WimpSim 3.04:
WimpAnn and WimpEvent

Simulation of WIMP annihilations in the Earth/Sun and propagation to a neutrino telescope

Programs written by

Joakim Edsjö[*]

Oskar Klein Centre for Cosmoparticle Physics and Department of Physics
Stockholm University, AlbaNova
SE-106 91 Stockholm, Sweden

January 9, 2013

WimpSim version: 3.04
Uses:
DarkSUSY 5.0.5 or later
nusigma 1.17 or later
Pythia 6.426 or later
SLALIB 2.5.5 (from 2.5.3 should work) (for positional astronomy), distributed under the Gnu Public License, v. 2

Abstract

The program package WimpSim consists of two parts, WimpAnn and WimpEvent. The program WimpAnn simulates WIMP annihilation in the Earth/Sun with the help of Pythia and neutrino interactions in the Sun are simulated with nusigma. Output is given event-by-event to large event files. Full three-flavour neutrino oscillations are also included in these simulations. Neutrino regeneration from τ decay arising in charged current interactions on the way out of the Sun are also included. WimpAnn will generate neutrino events (with all neutrino state amplitudes and phases intact) at 1 AU from the Sun (for annihilation in the Sun) and in the center of the Earth (for annihilation in the Earth).

The event files created by WimpAnn can be read by WimpEvent, that takes these neutrinos and propagates them to a neutrino telescope (including neutrino oscillations on their way). At the detector, neutrino interactions are simulated with nusigma and event files can be generated with either neutrinos or leptons and hadronic showers. Easy-to-use tables summarizing the fluxes are also produced with WimpEvent.

Both these programs keep track of all three neutrino flavours (and for neutrinos and antineutrinos).

---

[*]E-mail address: edsjo@fysik.su.se

Disclaimer

NOTE. This set of codes have been carefully tested by me and other people, but if you find any problems, please let me know. It is also an active code in the sense that improvements/additions will be added as requested. In particular, please be aware that as of version 3.0, WimpSim uses SLALIB [9] for accurate positional astronomy calculations. Even if this new interface has been tested to ensure that it is correct, please check to see that it agrees with your expectations and needs. If you need some changes in the event files, please let me know.

Also be aware of that this manual is only a quick write-up, don't hesitate to ask me if you have any questions.

As of version 3.0, the format of the event files has also changed. If you are reading event files, please update your event file reader. Version 3.0 of WimpEvent also requires version 3.0 of WimpAnn, i.e. you cannot use WimpEvent 3.0 with older WimpAnn files.

In version 3.03, the headers of the WimpAnn and WimpEvent event files were expanded. Hence, to ensure proper reading of the event files, only use event files from WimpAnn together with WimpEvent from the same WimpSim release.

Please check the files HISTORY-WimpAnn and HISTORY-WimpEvent for a more complete list of changes of the two programs.

1   Introduction

The two programs WimpAnn and WimpEvent are meant to work together, but for practical reasons they are split into two programs. The main reason for this split is that WimpAnn can generate event files without having to know about a particular detector location, whereas the final part of the simulations, where the detector location enters, is done with WimpEvent. The main advantage of this layout is that the time-consuming simulations with WimpAnn can be reused for many different detector configurations or time windows of observation.

This short note is meant to describe the basic physics parts that go into these simulations and describe the input and output format that these programs produce. It will also describe the most important internal data structures of the code, in case you want to peek inside it. We will start with a description of WimpAnn and then continue to WimpEvent.

2   WimpAnn - simulation of annihilations in the Earth/Sun

2.1   Introduction

WimpAnn simulates annihilations of WIMPs in the Sun and the Earth. In doing that it has to take into account various effects, like hadron interactions in the center of the Sun/Earth, neutrino interactions on the way out of the Sun and neutrino oscillations. Apart from neutrino oscillations, these processes are described in [1, 2, 3, 4]. The current version of WimpAnn is an updated version compared to those publications, where the main updates are:

- A new neutrino-nucleon scattering Monte Carlo nusigma [6] has been developed and is used for neutrino-nucleon scatterings.

- Neutrino oscillations are included with a numerical full three-neutrino oscillation code.

- More annihilation channels are added (monochromatic neutrinos).

- The program is now event based and creates large event tables instead of summary data tables (these are created in addition though).

The physics in the new version (or more specifically, the effect of neutrino oscillations) are described in [5]. In the following subsections, various part of WimpAnn will be described in some detail.

2.2   Annihilations in the center of the Sun/Earth

Wimp annihilations in the center of the Sun/Earth are produced with Pythia 6.414 or later. WimpAnn can simulate a set of different annihilation channels, which are listed in the Table 1.

2.3   Neutrino interactions

Neutrino interactions are generated with the routine nuNevent which is part of the nusigma [6] package developed by J. Edsjö. Using this package instead of Pythia for neutrino-nucleon interactions has a few advantages:

- nusigma does not have the hard-coded phase space cuts that Pythia has. The difference shows up at high outgoing lepton energy, where Pythia underestimates the cross section somewhat.

- It is about four times faster than Pythia.

	Pythia codes		
Channel number	Particle 1	Particle 2	Annihilation products
1	1	-1	$d\bar{d}$
2	2	-2	$u\bar{u}$
3	3	-3	$s\bar{s}$
4	4	-4	$c\bar{c}$
5	5	-5	$b\bar{b}$
6	6	-6	$t\bar{t}$
7	21	21	gg
8	24	-24	W^+W^-
9	23	-23	Z^0Z^0
10	13	-13	$\mu^-\mu^+$
11	15	-15	$\tau^-\tau^+$
12	12	-12	$\nu_e\bar{\nu}_e$
13	14	-14	$\nu_\mu\bar{\nu}_\mu$
14	16	-16	$\nu_\tau\bar{\nu}_\tau$
100	-	-	KK DM (UED B(1))

Table 1: The annihilation channels implemented into WimpAnn, together with the particle codes for these channels in Pythia. Note that channel 10 is just included to use the same channel numbering as in DarkSUSY, it gives no neutrinos for annihilations in the Sun/Earth. Channel 100 is using a standard set of branching fractions for UED DM B(1).

- Since it is separate from Pythia, there is no need to use Pythia for different kind of processes in the same run, which has some advantages.

Compared to other neutrino-nucleon scattering codes, nusigma also has the advantage to have either protons or neutrons as target (i.e. it doesn't assume an isoscalar target). As of this writing, nusigma uses the CTEQ6-DIS structure functions for protons and neutrons.

2.4   Neutrino oscillations

Neutrino oscillations are included in a very general three-neutrino scenario, i.e. we make no approximations of two-neutrino oscillations (even if this is a valid approximation for most neutrino oscillation parameters for part of the passage of the Sun). We refer the interested reader to [5] for further details on how these are implemented. We can summarize the procedure as follows: Instead of having flavour eigenstates, we define a neutrino state vector

$$\nu = \begin{pmatrix} a_e \\ a_\mu e^{i\phi_\mu} \\ a_\tau e^{i\phi_\tau} \end{pmatrix}. \tag{1}$$

where the square of the amplitudes tell us what the probability is to find the neutrino in a given flavour eigenstate at a particular location. Note that we rotate the state as done above to set the phase of ν_e to zero, as only the relative phases matter. We then evolve these states with a full vacuum and matter Hamiltonian, using a general parameterization of the neutrino oscillation parameters. We use the density profile of the Sun as given by the Standard Solar Model BS05(OP) [7]. This is coded into DarkSUSY [8], so we link WimpAnn to DarkSUSY. Via the link to DarkSUSY we can also choose different solar models if preferred.

2.5   Program layout

The layout of WimpAnn is as follows: First annihilation events are created with Pythia, possible interactions of the annihilation products are taken into account, and then the neutrinos are propagated out from the Sun. For practical reasons (and to gain speed), events are created in chunks of 100 000 events at a time. This is not anything you as a user need to worry about as this division is taken care of by WimpAnn. As events are created, the same annihilation events are used both for annihilation in the Sun and the Earth (to gain speed), but interactions are of course treated differently for the Earth and the Sun. These events are stored internally in data arrays before written to event files or filling up summary tables at the end.

2.6   Input parameters

WimpAnn requires the following input:

- The annihilation channel, see table 1

- The WIMP mass in GeV, M_{WIMP}

- If annihilations in the Earth should be simulated as well or not.

- The number of events (i.e. annihilations) to be simulated.

- If full output should be written to the screen or not (choose not to for typical batch runs)

- The neutrino mixing angle θ_{12} in degrees.

- The neutrino mixing angle θ_{13} in degrees.

- The neutrino mixing angle θ_{23} in degrees.

- The CP-breaking phase δ for neutrino oscillations.

- The mass squared difference Δ^2_{21} in eV^2.

- The mass squared difference $[\Delta^2_{31}]$ in eV^2.

- The output file name for events from the Sun.

- The output file name for events from the Earth.

- How many files to split the output event files into (default=1).

- The output file name for information files (with information about the run).

- The option to produce summary files (with summed distributions / fluxes) or not.

- The output file name for summary distributions for the Sun.

- The output file name for summary distributions for the Earth.

- The seed for the random number generator.

kfnui	Neutrino
1	ν_e
2	$\bar{\nu}_e$
3	ν_μ
4	$\bar{\nu}_\mu$
5	ν_τ
6	$\bar{\nu}_\tau$

Table 2: The neutrino codes used in WimpAnn and WimpEvent.

2.7 Internal arrays and variables of importance

In case you want to peek inside the code, here comes some descriptions about the most important data structures in the code. Pythia has its own set of particle codes and we need to convert between Pythia codes and the particle codes in e.g. nuNevent. For this purpose we define an array kfnui that takes Pythia particles codes kf and convert these to internal neutrino index codes, kfnui. The definition of these are given in Table 2

There is also another array nutype that converts from kfnui to neutrino codes in nuNevent (1 for neutrino and 2 for antineutrino).

2.7.1 Array descriptions

For the Sun, WimpAnn uses three internal arrays, nusi(j,i) with integer values, nusr(j,i) with real values and nusc(j,i) with complex values to store information about each neutrino (numbered by i). The entries in these tables are as follows

nusi(1,i)	Particle code (Pythia) for the parent of the neutrino.
nusi(2,i)	Neutrino code for the neutrino when created in the center of the Sun (see table 2).
nusi(3,i)	Number of hadron interactions (for $c\bar{c}$ and $b\bar{b}$) in the Sun.
nusi(4,i)	Number of neutrino interactions on the way out of the Sun.
nusi(5,i)	Neutrino code for the neutrino at last interaction in the Sun. Note, we only project on flavour eigenstates if a charged current interaction has occured, so for neutral currents this neutrino code is what we would have got if we had made a measurement at that particular radius.
nusi(6,i)	Neutrino status: Bit 0 set if neutrino is absorbed. Bit 1 set if interactions should be taken into account (reset to 0 when taken care of). Bit 2 set if the neutrino is a secondary neutrino from $\tau^{\pm}$ decay. Bit 3 set if the neutrino was originally created in the center of the Sun.
nusi(7,i)	Neutrino code at the surface of the Sun. Note that we do not project on flavour eigenstates at the surface of the Sun for the full propagation. The neutrino code at the surface of the Sun is what we would have got if we had made a measurement there.
nusr(1,i)	Neutrino energy when created in the center of the Sun.
nusr(2,i)	Neutrino energy in the Sun after the last interaction on the way out.
nusr(3,i)	Fractional radius in the Sun of the last interaction.
nusc(1,i)	Amplitude and phase of ν_e (or $\bar{\nu}_e$). (If the neutrino is a neutrino or anti-neutrino can be read-off from nusi(5,i).) At the end of the run, the amplitudes will be given at the Earth's perihelion distance to the Sun.
nusc(2,i)	Amplitude of ν_μ (or $\bar{\nu}_\mu$).
nusc(3,i)	Amplitude of ν_τ (or $\bar{\nu}_\tau$).

For the Earth, a similar set of arrays, nuei(j,i) and nuer(j,i) are used with the following entries

nuei(1,i)	Particle code (Pythia) for the parent of the neutrino.
nuei(2,i)	Neutrino code for the neutrino when created in the center of the Earth (see table 2).
nuei(3,i)	Number of hadron interactions in the Earth.
nuei(4,i)	Neutrino status: Bit 0 set if neutrino is absorbed.
nuer(1,i)	Neutrino energy when created in the Earth (could be different from nusr(1,i) in case of hadron interactions in the Sun.

Note that these arrays are filled for up to 100 000 annihilations at a time, after which summary tables are filled and events written to disk. The process then repeats until the desired number of annihilation events has been generated.

2.8 Output

The main output from WimpAnn is given as an event listing, where each neutrino is an event. For the Sun, the neutrino oscillation code has propagated all the neutrinos to a distance of 1 AU from the Sun (Earth's mean distance to the Sun), where a neutrino state is given by Eq. (1). We give the results on this general format, as one would in general want to propagate these neutrinos further to the detector, taking the path length through the Earth into account (this is what WimpEvent will do later). If we would project out the neutrino states on the flavour eigenstates at the Earth, we would lose the oscillation phases and would not be able to do this last propagation step correctly.

For annihilation in the Earth though, WimpAnn stores the results at creation close to the center. As they are created as flavour eigenstates, there is no need to store the amplitudes and phases in this case. For both the Sun and the Earth, the remaining part of the propagation to the detector is done with WimpEvent, described later in this document.

The output files from the Sun will have the following format

```
# Header with run information (program versions, input parameters etc)
Enu nutype a_e a_mu a_tau phi_mu phi_tau
```

where the amplitudes and phases are as given by Eq. (1) above. Enu is the energy of the neutrino (in GeV) and nutype is the neutrino type (1 for neutrino and 2 for anti-neutrino). Note that the phase of the electron neutrino has been rotated away.

For the Earth, the output files are a bit simpler,

```
# Header with run information (program versions, input parameters etc)
Enu nui
```

where Enu is the neutrino energy (in GeV) and nui is the neutrino type according to Table 2.

In addition to the event files described above, WimpAnn can also produce a set of summary tables. These tables contain the yields, dN_i/dz of

- Neutrinos (all six separately) at creation in the center of the Sun.

- Neutrinos (all six separately) at the surface of the Sun.

- Neutrinos (all six separately) at 1 AU from the Sun.

- Neutrinos (all six separately) arising from τ decay (after a charged current neutrino-nucleon interaction in the Sun) at the surface of the Sun.

- Neutrinos (all six separately) arising from τ decay (after a charged current neutrino-nucleon interaction in the Sun) at 1 AU from the Sun.

- Neutrinos (all six separately) at creation in the center of the Earth.

These tables are written to summary output files and can be used by other tools to plot fluxes/yields.

2.8.1   Format of Earth summary files

After the header, the earth summary files contains six lines, with 200 numbers each. These numbers are the differential yield, dN_i/dz in 200 bins at $z = 0.0025, 0.0075, \ldots, 0.9975$. z is the neutrino energy fraction in units of the WIMP mass, $z = E_\nu/M_{\mathrm{WIMP}}$. The six lines are for the six different (anti)neutrino flavours 1-6, given in Table 2.

2.8.2   Format of Sun summary files

For the Sun, the summary files contain roughly the same information, but now we have neutrinos sampled at different locations. Hence, there are now 30 lines after the header. Each one contains the differential neutrino yield dN/dz in 200 bins $z = 0.0025, 0.0075, \ldots, 0.9975$ as before. The meaning of the different lines are

Lines 1-6: Neutrino yields at creation at the center of the Sun for neutrino 1-6

Lines 7-12: Neutrino yields at the surface of the Sun $(R_\odot)$ for neutrino 1-6

Lines 13-18: Neutrino yields at 1 AU for neutrino 1-6

Lines 19-24: Neutrino yields from secondary neutrinos at the surface of the Sun $(R_\odot)$ for neutrino 1-6. Note that these yields only include the contribution from decay of τ^+/τ^- that arise in $\nu_\tau/\bar{\nu}_\tau$ charged current interactions on the way out of the Sun. The neutrino yields on lines 7-12 on the other hand include the *total* neutrino yields (both the primary component arising from the propagated neutrinos from the center and the secondary component).

Lines 25-30: Neutrino yields from secondary neutrinos at 1 AU for neutrino 1-6. Note that these yields only include the contribution from decay of τ^+/τ^- that arise in $\nu_\tau/\bar{\nu}_\tau$ charged current interactions on the way out of the Sun. The neutrino yields on lines 13-18 on the other hand include the *total* neutrino yields (both the primary component arising from the propagated neutrinos from the center and the secondary component).

Note that the yields given above are projected out on flavour eigenstates. In reality though, we keep all the amplitudes and phases, so the yields above are mainly given for illustrative purposes. For more details of the files, the user is referred to the code to see what is written to the files.

2.8.3   Splitting event files

As of WimpAnn 3.03, there is a possibility to split the output event file into several files. If you do this you need to make sure that the total number of annihilations and the number of files fulfill the following criteria:

a) There has to be equal number of annihilations in each split file

b) The number of annihilations in each split file should be evenly divisible by 100 000 (this is because the simulation is divided up into chunks of 100 000 annihilations to gain speed).

Also note that when WimpAnn files are fed into WimpSim the number of annihilations will enter in the event weights. For these split files, it is **the number of annihilations in each file** that will enter into these weights, **not** the total number of annihilations in the run. Hence, if you e.g. perform a run of 1 000 000 annihilations split into 10 files, the weights in the WimpEvent files will be a factor of 10 higher than if all 1 000 000 annihilations had been put into a single file.

For normal use, the easiest would be not to split the event files at all, but for distributed computing on e.g. a cluster it might be advantageous to utilize this option. If you are mostly interested in the summary files and not the actual event files, it is highly recommended to not split files.

3   WimpEvent - propagation to the detector and projecting on flavour eigenstates

3.1   Introduction

WimpEvent takes over where WimpAnn left the neutrinos and propagates these to the detector and projects them out on flavour eigenstates there. The basic philosophy with WimpEvent is that it

- is event based and generates events (neutrinos, charged leptons and hadronic showers) at the detector

- creates summary tables to be used by other tools (e.g. DarkSUSY).

- attaches a time stamp (time of the year or modified Julian date (MJD)) and a weight to each event.

As the simulation procedure is quite different for the Sun and the Earth, we will first discuss the Sun and later the Earth.

3.2   Annihilation in the Sun

3.2.1   Detector location and time stamp

As we need to propagate the neutrinos to the detector, we need to know where the detector is located. We also need to know at what time of the year (and time of the day) that the event occurs to know both a) how long the Sun-Earth distance is and b) to know how much of the Earth the neutrino has to propagate through.

In WimpEvent, this can be done in two different ways, one simplified approximate method and one more accurate. The accurate one is the preferred one, but the simplified one is kept for backwards compatibility. Let's describe the two methods below.

Simplified timing treatment. In this treatment we calculate the time stamps in an approximate fashion by assigning an (approximate) time stamp to each event. We here make three simplifying approximations

- We assume that perihelion occurs at Jan 1 at time 00:00.

- We assume that the winter solstice occurs at Jan 1 at time 00:00 (i.e. that the Earth's axis tilt is maximal at new year)

- We assume that the detector is on the back side of the Earth at Jan 1 at 00:00 (i.e. the Sun is exactly at South at new year, as seen from the detector)

These approximations are a bit rough, but will be good enough for most applications. However, some care need to be taken when treating the time stamp of the event. For example, the main use of the time stamp is to give the time of the year, which affects the Sun-Earth distance (due to the eccentricity of the Earth's orbit), but also to calculate how much of the Earth that is passed through. To calculate averaged spectra, this method is good enough, but for more detailed work one should use the accurate method below.

Accurate timing treatment - Modified Julian Dates. As of version 3.0 WimpSim is linked to SLALIB to perform accurate positional astronomy calculations. In particular, the time window of the simulations should now be given as modified Julian dates (MJDs) instead of approximate time of years as for the simplified method. The Earth's and Sun's location as well as the Earth's rotation is then accurately calculated for any given MJD. The SLA_EVP routine in SLALIB is used to calculate the Earth-Sun positions. Note, one can also use the more accurate routine SLA_EPV, but it is significantly slower and the improved accuracy is not needed for WimpSim. If the user anyway wants to switch to this routine, set the flag wesunacc to 2 (1 is the default SLA_EVP routine). To convert to the local sidereal time (LST), we include all available corrections. Be aware though that the difference between UT1 and UTC, $UT1 - UTC$ is by default put to zero as this difference can only be accurately known afterwards. It is by construction always less than 0.9 s so unless higher timing accuracy than this is needed this is of no concern for us.[†]

As most users might not be familiar with which MJDs to enter to WimpEvent routines are included to instead give calendar dates or times of solstices and equinoxes for any chosen year (which are then calculated on the fly). See below in the input section for more details.

The penalty for using the accurate calculations in SLALIB is an increased CPU time of typical WimpEvent runs of about 15% compared to the simplified timing treatment[‡]. Given this rather small time increase there is really no reason to use the simplified treatment anymore and it is only kept for backward compatibility with codes expecting event files on that form.

3.2.2 Neutrino oscillations

Once the time of the event has been chosen, we calculate the Sun-Earth distance for this time and propagate the event from 1 AU (as it has been left from WimpAnn) to the actual position of the Earth. This means that we perform the vacuum oscillations backwards or forward (which we can do in a consistent way as all the amplitudes and phases are stored in the WimpAnn event files). We do this with the same three-neutrino setup as before.

From the time of the event, we can also calculate the amount of material in the Earth that is passed through. We then use the Preliminary Reference Earth Model (PREM) to

---

[†]For most applications this accuracy should be more than enough. If needed, $UT1 - UTC$ can be found from the International Earth Rotation and Reference Systems Service [10]. $UT1 - UTC$ enters in the routine nupropsu.f as the variable dut1c, which by default is set to zero.

[‡]This is when using the default SLA_EVP routine. With the SLA_EPV routine, WimpEvent runs about 70% slower than in the simplified treatment.

perform a realistic propagation through the Earth to the detector. We of course use the same three-neutrino setup as before to do this in a fully consistent way.

3.2.3   Neutrino interactions at the detector

Once the neutrinos come to the detector we project them out on flavour eigenstates (according to the probabilities given by the amplitudes). We then let the neutrinos interact with nusigma to produce either charged current or neutral current interactions (or both). We are then left with both a charged lepton and a hadronic shower at the interaction place. Note that all angles are with respect to the Sun.[§] We will come back to this issue when discussion the output format below.

Note that the target materials coded into WimpEvent are ice and rock, for which default densities are 1 g/cm^3 and 2.65 g/cm^3 respectively. These densities enter the weights of the events and the range of produced muons. However, they are of no importance when limits are put on the fluxes in a given event file (as then the form only matters). They are of importance, however, when converting between fluxes and annihilation rates or different types of fluxes. In conversion programs, they will thus be of importance and a correction for the actual density is done there.

3.2.4   Inputs

WimpEvent takes the following inputs

- The file name for a WimpAnn event file.

- Latitude of the detector in degrees.

- Longitude of the detector in degrees (new as of WimpSim 3.0).

- Choice between simplified timing treatment (obsolete) or accurate timing treatment with modified Julian dates (MJDs).

- **For simplified timing treatment:** Time interval of simulation, $t = 0$ is at January 1 at 00:00 and $t = 1$ is at December 31 at 24:00. For example, if the detector is located at the South Pole, you would give the time interval as 0.245 and 0.755[¶] if you only want events where the Sun is below the horizon.

- **For accurate timing treatment with MJDs:** The MJDs of starting and ending times of the simulation. These dates can be given explicitly or more generically as follows:

    ```
    'MJD mjd-date',
        e.g. MJD 55927.0d0 for Jan 1, 2012 at 00:00
    'CAL yyyy-mm-dd hh:mm:ss.ss',
        e.g. CAL 2012-01-01 00:00:00.00 for Jan 1, 2012 at 00:00
    'SOW yyyy' for winter (December) solstice of year <yyyy>
    ```

[§]As we now for the accurate timing treatment have the full information about the Sun's position on the sky, it would be straightforward to add e.g. right ascension and declination (or azimuth and elevation) for all particles. It is easily done later by the neutrino telescope Monte Carlos though and to keep the event files small, all angles (except the position of the Sun) are with respect to the Sun. If other positional angles are needed, let me know.

[¶]You might naively think that you should enter the time interval as 0.25 and 0.75 to get the time when the Sun is below the horizon. However, if one takes the eccentricity of the Earth's orbit into account, this time interval has to be increased somewhat.

```
'SOS yyyy' for summer (June) solstice of year <yyyy>
'EQV yyyy' for vernal (March) equinox of year <yyyy>
'EQA yyyy' for autumnal (September) equinox of year <yyyy>
'DMI yyyy' for the date around winter solstice when declination is minimal
'DMA yyyy' for the date around summer solstice when declination is maximal
'DOV yyyy' for the date around vernal equinox when declination = 0
'DOA yyyy' for the date around autumnal equinox when declination = 0
 Example: 'EQV 2012' for vernal equinox of 2012
```

For the equinoxes and solstices, a calculation is performed on the fly to find the dates and times of these (accurate to within a few minutes). Note that equinoxes and solstices are calculated requiring that the ecliptic longitude of the Sun is multiples of 90 degrees. This agrees with the astronomical definition of equinoxes and solstices. The Dxx options instead calculate the equinoxes and solstices requiring that the declination is zero or maximal/minimal. For the equinoxes, the two calculations typically agree to within a minute, but for solstices the difference is a few minutes (due to the longitude of the Sun not being exactly zero). The Dxx options are included as they might be of more interest to a detector at the South Pole, like IceCube.

- A choice if neutral currents, charged currents or both should be simulated. In typical run, only charged currents would be of interest, but if you want to see the impact of events with only hadronic showers, you can include neutral currents as well.

- Target material (currently ice/water or rock)

- The file name for the output event file.

- A directory name where output summary tables should be written.

- The seed for the random number generator.

3.2.5   Internal parameters and arrays of importance

Most of the internal data structures should be self explanatory for the interested code reader.

3.2.6   Outputs – event files

The event files from the Sun will be slightly different depending on if the simplified or accurate timing treatment has been chosen.

Simplified timing treatment. In this case, the event files have the following basic structure

```
# Header with run information (program versions, input parameters etc)
### Event format is as follows:
# EV  number  time-stamp (year)  sun-nadir (degrees)
# Tag  particle-name  energy (GeV)  Sun-angle (deg)  weight
# EE (end of event)
### This is the last line of the wimpevent header
EV        1  0.2710324735149428      0.862502E+02
I nu_tau   0.186928E+03 0.000000E+00 0.353829E-30
O tau-     0.104597E+03 0.379520E+01 0.298568E-42
O hshow    0.832705E+02 -.479190E+01 0.298568E-42
```

```
EE
EV          2  0.9886523207995666        0.113372E+03
I nu_mu~   0.143808E+02 0.000000E+00 0.367730E-30
O mu+       0.106309E+02 0.343235E+01 0.144535E-43
O hshow     0.468948E+01 -.958520E+01 0.144535E-43
EE
```

The last line of the event file is

```
#END
```

Note that the header always ends with the line

```
### This is the last line of the wimpevent header
```

to simplify for other programs reading these event files.

Accurate timing treatment with MJDs. In this case, we store more information with each event (right ascension, declination, azimuth and elevation of the Sun) and instead of the simple time-of-year time stamp, we give the MJD. An event file then has the following structure:

```
# Header with run information (program versions, input parameters etc)
### Event format is as follows:
# EV  number  MJD  Sun-El (deg)  Sun-Az (deg)  Sun-Ra (deg) sun-dec (deg)
# Tag  particle-name  energy (GeV)  Sun-angle (deg)  weight  Az (deg)  El (deg)  Ra (deg)  Dec (deg)
# EE (end of event)
### This is the last line of the wimpevent header
EV        1    56026.19788530647      0.109146E+03 -.772037E+01 0.182233E+02 0.772037E+01
I nu_tau   0.186928E+03 0.000000E+00 0.354440E-30 0.109146E+03 -.772037E+01 0.182233E+02 0.772037E+01
O tau-     0.104597E+03 0.379520E+01 0.299083E-42 0.108489E+03 -.398187E+01 0.175662E+02 0.398187E+01
O hshow    0.832705E+02 -.479190E+01 0.299083E-42 0.109994E+03 -.124392E+02 0.190706E+02 0.124392E+02
EE
EV        2    56015.90608171586      0.214940E+03 -.380443E+01 0.882400E+01 0.380443E+01
I nu_mu    0.699271E+01 0.000000E+00 0.356516E-30 0.214940E+03 -.380443E+01 0.882400E+01 0.380443E+01
O mu-      0.549200E+01 0.121506E+02 0.143447E-43 0.225581E+03 -.981454E+01 0.194644E+02 0.981454E+01
O hshow    0.243899E+01 -.354475E+02 0.143447E-43 0.183875E+03 0.137038E+02 -.222409E+02 -.137038E+02
EE
```

The last line of the event file is

```
#END
```

Note that the header always ends with the line

```
### This is the last line of the wimpevent header
```

to simplify for other programs reading these event files. The event file above is written with full output. There is also an option (`weradec=0`) to omit right ascension and declination for the particles in the event (to keep the event files smaller).

As it is important that each event is kept if you use these files to calculate limits on WIMP annihilations, it is preferable to check that the event files are not corrupted (i.e. all events are there in order and that the file ends with the `#END` token). Also note that nusigma simulations of the neutrino-nucleon scattering are only performed if the neutrino energy are above 1 GeV. Hence, for lower energies, the events will only contain the incoming neutrinos.

In the example above, the header has been truncated compared to a real file, and only the first two events are shown. Each event contains the following information

- event number

- **For simplified timing treatment:** time stamp (in years from Jan 1 00:00 to December 31 at 24:00)

- **For accurate timing treatment with MJDs:** The modified Julian date (MJD).

- **For simplified timing treatment:** the nadir angle (180° - zenith angle) of the Sun at the time of the event

- **For accurate timing treatment with MJDs:** The azimuth$^\parallel$ and elevation of the Sun (degrees) at the time of event for the given latitude and longitude of the detector.

- **For accurate timing treatment with MJDs:** The right ascension and declination of the Sun (degrees) at the time of event. These are given at the epoch of the MJD.

- the incoming neutrino, with energy (in GeV), angle (in degrees) relative to the Sun and a weight. **For accurate timing treatment:** azimuth$^\parallel$and elevation are also given. If `weradec=1`, also right ascension and declination are given (as in the example above).

- the outgoing charged lepton, with energy (in GeV), angle (in degrees) relative to the Sun and a weight. The energy is the total energy (including the rest mass (if applicable) and the kinetic energy). **For accurate timing treatment:** azimuth$^\parallel$and elevation are also given. If `weradec=1`, also right ascension and declination are given (as in the example above).

- the outgoing hadronic shower, with energy (in GeV), angle (in degrees) relative to the Sun and a weight. The energy is the total energy (including the invariant mass and the kinetic energy). **For accurate timing treatment:** azimuth$^\parallel$and elevation are also given. If `weradec=1`, also right ascension and declination are given (as in the example above).

Each particle contains a tag indicating if it is incoming (I) or outgoing (O). If you want to run the neutrinos through your own neutrino-nucleon scattering Monte Carlo, just disregard the outgoing particles. The first angle of the neutrinos, charged leptons and hadronic shower is with respect to the Sun (in degrees). To put the particles into a neutrino telescope Monte Carlo, there are several options

a) read off where the Sun is from the given positional information (or recalculate it from the MJD with your own software) and spread the particles out around that direction according to the Sun-angles in the files above. Please note that (due to momentum conservation) the charged lepton and hadronic shower are in a plane and also note that you have an azimuthal angle around the neutrino direction that you should choose randomly, or

b) use the azimuth and elevation (or right ascension and declination) of the tracks in the event files directly. These ones are already spread around the direction of the Sun and ready to use as-is.

You should also extract MJD and use this in your simulation. Note that if you want fully correct results it is important that the date is the same in your simulations and in WimpEvent

---

$^\parallel$Note that IceCube defines the azimuth a bit differently than standard astronomical definitions. They define is at the angle from 90° east, positive westwards. To get IceCube azimuth from WimpEvent, run with a longitude of 90° and calculate IceCube azimuth as $360° - Az$.

(so that the path length through the Earth and the distance from the Sun to the Earth is correct for the oscillations effects).

Please note that for the hadronic shower, the invariant mass, m_h, is calculated from the kinematical variables of the event, but it is not recorded in the event file. If needed, it can however easily be calculated from requiring that the transverse momentum has to be conserved, i.e. that

$$\sqrt{E_l^2 - m_l^2} \sin \theta_l = -\sqrt{E_h^2 - m_h^2} \sin \theta_h$$

where E_l, θ_l and m_l are the energy, angle and mass of the outgoing lepton and where E_h and θ_h are the energy and angle of the hadronic shower (the minus sign comes from the convention that the hadronic shower angle comes with a minus sign in the event files).

The weights on the events are defined as follows: weight is the normalization of the event such that

$$\Gamma = \sum_{\text{events}} \text{weight} \tag{2}$$

is the volumetric flux of the particle of interest (e.g. muons) in units of particles cm^{-3} annihilation^{-1}. For the incoming neutrinos, the weights are instead on the fluxes, i.e.

$$\Phi = \sum_{\text{events}} \text{weight} \tag{3}$$

is the flux of showers in units of number of particles cm^{-2} annihilation^{-1}. Note, in particular, that the weights include the factor

$$\frac{1}{N_{\text{ann}}}$$

so it is only after you sum all the events that you get the total yield/flux. Please also note that if you have used the option in WimpAnn to split the event files into several files, it is **the number of annihilations in each file** that enter in the weight above. See section 2.8.3 for more details.

3.2.7 Outputs – summary files

If you have run WimpEvent with the option to create summary files to a directory, you get four kinds of files:

- 1D differential (in energy) yields

- 1D integrated (above a given energy threshold) yields

- 2D differential (in energy and angle) yields

- 2D integrated (above a given energy threshold and below a given angle) yields

The general layout of a 1D file is that it contains a header explaining the run info (as for the WimpAnn files) and then follows 6 lines that give the yield dN/dz in 100 bins $z = 0.005, 0.015, \ldots, 0.995$, with z being the relevant energy as fraction of the WIMP mass, $z = E/M_{\text{WIMP}}$. However, the yields are now not only for neutrinos, but also for the corresponding charged leptons and hadronic showers (see more info below) that are created at the neutrino-nucleon interaction. The 2D files contain 50 lines for each neutrino (or charged lepton or hadronic shower) type. Each line is for one of 50 bins in $z = 0.01, 0.03, \ldots, 0.99$. Each line contains 91 values corresponding to $d^2N/dzd\theta$, where θ is binned in 91 bins from 0 to 30° and the meaning of θ is the angle from the center of the Sun or the center of the Earth for Sun files and the direction from the centre of the Earth (the nadir angle) for the Earth files. The θ bins are as follows

- The first 50 bins are for $\theta = 0.1°, 0.3°, \ldots, 9.9°$

- The next 40 bins are for $\theta = 10.25°, 10.75°, \ldots, 29.75°$

- The final bin in θ is for $\theta > 30°$

Furthermore, both the 1D files and the 2D files are available both for the differential yield (as outlined above), but also for the integrated yield up to a given theta and above a given energy. For the integrated yields, the lower limit of each energy bin and the upper limit of each theta bin are used, otherwise, the bin numbering is the same as above.

For each of these, there are five different type of files (indicated by the so called flux index, f[no] in the filename. These different types are

flux index = 1 Neutrino yields for the 6 different neutrino types in table 2. For 1D fluxes, the differential (integrated) flux is given as one line for increasing energies. For 2D fluxes, the differential (integrated) flux is given as one line per angle in the tabulation. These yields are at the detector and the units are cm^{-2} annihilation^{-1} for integrated yields. Differential yields are in addition per z and per degree.

flux index = 2 Same as flux index 1, but for the corresponding charged leptons at the neutrino-nucleon vertex. These yields are at the detector and the units are cm^{-3} annihilation^{-1} for integrated yields. Differential yields are in addition per z and per degree.

flux index = 3 Same as flux index 1, but for the corresponding charged leptons at the detector. Note, as only muons propagate in the detector medium, only μ^- and μ^+ are included in this file. These yields are at the detector and the units are cm^{-2} annihilation^{-1} for integrated yields. Differential yields are in addition per z and per degree.

flux index = 4 Same as flux index 1, but for hadronic showers coming from charged current events originating from the six neutrino types in Table 2. These yields are at the detector and the units are cm^{-3} annihilation^{-1} for integrated yields. Differential yields are in addition per z and per degree.

flux index = 5 Same as flux index 4, but for hadronic showers coming from charged current interactions. These yields are at the detector and the units are cm^{-3} annihilation^{-1} for integrated yields. Differential yields are in addition per z and per degree.

See wimpevent/src/tabsave.f for more information on the actual format in the summary files.

For each of these flux types there are 8 different files (Earth/Sun, 1D/2D, differential or integrated). The naming convention of the files are that they are called:

```
we-m[mass]-ch[channel]-[where]-[dimension]-[diff/int]-f[flux-type].dat
```

where [mass] is the mass in GeV, [channel] is the number of the annihilation channel, [where] is ea or su for the Earth/Sun, [dimension] is 1D or 1D, [diff/int] is diff or int for differential or integrated yields respectively and [flux-type] is the flux type number given above. For example, the differential 1D file for annihilation of 250 GeV WIMPs in the Earth to tau+ tau- (channel 11) is for flux type 2 called

```
we-m250-ch11-ea-1D-diff-f2.dat
```

Note. Please note that the differential yields are per degree and z, i.e. the bin size is included in the yield tables (i.e. they are not just bins filled with events).

These summary files can be added with the script mktab.pl for use in DarkSUSY. Note that adding files like this requires a full simulation according to a specific setup (done with wasetup and wesetup).

3.3 Annihilations in the Earth

For the Earth, we don't need to know either the detector location and time of the event as everything is symmetric for all detectors and time. Hence, we only propagate the neutrinos out from the center to the detector taking neutrino oscillations into account. The fact that the annihilations do not occur exactly in the center of the Earth (the size of the region depends on the mass) is also taken into account (as can be seen on the neutrino angles not being zero).

3.3.1 Output - event files

A typical event file from the Earth looks like

```
# Header containing run information etc
### Event format is as follows:
# EV   number
# Tag  particle-name  energy (GeV)  nadir-angle (deg)  azi-angle (deg)  weight
# EE (end of event)
### This is the last line of the wimpevent header
EV        1
I nu_tau   0.223358E+03 0.229415E+01 0.305915E+03 0.191655E-23
O tau-     0.124982E+03 0.128307E+01 0.153856E+03 0.195255E-35
O hshow    0.993161E+02 0.666976E+01 0.312468E+03 0.195255E-35
EE
EV        2
I nu_mu    0.699271E+01 0.124181E+01 0.864496E+01 0.204634E-23
O mu-      0.545576E+01 0.125813E+02 0.293070E+03 0.823361E-37
O hshow    0.247652E+01 0.348429E+02 0.105801E+03 0.823361E-37
EE
```

The files thus look very similar to the Sun ones, but there are some differences. The main difference is that the events contain no time stamp as this is not needed. The events also contain real angles (with respect to a coordinate system fixed in the detector), where nadir angles (180° - the zenith angle) and the azimuthal angles (define your reference direction for zero azimuthal angle as you wish) of both the neutrinos, charged leptons and hadronic shower are given.

As for the Sun, please also note that if you have used the option in WimpAnn to split the event files into several files, it is **the number of annihilations in each file** that enter in the weight above. See section 2.8.3 for more details.

3.3.2 Output - summary files

These summary files have the same format as for the Sun described above.

4 Scripts

There are a set of scripts available to set up series of WimpAnn and WimpEvent runs. These are located in the `scr` directory. The scripts of most interest are probably:

`wasetup.pl` Script to set up WimpAnn runs. This script creates a directory `<dir>` of your choice with a copy of WimpAnn and a series of run scripts `<dir>/runs` that run WimpAnn for a range of masses from 3 GeV to 10 TeV. The run scripts are put in the `<dir>/runs` directory. Also setup is a run script `<dir>/scr/run-wa.one` to run the run scripts in an automated fashion. This script will rune one of the scripts in `<dir>/runs` at a time and them move them to `<dir>/runs-done`. If you want to change settings, mass ranges, channels etc, change in `scr/wasetup.pl` before running.

`wesetup.pl` Script to set up WimpEvent runs. This script sets up run scripts in a directory `<dir>` created with `wasetup.pl`. The scripts are put in `<dir>/runs`. As with WimpAnn above, a script `<dir>/scr/run-we.one` is available to run one run script at a time until all are gone through. If you want to change masses, channels etc, make sure to edit `scr/wesetup.pl` before running.

`wagenplots.pl` Script to generate plots from all summary files in a WimpAnn run directory. Run after the WimpAnn runs have finished to produce plots of all distributions and results. Calls `wasum2plots.pl`. Requires PAW to run.

`wegenplots.pl` Script to generate plots from all summary files in a WimpEvent run directory. Run after the WimpEvent runs have finished to produce plots of all distributions and results. Calls `wesum2plots.pl`. Requires PAW to run.

`waplots2latex.pl` Script to put all plots generated by `wagenplots.pl` and put them into one (huge) pdf file for easy browsing of results.

`weplots2latex.pl` Script to put all plots generated by `wegenplots.pl` and put them into one (huge) pdf file for easy browsing of results.

`mktab.pl` Script to generate tables for DarkSUSY from WimpEvent summary files. Calls `mktab-one.pl`. On some systems, the Perl engine seems to have trouble running through this massive amount of data and if the script fails, it might be worthwhile to try on another system, or run `mktab-one.pl` manually.

5 Other tools

Distributed with the code is also a few test programs:

`sladate` This program calculates the Sun's position for a given date. As it allows input of dates in many different format, it can also be use to convert between date formats. It can also calculate equinoxes and solstices.

`slatesting` This program is purely for testing the SLALIB interface.

6 Download

The WimpSim package can be downloaded from the page `http://www.fysik.su.se/~edsjo/wimpsim`.

References

[1] J. Edsjö, Master thesis, 1993.

[2] J. Edsjö, PhD thesis, [arXiV: hep-ph/9704384]

[3] L. Bergström, J. Edsjö and P. Gondolo, Phys. Rev. **D55** (1997) 1765. [arXiv: hep-ph/9607237]

[4] L. Bergström, J. Edsjö and P. Gondolo, Phys. Rev. **D58** (1998) 103519. [arXiv: hep-ph/9806293]

[5] J. Edsjo, M. Blennow and T. Ohlsson, [arXiv: 0709.3898]

[6] Nusigma 1.15 J. Edsjö.

[7] J.N. Bahcall, A.M. Serenelli and S. Basu, ApJ **621** (2005) L83.

[8] DarkSUSY. P. Gondolo, J. Edsjö, P. Ullio, L. Bergström, M. Schelke and E.A. Baltz, JCAP **0407** (2004) 008. [arXiv: astro-ph/0406204], darksusy.org.

[9] SLALIB. P.T. Wallace, Starlink User Note 67.70, http://star-www.rl.ac.uk/docs/sun67.htx/sun67.html

[10] International Earth Rotation and Reference Systems Service, http://iers.org.