arguments are available:

`--format <value>` Format of input file(s). See section 2.5.

`--output <value>` Name of the output file.

`--mcformat <value>` Format of output file. Can be `NUANCE` (used e.g. by Hyper-Kamiokande) or `RATPAC` (used e.g. by WATCHMAN).[3]

[1] Python 3.5 or higher is strongly recommended, though Python 2.7 is still supported.

[2] This sample file, which is for a $20\,M_\odot$ progenitor from [2], is included as part of sntools. See `http://asphwww.ph.noda.tus.ac.jp/snn/` for more information.

[3] For a description of these formats see `http://neutrino.phy.duke.edu/nuance-format/` and `https://rat.readthedocs.io/en/latest/generators.html#external`.

`--detector <value>` Detector configuration. See section 2.2.

`--channel <value>` Interaction channel to generate events for. See section 2.3.

`--ordering <value>` Neutrino mass ordering. See section 2.4.

`--distance <value>` Distance to supernova (in kpc).

`--starttime <value>` **and** `--endtime <value>` Generate events in a certain time range (in milliseconds). If these aren't specified, events are generated for the full time range for which fluxes are given in the input file.

`--randomseed <value>` Set an integer as a seed for the random number generator, to generate events reproducibly.[4]

`--verbose` Print more detailed output.

`--version` Print the current version number and exit.

The command `sntools -h` displays a more detailed overview over all arguments, including their default values.

2 More Details

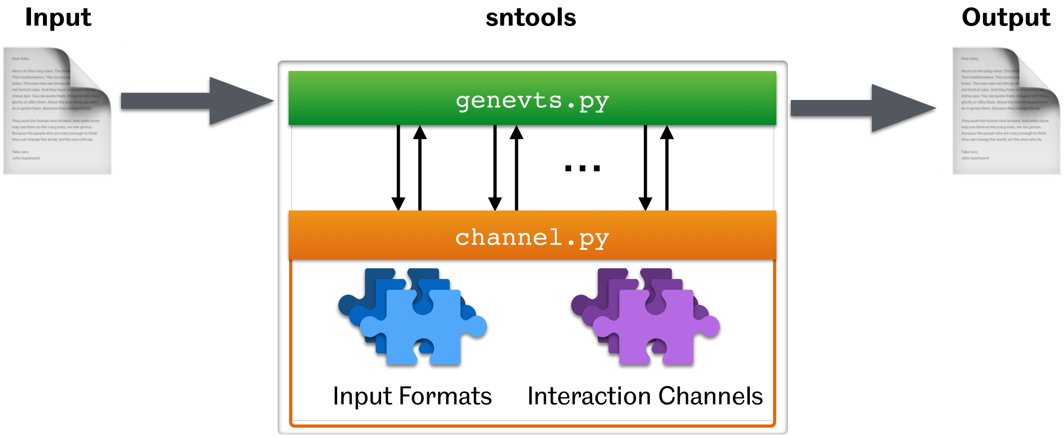

Figure 1: Overview over the structure of sntools. See text for a detailed description.

An overview over the structure of sntools is given in figure 1.

The main user interface is provided by the file `genevts.py`, which parses command line arguments and then calls code in `channel.py` to generate events. Events are generated separately for each combination of interaction channel and input species.[5] The

[4] Note: Python 2.7 will not give identical results to Python 3.x for the oxygen-16 CC channels.

[5] sntools uses the species ν_e, $\bar{\nu}_e$, ν_x and $\bar{\nu}_x$, with ν_x ($\bar{\nu}_x$) representing any one of ν_μ and ν_τ ($\bar{\nu}_\mu$ and $\bar{\nu}_\tau$). Since the energy of supernova neutrinos is too low to produce $\mu^\pm$ or $\tau^\pm$ in a detector, neutrinos with those two flavours interact in the same ways.

code first calculates the total number of events expected in each 1 ms bin, which is given by

$$N(t) = \iint \frac{\mathrm{d}\,\Phi(t, E_\nu)}{\mathrm{d}E_\nu} \frac{\mathrm{d}\,\sigma(E_\nu, E_e)}{\mathrm{d}E_e} \,\mathrm{d}E_e \,\mathrm{d}E_\nu, \tag{1}$$

where $\Phi(t, E_\nu)$ is the neutrino flux and $\sigma(E_\nu, E_e)$ is the cross section of the current interaction channel. It then picks the actual number of events to generate within that time bin from a Poisson distribution with expectation value $N(t)$. Finally, it generates events by rejection sampling from the energy spectrum of neutrino interactions at that time and the distribution of outgoing particle directions.

The event generation code relies on a plug-in architecture to support various different input formats and interaction channels. Input format plug-ins provide functions that read in the data from an input file and return the number luminosity as a function of time and energy. Interaction channel plug-ins specify properties of the interaction channel, like the neutrino species that undergo this interaction, and provide functions to calculate quantities like the differential cross section $\mathrm{d}\sigma(E_\nu, E_e)/\mathrm{d}E_e$ or the kinematically allowed energy range. This modular design makes sntools easily extensible, with roughly 100 lines of code required to add a new input format or interaction channel.

Finally, `genevts.py` collects the events generated in all interaction channels and writes them to an output file, which can be used as input for a full detector simulation.

The rest of this chapter describes the various components of the code and the physics behind them in more detail.

2.1 Detector Materials

sntools currently supports three detector materials: water, liquid scintillator and water-based liquid scintillator. Water is assumed to consist of an oxygen-16 nucleus and two hydrogen nuclei (free protons). Liquid scintillator is assumed to consist of n carbon-12 nuclei and $2n$ hydrogen nuclei (free protons). Water-based liquid scintillator is assumed to be a mixture of $x\,\%$ liquid scintillator and $(100 - x)\,\%$ water.

The detector material doesn't need to be specified explicitly; it is determined by the detector configuration.

2.2 Detector Configurations

Select a detector configuration by using the `--detector <value>` command line option. The following detector configurations are currently implemented:

`HyperK` Inner detector of Hyper-Kamiokande [3]. Cylinder with a height of 65.8 m and a diameter of 64.8 m, filled with water.[6]

`SuperK` Inner detector of Super-Kamiokande [4]. Cylinder with a height of 36.2 m and a diameter of 33.7 m, filled with water.

[6] In September 2019, the dimensions were changed slightly compared to the 2018 Design Report.

`WATCHMAN` WATCHMAN detector [5]. Cylinder with a height and diameter of 12.8 m, filled with water.

`WATCHMAN-LS` Same as `WATCHMAN`, but filled with liquid scintillator.

`WATCHMAN-WbLS` Same as `WATCHMAN`, but filled with water-based liquid scintillator (3 % LS, 97 % water).

`THEIA25` THEIA-25 geometry [6]. Box with dimensions 20 m × 18 m × 70 m, filled with 25 kt of water-based liquid scintillator (10 % LS, 90 % water).

`THEIA100` THEIA-100 geometry [6]. Cylindrical detector filled with 100 kt of water-based liquid scintillator (10 % LS, 90 % water).

2.3 Interaction Channels

sntools supports multiple different interaction channels described in this section. By default, it will generate events across all channels that are available in the selected detector, but it can be restricted to a single channel by using the `--channel <value>` command line argument, where `<value>` can be one of `ibd`, `es`, `o16e`, `o16eb`, `c12e`, `c12eb` and `c12nc`.

In water Cherenkov detectors like Hyper-Kamiokande, the dominant interaction channel for supernova neutrinos is inverse beta decay, which makes up about 90 % of events. Another important interaction channel is elastic scattering on electrons, which makes up only a few per cent of events but provides precise information on the direction of the supernova. The cross sections for both interactions have been calculated to a high level of precision.

Another important subdominant channel are charged-current interactions of ν_e and $\bar{\nu}_e$ on ^{16}O nuclei. While this channel suffers from large theoretical uncertainties, it is very sensitive to the high-energy tail of supernova neutrino fluxes, so that the number of events in this channel can vary greatly between models.

2.3.1 `ibd`: Inverse Beta Decay

In both water and liquid scintillator, inverse beta decay (IBD; $\bar{\nu}_e + p \rightarrow n + e^+$) is the dominant interaction channel for supernova neutrinos due to its relatively high cross section and low energy threshold of $E_\nu^{\text{thr}} \approx 1.8\,\text{MeV}$, as well as the large number of free protons in the detector. The observed energy of IBD events is closely related to the neutrino energy, making this an excellent channel to reconstruct the $\bar{\nu}_e$ spectrum.

In sntools, IBD is implemented using the full tree-level cross section calculated in reference [7] and including radiative corrections based on the approximation from reference [8].[7]

[7] That calculation uses the limit $m_e \rightarrow 0$. This approximation is accurate to better than 0.1 % above $E_e = 1\,\text{MeV}$.

2.3.2 es: Neutrino-Electron Scattering

In both water and liquid scintillator, elastic neutrino-electron scattering ($\nu + e^- \rightarrow \nu + e^-$) is a subdominant interaction channel due to its low cross section, which is only partially compensated by the large number of electrons in the detector. Elastic scattering events make up only a few per cent of all events but their angular distribution is strongly peaked into a forward direction, pointing away from the supernova. In detectors that can reconstruct the direction of scattered electrons, this channel can therefore be used to determine the direction of a supernova [9, 3].

Elastic scattering is sensitive to all neutrino flavours. However, the cross section of ν_e and $\bar{\nu}_e$, which can interact through both neutral and charged currents, is higher than that of ν_x and $\bar{\nu}_x$, which can interact only through neutral currents.

In sntools, elastic scattering is implemented using the tree-level cross sections from standard electroweak theory calculated by 't Hooft [10] and including one-loop electroweak and QCD corrections as well as QED radiative corrections as calculated in reference [11].

2.3.3 o16e and o16eb: Charged-Current Interactions on ^{16}O

In water, charged-current interactions of ν_e and $\bar{\nu}_e$ on ^{16}O nuclei,

$$\nu_e +^{16} \mathrm{O} \rightarrow e^- + X \tag{2}$$

$$\bar{\nu}_e +^{16} \mathrm{O} \rightarrow e^+ + X, \tag{3}$$

are a subdominant interaction channel.[8] Due to the high energy threshold of both interactions of approximately 15 MeV and 11 MeV, respectively, as well as the steep energy-dependence of the cross sections, the number of events in each channel is a very sensitive probe of the high-energy tail of the supernova neutrino flux. It may vary by more than two orders of magnitude depending on the supernova models and oscillation scenario.

In sntools, the cross section for both interaction channels is based on a new shell model calculation [12] which selected 42 different nuclear states and calculated their respective partial cross sections. To simplify the evaluation of the cross section, the implementation in sntools uses the four-group fit presented in reference [13]. This fit matches the cross sections calculated from the full set of nuclear states to within a few per cent at neutrino energies of up to 100 MeV. For a typical supernova neutrino flux, the difference in the resulting event spectra when using the four groups instead of all 42 nuclear states is very small.

An older approximation of both cross sections, which was commonly used in the literature until recently, can still be found in `interaction_channels/o16e_old.py` and `interaction_channels/o16eb_old.py`.[9]

[8] Charged-current interactions of other neutrino species do not occur, since the energy of supernova neutrinos is too small to produce muons or τ leptons.

[9] That approximation is based on a fit [14] to an earlier calculation [15] which tabulated the resulting

2.3.4 `c12e` and `c12eb`: Charged-Current Interactions on ^{12}C

In liquid scintillator, charged-current interactions of ν_e and $\bar{\nu}_e$ on ^{12}C nuclei,

$$\nu_e +^{12}\mathrm{C} \rightarrow e^- + X \tag{4}$$

$$\bar{\nu}_e +^{12}\mathrm{C} \rightarrow e^+ + X, \tag{5}$$

are a subdominant interaction channel.[10] Due to the high energy threshold of both interactions of approximately 17 MeV and 14 MeV, respectively, as well as the steep energy-dependence of the cross sections, the number of events in each channel is a very sensitive probe of the high-energy tail of the supernova neutrino flux.

In sntools, the cross section for both interaction channels is based on the calculation in reference [16]. To interpolate between the tabulated cross sections, sntools uses the fit

$$\sigma(E_\nu) = \sigma_0 \cdot \left[a_1(E_\nu - E_{\mathrm{thr}}) + a_2(E_\nu - E_{\mathrm{thr}})^2 + a_3(E_\nu - E_{\mathrm{thr}})^3\right], \tag{6}$$

where E_ν is the neutrino energy, E_{thr} is the energy threshold of the interaction and σ_0 and a_i are fit parameters. The energy of the outgoing lepton is approximated as $E_e = E_\nu - E_{\mathrm{thr}}$.

2.3.5 `c12nc`: Neutral-Current Interactions on ^{12}C

In liquid scintillator, neutral-current interactions on ^{12}C nuclei,

$$\nu +^{12}\mathrm{C} \rightarrow \nu' +^{12}\mathrm{C}^* \tag{7}$$

$$^{12}\mathrm{C}^* \rightarrow^{12}\mathrm{C} + \gamma \tag{8}$$

are a subdominant interaction channel that neutrinos and antineutrinos of all flavours contribute to equally. In sntools, the cross section for this interaction channel is based on the theoretical calculation in [17]

$$\sigma(E_\nu) = 1.08 \times 10^{-38}\,\mathrm{cm}^2 \left(\frac{E_\nu - E_{\mathrm{thr}}}{m_N}\right)^2 \cdot \beta\kappa, \tag{9}$$

where E_ν is the neutrino energy, $E_{\mathrm{thr}} \approx 15$ MeV is the energy threshold of the interaction, $m_N \approx 939$ MeV is the nucleon mass and $\beta\kappa = 1.11$ [18]. The energy of the outgoing γ is $E_\gamma = E_{\mathrm{thr}}$.

2.4 Treatment of Neutrino Flavour Conversion

sntools implements three different mass ordering scenarios that can be selected by using the `--ordering <value>` command line argument.[11]

total cross sections, instead of partial cross sections for each excitation energy. Therefore, the fit assumed a simplified scenario where all final nuclear states shared the energy of the ground state. This significantly overestimated the energy spectrum of the emitted $e^\pm$.

[10] Footnote 8 applies here as well.

[11] The alias `--hierarchy <value>` also exists.

The first scenario, `noosc`, assumes that neutrino oscillations do not take place such that the flux of a neutrino species ν_i observed by a detector on Earth, Φ_{ν_i}, is identical to the fluxes originating within the supernova, $\Phi^0_{\nu_i}$, which are given in the input files that sntools uses.[12]

The other two scenarios, `normal` and `inverted`, assume that adiabatic flavour conversion happens via the MSW effect since neutrinos traverse a smoothly varying density profile while exiting the star. The resulting observed fluxes are linear combinations of the initial fluxes, which, for normal mass ordering, are given by [19]

$$\begin{aligned}
\Phi_{\nu_e} &= \sin^2\theta_{13} \cdot \Phi^0_{\nu_e} + \cos^2\theta_{13} \cdot \Phi^0_{\nu_x} \\
\Phi_{\bar{\nu}_e} &= \cos^2\theta_{12}\cos^2\theta_{13} \cdot \Phi^0_{\bar{\nu}_e} + (1-\cos^2\theta_{12}\cos^2\theta_{13}) \cdot \Phi^0_{\bar{\nu}_x} \\
2\Phi_{\nu_x} &= \cos^2\theta_{13} \cdot \Phi^0_{\nu_e} + (1+\sin^2\theta_{13}) \cdot \Phi^0_{\nu_x} \\
2\Phi_{\bar{\nu}_x} &= (1-\cos^2\theta_{12}\cos^2\theta_{13}) \cdot \Phi^0_{\bar{\nu}_e} + (1+\cos^2\theta_{12}\cos^2\theta_{13}) \cdot \Phi^0_{\bar{\nu}_x},
\end{aligned} \tag{10}$$

while for inverted ordering, they are

$$\begin{aligned}
\Phi_{\nu_e} &= \sin^2\theta_{12}\cos^2\theta_{13} \cdot \Phi^0_{\nu_e} + (1-\sin^2\theta_{12}\cos^2\theta_{13}) \cdot \Phi^0_{\nu_x} \\
\Phi_{\bar{\nu}_e} &= \sin^2\theta_{13} \cdot \Phi^0_{\bar{\nu}_e} + \cos^2\theta_{13} \cdot \Phi^0_{\bar{\nu}_x} \\
2\Phi_{\nu_x} &= (1-\sin^2\theta_{12}\cos^2\theta_{13}) \cdot \Phi^0_{\nu_e} + (1+\sin^2\theta_{12}\cos^2\theta_{13}) \cdot \Phi^0_{\nu_x} \\
2\Phi_{\bar{\nu}_x} &= \cos^2\theta_{13} \cdot \Phi^0_{\bar{\nu}_e} + (1+\sin^2\theta_{13}) \cdot \Phi^0_{\bar{\nu}_x}.
\end{aligned} \tag{11}$$

In both cases, the factor of 2 in the last two equations accounts for the fact that ν_x ($\bar{\nu}_x$) includes the fluxes of ν_μ and ν_τ ($\bar{\nu}_\mu$ and $\bar{\nu}_\tau$). These equations assume purely adiabatic transition (corresponding to $P_H = 0$ in [19, 20]) as explained below. Values for θ_{12} and θ_{13} are taken from the Particle Data Group [21].

In cases where the detected flux is a mixture of original fluxes of different species, sntools generates events for each original species separately with the appropriate weighting factor applied. For example, when generating inverse beta decay events in the normal ordering, sntools will generate events first using the input flux $\Phi_{\bar{\nu}_e} = \cos^2\theta_{12}\cos^2\theta_{13} \cdot \Phi^0_{\bar{\nu}_e}$ and then using the input flux $\Phi_{\bar{\nu}_e} = (1-\cos^2\theta_{12}\cos^2\theta_{13}) \cdot \Phi^0_{\bar{\nu}_x}$, before finally combining both sets of events into one output file.

Several other effects that may induce additional time- and energy-dependent flavour conversion are not currently implemented in sntools. They are discussed in the following.

After the accretion phase, the revived shock front travels outwards and passes through the layer within the star where the adiabatic flavour conversion described above takes place. This causes a sudden change in the matter and electron density and can severely impact the flavour conversion processes [22]. Since this occurs after the shock wave is revived, it mainly affects the late-time part of the supernova neutrino signal.[13] Furthermore, this effect is highly dependent on the the density structure of each individual

[12] For simplicity, throughout this section I omit the geometrical factor $\frac{1}{4\pi d^2}$ which depends on the distance d of the supernova.

[13] For the Totani model, this effect is expected to become relevant more than 1 s after core-bounce [20]. Most other computer simulations don't extend out to more than a few 100 ms and would thus likely not exhibit this effect.

progenitor and cannot be taken into account by sntools. Instead, where appropriate, groups performing supernova simulations will need to include this effect in their codes and publish neutrino flux data that takes it into account.

Near the centre of the supernova, the high neutrino density could induce a matter effect that causes self-induced flavour conversion [23, 24]. These collective effects—and their observable consequences, which could include energy-dependent flavour conversion (so called "spectral splits")—are the subject of intense theoretical study [25, 26, 27, 28, 29], though no clear picture has yet emerged of how these effects will manifest in a given supernova. See reference [30] for a recent review. Flavour conversion may also be suppressed in dense matter [31, 32]. As a result, the consequences in a realistic supernova are currently not well understood and may depend on the progenitor [33]. They can therefore not be taken into account by sntools. Once theoretical understanding has improved, it may be more appropriate to include these effects in individual supernova simulations and publish neutrino flux data that takes this into account.

Finally, depending on the location of the supernova relative to the detector, the neutrino detector may be "shadowed" by the Earth. As neutrinos traverse the Earth's matter potential before detection, they can undergo energy-dependent flavour transitions which are, in principle, detectable in large liquid scintillator or water Cherenkov detectors [34]. The presence and amplitude of this shadowing effect depends on a combination of the location of the progenitor within the Milky Way, the detector location on Earth [35] and, due to the Earth's rotation, the time of day that the neutrinos arrive on Earth. These are not currently included in sntools.

2.5 Input Formats

sntools supports multiple different input formats for the neutrino fluxes from a simulation. To select one, use the `--format <value>` command line argument, where `<value>` can be one of `gamma`, `warren2020`, `nakazato`, `princeton` or `totani`. This section will briefly describe each format and the processing steps necessary to calculate the spectral number luminosity.

All formats contain separate information on the three species ν_e, $\bar{\nu}_e$ and ν_x. The fluxes of ν_x and $\bar{\nu}_x$ are typically assumed to be equal and most simulations don't provide separate fluxes for ν_x and $\bar{\nu}_x$. For simplicity, references to each species are omitted in the following.

2.5.1 Gamma Format

Files in this format contain, for each time step t_n, the luminosity L, mean energy $\langle E_\nu \rangle$ and mean squared energy $\langle E_\nu^2 \rangle$ of neutrinos. A sample file is provided under `fluxes/sample-gamma.txt`.

To reconstruct the neutrino spectrum from this, we assume that it is described by a normalized Gamma distribution [36, 37] given by

$$f(E_\nu) = \frac{E_\nu^\alpha}{\Gamma(\alpha+1)} \left(\frac{\alpha+1}{A}\right)^{\alpha+1} \exp\left[-\frac{(\alpha+1)E_\nu}{A}\right]. \tag{12}$$

In this formula, A is an energy scale, while α determines the shape of the distribution: $\alpha = 2$ corresponds to a Maxwell-Boltzmann distribution, while $\alpha > 2$ corresponds to a "pinched" spectrum, which is more typical for neutrino spectra from supernovae.

The first two energy moments of the distribution are

$$\langle E_\nu \rangle = \int_0^\infty \mathrm{d}E_\nu \, E_\nu f(E_\nu) = A \tag{13}$$

$$\langle E_\nu^2 \rangle = \int_0^\infty \mathrm{d}E_\nu \, E_\nu^2 f(E_\nu) = \frac{\alpha+2}{\alpha+1} A^2, \tag{14}$$

and therefore,

$$\alpha = \frac{\langle E_\nu^2 \rangle - 2\langle E_\nu \rangle^2}{\langle E_\nu \rangle^2 - \langle E_\nu^2 \rangle}. \tag{15}$$

Thus, the shape of the spectral number luminosity is uniquely determined by the mean energy $\langle E_\nu \rangle$ and the mean squared energy $\langle E_\nu^2 \rangle$, while the normalization is provided by $L/\langle E_\nu \rangle$. To determine the spectral number luminosity at arbitrary times, each of the three parameters is interpolated separately before calculating the spectral number luminosity using the interpolated values.

2.5.2 Warren2020 Format

Files in this format contain, for each time step t_n, the luminosity L, mean energy $\langle E_\nu \rangle$ and RMS energy $\sqrt{\langle E_\nu^2 \rangle}$ of neutrinos in the HDF5 format. After reading data from the input file, it is processed as described in section 2.5.1.

2.5.3 Nakazato Format

Files in this format[14] contain, for 20 energy bins E_k during each time step t_n, the quantities $\Delta N_k(t_n)/\Delta E_k$ and $\Delta L_k(t_n)/\Delta E_k$, which reflect the number luminosity and luminosity at those energies, respectively. For each energy bin, sntools calculates the mean energy within that bin, which is given by

$$\langle E_k \rangle = \frac{\frac{\Delta L_k(t_n)}{\Delta E_k}}{\frac{\Delta N_k(t_n)}{\Delta E_k}}, \tag{16}$$

and set the differential neutrino number flux at that energy to $\Delta N_k(t_n)/\Delta E_k$. Finally, a linear interpolation in time and cubic spline interpolation in energy are used to determine the spectral number luminosity at an arbitrary time and energy.

2.5.4 Princeton Format

Files in this format contain, for each time step t_n, the spectral luminosity $\mathrm{d}L/\mathrm{d}E$ for 20 logarithmically spaced energy bins E_k. I divide this by the central energy $\sqrt{E_k E_{k+1}}$

[14] See `http://asphwww.ph.noda.tus.ac.jp/snn/guide.pdf` for more details.

of the respective bin to get the spectral number luminosity at that energy. Finally, a linear interpolation in time and cubic interpolation in energy are used to determine the spectral number luminosity at an arbitrary time and energy.

This follows the procedure described in reference [38]. It is similar to that used for the Nakazato format described above, though with a different definition of the bin energy.

2.5.5 Totani Format

Files in this format contain, for each time step t_n, the total number of neutrinos emitted until that time, which makes it possible to calculate the number N_n of neutrinos emitted since the previous time step.

For 20 energy bins E_k per time step, a quantity X_k is provided, which is proportional to the number of neutrinos emitted during that time step and in that energy bin. I divide this by the width of each energy bin to get

$$X_k^{\text{spec}} = \frac{X_k}{E_{k+1} - E_k}, \tag{17}$$

which is proportional to the spectral number emission during that time step.

Integrating X_k^{spec} over all energy bins and dividing it by that integral gives the spectral number emission during that time step normalized to 1, X_k^{norm}. The spectral number luminosity at time t_n and energy E_k is given by

$$\frac{\mathrm{d}\,\mathrm{NL}(t_n, E_k)}{\mathrm{d}E} = \frac{N_n}{t_n - t_{n-1}} \cdot X_k^{\text{norm}}. \tag{18}$$

Finally, a linear interpolation in time and log cubic spline interpolation in energy are used to determine the spectral number luminosity at an arbitrary time and energy.[15]

3 Getting Help and Contributing

To report problems or ask for help, open an issue on GitHub or email the lead developer (see title page of this document).

sntools is designed to be versatile and support many different detectors and input formats. **Your contributions are welcome!** To help extend sntools, follow these steps:

- Clone the GitHub repository and run `pip install -r requirements.txt` to install all dependencies.
- Then, edit the code—see the subsections below for help with the most common ways to extend sntools.

[15] This approach closely follows the one used in code provided by Totani. There is excellent agreement of the calculated fluxes between Totani's code and sntools, with differences of at most a few per mille due to slight differences in the numerical interpolation algorithms used.

- To run the code you developed, use `python sntools/genevts.py [arguments]` instead of `sntools [arguments]`.
- To ensure that your changes have not introduced any bugs, run `python -m unittest discover` from the top-level directory of the repository. (Tests will also run automatically whenever you submit a pull request on GitHub.)
- Finally, please submit a pull request with your contributions!

sntools is released under a BSD license (see `LICENSE.md`) and with a Contributor Code of Conduct (see `CODE_OF_CONDUCT.md`). By participating in this project you agree to abide by its terms.

3.1 Add a New Detector Configuration

In `detectors.py`:

- Add the name of your detector configuration to the list `supported_detectors`.
- Add an `elif` clause for your detector in the `__init__` function of the `Detector` class and set the detector dimensions and material there.

If your new detector configuration uses a detector material that is not yet implemented, see section 3.2.

3.2 Add a New Detector Material

In `detectors.py`, create a new dictionary for your detector material which contains three entries:

- `molecular_weight`: Molecular weight of the material in g/mol.
- `density`: Density of the material in g/cm^3.
- `channel_weight`: Dictionary with supported interaction channels as keys and their respective weighting factors as values. The weighting factors are the number of potential targets per molecule. For example, water has 2 protons and 10 electrons per molecule, so the weighting factor is 2 for `ibd` and 10 for `es`.

If necessary, add new interaction channels which may occur in this detector material (see section 3.3). Finally, add a new detector configuration which uses this detector material (see section 3.1).

3.3 Add a New Interaction Channel

- In the `fluxes/` folder, copy `_example.py` to create a new module for your interaction channel and implement the relevant physics in there. (The source code contains extensive comments explaining all variables and functions.)
- In `genevts.py`, add the channel in the `parse_command_line_options()` function.
- In `detectors.py`, add the channel to all relevant detector materials.

3.4 Add a New Input Format

Ask whoever provided you with that input file, whether they could make the fluxes available in the `gamma` format described in section 2.5.1. If that's not possible:

- In the `formats/` folder, copy `gamma.py` to create a new module for your interaction channel and implement the three functions in there. (See the doc strings in the source code for details.)
- In `genevts.py`, add the format in the `parse_command_line_options()` function.

References

[1] Jost Migenda. *Supernova Model Discrimination with Hyper-Kamiokande.* PhD thesis, University of Sheffield, 2019. `arXiv:2002.01649`.

[2] Ken'ichiro Nakazato, Kohsuke Sumiyoshi, Hideyuki Suzuki, Tomonori Totani, Hideyuki Umeda, and Shoichi Yamada. Supernova Neutrino Light Curves and Spectra for Various Progenitor Stars: From Core Collapse to Proto-neutron Star Cooling. *Astrophys. J. Suppl.*, 205:2, 2013.

[3] K. Abe et al. Hyper-Kamiokande Design Report. 2018. `arXiv:1805.04163`.

[4] S. Fukuda et al. The Super-Kamiokande detector. *Nucl. Instrum. Meth.*, A501:418–462, 2003.

[5] M. Askins et al. The Physics and Nuclear Nonproliferation Goals of WATCHMAN: A WAter CHerenkov Monitor for ANtineutrinos. 2 2015. `arXiv:1502.01132`.

[6] M. Askins, Z. Bagdasarian, N. Barros, E. W. Beier, E. Blucher, R. Bonventre, E. Bourret, E. J. Callaghan, J. Caravaca, M. Diwan, S. T. Dye, J. Eisch, A. Elagin, T. Enqvist, V. Fischer, K. Frankiewicz, C. Grant, D. Guffanti, C. Hagner, A. Hallin, C. M. Jackson, R. Jiang, T. Kaptanoglu, J. R. Klein, Yu. G. Kolomensky, C. Kraus, F. Krennrich, T. Kutter, T. Lachenmaier, B. Land, K. Lande, J. G. Learned, V. Lozza, L. Ludhova, M. Malek, S. Manecki, J. Maneira, J. Maricic, J. Martyn, A. Mastbaum, C. Mauger, F. Moretti, J. Napolitano, B. Naranjo, M. Nieslony, L. Oberauer, G. D. Orebi Gann, J. Ouellet, T. Pershing, S. T. Petcov, L. Pickard, R. Rosero, M. C. Sanchez, J. Sawatzki, S. H. Seo, M. Smiley, M. Smy, A. Stahl, H. Steiger, M. R. Stock, H. Sunej, R. Svoboda, E. Tiras, W. H. Trzaska, M. Tzanov, M. Vagins, C. Vilela, Z. Wang, J. Wang, M. Wetstein, M. J. Wilking, L. Winslow, P. Wittich, B. Wonsak, E. Worcester, M. Wurm, G. Yang, M. Yeh, E. D. Zimmerman, S. Zsoldos, and K. Zuber. Theia: an advanced optical neutrino detector. *The European Physical Journal C*, 80(5):416, 2020.

[7] A. Strumia and F. Vissani. Precise quasielastic neutrino/nucleon cross-section. *Phys. Lett.*, B564:42–54, 2003.

[8] A. Kurylov, M. J. Ramsey-Musolf, and P. Vogel. Radiative corrections to low-energy neutrino reactions. *Phys. Rev.*, C67:035502, 2003.

[9] K. Abe et al. Real-Time Supernova Neutrino Burst Monitor at Super-Kamiokande. *Astropart. Phys.*, 81:39–48, 2016.

[10] Gerard 't Hooft. Predictions for neutrino - electron cross-sections in Weinberg's model of weak interactions. *Phys. Lett.*, 37B:195–196, 1971.

[11] John N. Bahcall, Marc Kamionkowski, and Alberto Sirlin. Solar neutrinos: Radiative corrections in neutrino - electron scattering experiments. *Phys. Rev.*, D51:6146–6158, 1995.

[12] Toshio Suzuki, Satoshi Chiba, Takashi Yoshida, Koh Takahashi, and Hideyuki Umeda. Neutrino-nucleus reactions on ^{16}O based on new shell-model Hamiltonians. *Phys. Rev.*, C98(3):034613, 2018.

[13] Ken'ichiro Nakazato, Toshio Suzuki, and Makoto Sakuda. Charged-current scattering off ^{16}O nucleus as a detection channel for supernova neutrinos. *PTEP*, 2018(12):123E02, 2018.

[14] R. Tomas, D. Semikoz, G. G. Raffelt, M. Kachelriess, and A. S. Dighe. Supernova pointing with low-energy and high-energy neutrino detectors. *Phys. Rev.*, D68:093013, 2003.

[15] E. Kolbe, K. Langanke, and P. Vogel. Estimates of weak and electromagnetic nuclear decay signatures for neutrino reactions in Super-Kamiokande. *Phys. Rev.*, D66:013007, 2002.

[16] E. Kolbe, K. Langanke, and P. Vogel. Weak reactions on C-12 within the continuum random phase approximation with partial occupancies. *Nucl. Phys. A*, 652:91–100, 1999.

[17] T.W. Donnelly and R.D. Peccei. Neutral Current Effects in Nuclei. *Phys. Rept.*, 50:1, 1979.

[18] B Armbruster et al. Measurement of the weak neutral current excitation C-12(nu(mu) nu'(mu))C*-12(1+,1,15.1-MeV) at E(nu(mu)) = 29.8-MeV. *Phys. Lett. B*, 423:15–20, 1998.

[19] Amol S. Dighe and Alexei Yu. Smirnov. Identifying the neutrino mass spectrum from the neutrino burst from a supernova. *Phys. Rev.*, D62:033007, 2000.

[20] G. L. Fogli, E. Lisi, A. Mirizzi, and D. Montanino. Probing supernova shock waves and neutrino flavor transitions in next-generation water-Cerenkov detectors. *JCAP*, 0504:002, 2005.

[21] P.A. Zyla et al. Review of particle physics. *PTEP*, 2020(08):083C01, 2020.

[22] Richard C. Schirato and George M. Fuller. Connection between supernova shocks, flavor transformation, and the neutrino signal. 2002. `arXiv:astro-ph/0205390`.

[23] Huaiyu Duan, George M. Fuller, J. Carlson, and Yong-Zhong Qian. Coherent Development of Neutrino Flavor in the Supernova Environment. *Phys. Rev. Lett.*, 97:241101, 2006.

[24] Huaiyu Duan, George M. Fuller, J Carlson, and Yong-Zhong Qian. Simulation of Coherent Non-Linear Neutrino Flavor Transformation in the Supernova Environment. 1. Correlated Neutrino Trajectories. *Phys. Rev.*, D74:105014, 2006.

[25] Georg G. Raffelt and Alexei Yu. Smirnov. Self-induced spectral splits in supernova neutrino fluxes. *Phys. Rev.*, D76:081301, 2007.

[26] Basudeb Dasgupta, Amol Dighe, Georg G. Raffelt, and Alexei Yu. Smirnov. Multiple Spectral Splits of Supernova Neutrinos. *Phys. Rev. Lett.*, 103:051105, 2009.

[27] Alexander Friedland. Self-refraction of supernova neutrinos: mixed spectra and three-flavor instabilities. *Phys. Rev. Lett.*, 104:191102, 2010.

[28] Huaiyu Duan and Alexander Friedland. Self-induced suppression of collective neutrino oscillations in a supernova. *Phys. Rev. Lett.*, 106:091101, 2011.

[29] Ignacio Izaguirre, Georg Raffelt, and Irene Tamborra. Fast Pairwise Conversion of Supernova Neutrinos: A Dispersion-Relation Approach. *Phys. Rev. Lett.*, 118(2):021101, 2017.

[30] Sovan Chakraborty, Rasmus Hansen, Ignacio Izaguirre, and Georg Raffelt. Collective neutrino flavor conversion: Recent developments. *Nucl. Phys.*, B908:366–381, 2016.

[31] A. Esteban-Pretel, A. Mirizzi, S. Pastor, R. Tomas, G. G. Raffelt, P. D. Serpico, and G. Sigl. Role of dense matter in collective supernova neutrino transformations. *Phys. Rev.*, D78:085012, 2008.

[32] Masamichi Zaizen, Takashi Yoshida, Kohsuke Sumiyoshi, and Hideyuki Umeda. Collective neutrino oscillations and detectabilities in failed supernovae. *Phys. Rev.*, D98(10):103020, 2018.

[33] Sovan Chakraborty, Alessandro Mirizzi, Ninetta Saviano, and David de Sousa Seixas. Suppression of the multi-azimuthal-angle instability in dense neutrino gas during supernova accretion phase. *Phys. Rev.*, D89(9):093001, 2014.

[34] Amol S. Dighe, Mathias Th. Keil, and Georg G. Raffelt. Identifying earth matter effects on supernova neutrinos at a single detector. *JCAP*, (06):006, 2003.

[35] A. Mirizzi, Georg G. Raffelt, and P. D. Serpico. Earth matter effects in supernova neutrinos: optimal detector locations. *JCAP*, 05(012), 2006.

[36] M. T. Keil, Georg G. Raffelt, and H.-Th. Janka. Monte carlo study of supernova neutrino spectra formation. *Astrophys. J.*, 590:971–991, 2003.

[37] Irene Tamborra, B. Müller, L. Hüdepohl, H.-Th. Janka, and Georg G. Raffelt. High-resolution supernova neutrino spectra represented by a simple fit. *Phys. Rev.*, D86(125031), 2012.

[38] Shaquann Seadrow, Adam Burrows, David Vartanyan, David Radice, and M. Aaron Skinner. Neutrino Signals of Core-Collapse Supernovae in Underground Detectors. *Mon. Not. Roy. Astron. Soc.*, 480(4):4710–4731, 2018.