SOLARPROP: Charge-sign dependent solar modulation for everyone☆

Rolf Kappl

Bethe Center for Theoretical Physics and Physikalisches Institut der Universität Bonn, Nussallee 12, 53115 Bonn, Germany

ARTICLE INFO

ABSTRACT

We present SOLARPROP, a tool to compute the influence of charge-sign dependent solar modulation for cosmic ray spectra. SOLARPROP is able to use the output of popular tools like GALPROP or DRAGON and offers the possibility to embed new models for solar modulation. We present some examples for proton, antiproton and positron fluxes in the light of the recent PAMELA and AMS-02 data.

Program summary

Program title: SOLARPROP

Catalogue identifier: AFAP_v1_0

Program summary URL: http://cpc.cs.qub.ac.uk/summaries/AFAP_v1_0.html

Program obtainable from: CPC Program Library, Queen's University, Belfast, N. Ireland

Licensing provisions: MIT licence (MIT)

No. of lines in distributed program, including test data, etc.: 15347

No. of bytes in distributed program, including test data, etc.: 125635

Distribution format: tar.gz

Programming language: C++ll.

Computer: PC.

Operating system: Linux.

Classification: 1.1, 1.6.

External routines: cfitsio, CCFITS

Nature of problem:
Calculation of the influence on cosmic rays by the heliosphere including drift effects.

Solution method:
Stochastic differential equations.

Additional comments:
Simple interface for text and FITS format input and output.

Running time:
Between a few seconds and a few minutes depending on the physical model.

1. Introduction

The study of high energetic cosmic rays is a well established method to constrain possible decay or annihilation of dark matter to standard model particles. Cosmic rays may either have secondary origin e.g. from spallation of primary particles or originate from primary

---

sources like supernova remnants, pulsars or the decay or annihilation of dark matter. The additional cosmic ray flux from dark matter would increase the measured experimental values at earth. To determine the astrophysical background for cosmic rays as well as their possible fraction originating from dark matter, particles have to be propagated through the galaxy from their origin to the solar system. The propagation of charged cosmic rays is well studied in the literature [1–5] and despite the still large uncertainty, an enormous progress has been made during the last years. As an outcome freely available tools like GALPROP [2,3] and DRAGON [4,5] exist, which simulate the propagation of cosmic rays through the galaxy.

Once, the cosmic rays have reached the solar system, they are further influenced by the solar wind and the magnetic field of the sun [6]. This effect is called solar modulation. It can be approximately taken into account with the so-called force-field method [7]. This approximation is independent of the charge of the cosmic ray particles. The strength of solar modulation depends on the solar activity and thus on the 22 year solar cycle. High solar activity results in a stronger effect and thus a lower cosmic ray flux, whereas lower solar activity results in a smaller effect and a higher cosmic ray flux at the top of the earth's atmosphere.[1] The study of solar modulation is a field of intensive research and many models which are able to provide a better estimate than the force-field approximation for the influence of the heliosphere on cosmic ray transport have been proposed (see [8,9] for recent effective approaches based on the force-field approximation).

Solar modulation can be well described by a Fokker–Planck equation [6]. It has become popular to rewrite this equation to a set of stochastic differential equations (see e.g. [10–16]) which can be solved with Monte Carlo methods. SOLARPROP [17] is a simple Monte Carlo tool which is flexible enough that arbitrary models for solar modulation in the heliosphere can be embedded. It includes several simple models which show the effect of a more realistic model for solar modulation. We outline that charge-sign dependent models nicely fit the experimental data and are an advantage to the force-field approximation. The force-field approximation is based on one free parameter, the so-called Fisk potential ϕ. It serves as a free fit parameter, adjusted a posteriori to model the effect of solar modulation. In contrast, the models in SOLARPROP are based on experimental values of the solar magnetic field. This makes it possible to model solar modulation without a fit parameter but in a predictive framework. For any given date in the past, different experimental values for the solar activity like the tilt angle of the heliospheric current sheet are used to model the effect of solar modulation in a predictive way.

A better understanding of solar modulation is especially interesting for antiprotons as their flux peaks around a kinetic energy of a few GeV, an energy range where the effect of solar modulation is very strong. Due to the low astrophysical background, antiprotons have been used to constrain dark matter models (see e.g. [18] for an analysis of the BESS-Polar II data). A lot of balloon based data are available at a few GeV and a proper treatment of solar modulation can help to decrease the uncertainties on indirect dark matter detection.

SOLARPROP is easy to use and easy to extend with new custom models. The program is able to deal with input in FITS format from GALPROP and DRAGON, as well as with input from text files. The output is provided in the same formats. This makes it possible to include SOLARPROP in the chain of cosmic ray propagation tools. A realistic treatment of cosmic ray propagation from the production in the galaxy to the detection at the top of the earth's atmosphere is thus possible.

2. Physical background

The transport equation for cosmic rays in the heliosphere is given by [6,19]

$$\frac{\partial f}{\partial t} = \nabla \cdot (\kappa \cdot \nabla f) - (\mathbf{V} + \mathbf{V}_D) \cdot \nabla f + \frac{1}{3}(\nabla \cdot \mathbf{V})\frac{\partial f}{\partial \ln p}. \tag{1}$$

f is the particle phase space distribution function, κ the diffusion tensor, $\mathbf{V}$ the solar wind velocity, $\mathbf{V}_D$ the particle drift velocity in the heliospheric magnetic field and p the particle momentum. This Fokker–Planck equation can be rewritten to an equivalent set of stochastic differential equations (SDEs) [20,21]. The idea has been used by many authors [10–16] to solve the propagation of cosmic rays in the heliosphere with Monte Carlo techniques.

The SDEs can be written as

$$dx_i = A_i(x_i)dt + \sum_j B_{ij}(x_i)dW_j \tag{2}$$

where x_i are some coordinates of pseudo-particles, t is the time, A_i is a drift and B_{ij} a diffusion term. W_j is a Wiener process which can be related to a Gaussian distribution dw_j with mean zero and standard deviation of one $N(0, 1)$ by $dW_j = \sqrt{dt}dw_j$. The desired particle phase space distribution f is now obtained by solving the SDEs for a large sample of pseudo-particles. The SDEs for a simple one dimensional example are given by [10]

$$\Delta r = \left(-V + \frac{2\kappa_{rr}}{r}\right)\Delta t + \sqrt{2\kappa_{rr}\Delta t}\,dw_r, \tag{3}$$

$$\Delta T = \frac{2V}{3r}\frac{T^2 + 2Tm}{T + m}\Delta t \tag{4}$$

with radial coordinate r, particle mass m and kinetic energy T. V labels a constant solar wind and κ_{rr} is the energy dependent diffusion constant. The initial condition at the boundary of the heliosphere which we typically set to $r_{\text{Boundary}} = 100$ AU for $t = 0$ is the local interstellar flux. One can solve the SDEs in the so-called forward time approach. One starts with initial discrete kinetic energies T_i and simulates for small time steps Δt the evolution of T_i according to the SDE until either some large finite time or the position of the earth at $r_{\text{Earth}} = 1$ AU is reached. If the earth is reached, the final value $T = T_i$ is recorded. Doing this for many pseudo-particles results in a continuous energy distribution for the same initial energy T_i of pseudo-particles. After normalization to one initial pseudo-particle this

---

[1] With top of the earth's atmosphere we mean the average flux as published by the experimental collaborations, corrected from the influence of the magnetic field of the earth.

gives us a kind of Green's function $G(T, T_i)$ which indicates for every initial energy T_i of some particle the probability to have kinetic energy T after reaching the earth (see e.g. [14] for a more detailed discussion).

SOLARPROP solves the SDEs in a different, so-called backward approach. Pseudo-particles start at the earth and are simulated backward in time until they reach the heliosphere. It has been shown that this approach is usually faster than the forward approach [21,22]. To incorporate the boundary condition, we weight the local interstellar flux $\Phi^{\mathrm{LIS}}(T)$ with the Green's function to obtain the flux at the top of the atmosphere $\Phi^{\mathrm{TOA}}(T)$. The flux is related to the phase space distribution by

$$\Phi^{\mathrm{LIS}}(T) = p^2 f^{\mathrm{LIS}}(T) = (T^2 + 2Tm)f^{\mathrm{LIS}}(T) \tag{5}$$

and we get

$$f^{\mathrm{TOA}}(T) = \sum_i G(T, T_i)f^{\mathrm{LIS}}(T_i), \qquad \Phi^{\mathrm{TOA}}(T) = p^2 f^{\mathrm{TOA}}(T). \tag{6}$$

As the propagation depends on the mass of the cosmic rays, SOLARPROP calculates a Green's function for every given cosmic ray species independently. The main physical input is the definition of the solar magnetic field, the diffusion tensor and the treatment of the drift especially along the heliospheric current sheet (HCS). SOLARPROP has several different built-in models and can also easily extended by the user.

3. Installation

SOLARPROP is freely available for download under [17]. It needs the two packages `cfitsio` and `CCfits` to handle the FITS format. The packages can be downloaded and installed from

http://heasarc.gsfc.nasa.gov/fitsio/

and

http://heasarc.gsfc.nasa.gov/fitsio/ccfits/.

One has to tell SOLARPROP where these packages are installed on the system later during the installation process. After successful installation of the libraries, the following steps are necessary to install SOLARPROP. First, the code has to be extracted in a given favorite directory

```
tar -xvf solarprop.tar.gz
```

Let us assume the packages `cfitsio` and `CCfits` are installed in /usr/local/cfitsio and /usr/local/CCfits. There are two ways to tell SOLARPROP where the packages are. Either through environment variables or during the `configure` step of the installation.

1. Using environment variables the steps are

```
setenv LDFLAGS "-L/usr/local/cfitsio -L/usr/local/CCfits/lib"
setenv CPPFLAGS "-I/usr/local/cfitsio/include \
                -I/usr/local/CCfits/include"
```

for `csh` based systems or

```
export LDFLAGS="-L/usr/local/cfitsio -L/usr/local/CCfits/lib"
export CPPFLAGS="-I/usr/local/cfitsio/include \
                -I/usr/local/CCfits/include"
```

on `bash` like systems. Finally one has to run
```
./configure
```
2. One can instead also use `configure` options and do

```
./configure --with-CCfits=/usr/local/CCfits \
            --with-cfitsio=/usr/local/cfitsio
```

To install SOLARPROP in an individual directory one can use the `-prefix` option like

```
./configure --prefix=/local/solarprop --with-CCfits=/usr/local/CCfits \
            --with-cfitsio=/usr/local/cfitsio
```

After that the next steps are

```
make
make install
```

to install SOLARPROP. The executable should be available in the subdirectory `./bin`. The simple call `./bin/SOLARPROP -v` should result in `This is SOLARPROP version 1.0` if the installation is successful.

4. Usage of SOLARPROP

After successful installation SOLARPROP can be used from the command line. There is no interactive user interface, everything is controlled with command line arguments and a control file. The program modulates a given or built in local interstellar flux and writes the computed top of the atmosphere (TOA) flux to a file. The physical model responsible for solar modulation is selected via a parameter in the control file. The local interstellar flux can be given either by a text or FITS file or by the parametrization of the physical model. For an easy start we discuss different simple use cases.

4.1. Use case 1: Input from GALPROP

GALPROP stores the computed local interstellar flux of all defined cosmic ray species in a FITS file (see Section 4.5 or [23] for a description of the format). This file can directly be used as input for SOLARPROP.

```
./solarprop -c control.dat -o example -i nuclei_54_067f0001 -a angle.dat \
            -n nm.dat
```

The output of SOLARPROP is determined by the keywords in control.dat. An example for a control file is given below:

```
model standard2D
outputFormat Both
modNumber 1
```

If this control file is used, SOLARPROP creates two output files, example.dat with text output and example.fit with the same data as FITS output. The keyword modNumber has the value 1, thus SOLARPROP calculates the modulation of the first cosmic ray species in the input FITS file from GALPROP. All possible control file options are displayed in Tables 2–4.

4.2. Use case 2: Input from DRAGON

DRAGON stores the computed local interstellar flux of the different cosmic ray species in a three dimensional FITS file, in a two dimensional FITS file and in a text file. SOLARPROP is able to handle the two dimensional FITS file. A brief description of the format can be found in Section 4.5. A simple program call would be

```
./solarprop -c control.dat -o example -i run_2D_spectrum.fits -n nm.dat
```

The output of SOLARPROP is determined by the keywords in control.dat. Consider the following control file:

```
model standard2D
outputFormat Both
modNumber 1
polarity 1
angle 30
```

SOLARPROP creates two output files, example.dat with text output and example.fit with the same data as FITS output. The keyword modNumber has the value 1, thus SOLARPROP calculates the modulation of the first cosmic ray species in the FITS file from DRAGON. The solar polarity and the tilt angle of the HCS are manually provided in the control file with the keywords polarity and angle. All possible control file options are displayed in Tables 2–4.

4.3. Use case 3: Input from text file

SOLARPROP can also handle input from text files. In this case the kinetic energy of the nuclei has to be given in GeV in the first column, the flux is expected in the second column. The mass and charge of the cosmic ray nuclei can be provided through the mass and charge keywords in the control file.

```
./solarprop -c control.dat -o example -i textInput.dat -a angle.dat -n nm.dat
```

The output of SOLARPROP is determined by the keywords in control.dat. A possible control file is shown below:

```
model standard2D
outputFormat FITS
mass 0.938
charge 1
```

Here outputFormat is given as FITS, thus SOLARPROP creates one output file example.fit with FITS output. The FITS file consists of two columns. The first one with the kinetic energy and the second one with the modulated flux. The keywords mass and charge specify the cosmic ray species to be protons.

4.4. General use of SOLARPROP

All possible command line arguments of SOLARPROP are given in Table 1. Everything else is controlled by the control file. Every line of the control file is interpreted as key value pair. The first word is a keyword, after a blank, an option should be provided. All keywords are listed in Tables 2 and 3. There exists a default behavior of SOLARPROP for each keyword, for example in case no option is given, or the option is ill-defined. The possible options are given in Table 4.

A general program call is:

```
./solarprop -c control.dat -a angle.dat -n nm.dat -o example -i input.fits
```

The two files angle.dat and nm.dat store the tilt angle and neutron monitor data which are necessary for the model standard2D (see the discussion in Section 5.2). If another model is used, these files are not necessary. If one wants to adjust the result from SOLARPROP, e.g. to find a better agreement with a data set, the options kappaScaling and BfieldScaling can be used. The first one parametrizes

Table 1
Command line options.

Option	Description and argument
-c, --control	Argument is the file name where the control file is located.
-a, --angle	Argument is the file name where the tilt angle file is located. The format should match the definitions from [24].
-o, --output	Argument is the file name where the output should be stored. The file ending is added automatically depending on the option provided in the control file.
-i, --input	Argument is the file name where the input file is located. SOLARPROP automatically determines if the input is in FITS or Text format.
-n, --neutron	Argument is the file name where the file with the neutron monitor data is located. The file should be in the format from [25].
-v, --version	No argument necessary. If this option is chosen SOLARPROP only displays the version number.

Table 2
Control file options (Part 1).

Option	Description
model	Defines the physical model which should be used for solar modulation. Default is ref1.
phi	Force-field value in GV if provided. Result is stored in separate output. Default is no force-field approximation is computed.
modNumber	If provided number of cosmic ray species which should be used for solar modulation. If no value is provided, all cosmic ray species from the input file are modulated. Default is that all species are modulated.
mass	Mass of the cosmic ray species can be manually given. Is only considered if input is in Text format or no input is provided. Default is 0.938 GeV.
charge	Charge of the cosmic ray species can be manually given. Is considered if input is in Text format or no input is provided. Default is 1.
tiltModel	Defines if R model or L model for the tilt angle should be used. Default is R model. Option is only used if the command line option -a, --angle is used.
year	Defines the start year for the determination of the tilt angle and the solar polarity. Average over the whole year is used if no month is given. Default is 2014.
month	Defines the start month for the determination of the tilt angle and the solar polarity. Average over the month is used if yearEnd and monthEnd are not provided.
yearEnd	Defines the end year for the determination of the tilt angle. Is considered if also monthEnd is provided.
monthEnd	Defines the end month for the determination of the tilt angle. Is considered if also yearEnd is provided. The average for the time period between month.year and monthEnd.yearEnd is determined.

Table 3
Control file options (Part 2).

Option	Description
polarity	Defines solar polarity manually. Default is 1. Option is only used if the command line option -a, --angle is not used.
angle	Defines the tilt angle manually. Default is 0. Option is only used if the command line option -a, --angle is not used.
outputFormat	Defines the format for the output file. Default is Text.
kappaScaling	Normalization of the diffusion tensor κ can be manually adjusted. Default is 1.
BfieldScaling	Normalization of the magnetic field B can be manually adjusted. Default is 1.
index	Spectral index for output rescaling. Output flux is rescaled with T^{index}. Default is no rescaling 0.
dt	Used time step Δt can be adjusted manually. Be careful with this option. A too large time step increases the speed, but results in wrong results. A too small time step results in a very high computing time. Default depends on the chosen model.
extraBins	Number of extra bins which should be added between the given local interstellar spectrum data points. Computed bins are only added if the given data points from the input file are not dense enough. Default is 5.
total	Number of pseudo-particles which are modulated for every energy bin. Default is 1000.
nmValue	Value for the diffusion tensor normalization in the standard2D model. Default is 500. Option is only used if the model standard2D is selected and command line option -n, --neutron is not used.

the normalization of the diffusion, whereas the second one parametrizes the strength of the solar magnetic field. A larger value for kappaScaling e.g. 1.5 makes the diffusion more efficient and results in a higher top of the atmosphere flux.

If the option phi is given in the control file, SOLARPROP also calculates the result for the force-field approximation. This result is stored in a file with the name exampleForce.dat or exampleForce.fit depending on the value for outputFormat.

4.5. Input and output of SOLARPROP

The local interstellar input flux for SOLARPROP can be provided through the command line option -i in three different ways.

1. If no input file is provided, SOLARPROP tries to find a reference local interstellar flux provided by the chosen model. If that is not successful, a warning is displayed and a vanishing local interstellar flux is assumed. In that case the output is useless.
2. The local interstellar flux can be imported from a text file. The first column is assumed to contain the kinetic energy in GeV and the second column the local interstellar flux in $m^{-2}\,sr^{-1}\,s^{-1}\,GeV^{-1}$. The mass and charge of the cosmic ray species have to be given in the control file.
3. If a FITS file is given, SOLARPROP checks automatically if the file is in GALPROP or DRAGON like format. Other formats are not supported. In case a modNumber is given, only the desired cosmic ray species is modulated. If no modNumber is found, SOLARPROP modulates all cosmic ray species found in the import file.

Table 4

Viable values for the keywords in the control file as described in Tables 2 and 3.

Keyword	Allowed parameters
model	ref1, ref2, ref3, ref4, standard2D or custom.
phi	Every numerical value. Unit is GV.
modNumber	Integer value. Value should not be higher than the number of cosmic ray species in the input FITS file.
mass	Every numerical value. Unit is GeV.
charge	Integer value.
tiltModel	L or R.
year	Four digit integer value.
month	One or two digit integer value.
yearEnd	Four digit integer value.
monthEnd	One or two digit integer value.
polarity	1 or −1.
angle	Every numerical value between 0 and 90. Unit is degree.
outputFormat	FITS, Text or Both.
kappaScaling	Every numerical value.
BfieldScaling	Every numerical value.
index	Every numerical value.
dt	Integer value.
extraBins	Integer value.
total	Integer value.
nmValue	Every numerical value.

The output of SOLARPROP is stored in files whose names are provided through the option -o. The output depends on the keywords modNumber and outputFormat and on the format of the input. We can distinguish between two cases.

1. If the input is from a text file or the local interstellar flux is from an internal reference model, independent of the modNumber only one cosmic ray species is modulated as only one is provided. In the case of text file output, in the first column the kinetic energy in GeV is written. In the second column the top of the atmosphere flux is provided in units of m^{-2} sr^{-1} s^{-1} GeV^{-1} (if index is 0). The energy bins are always the same as the input ones from the local interstellar flux.

   In case of FITS output, two keywords, namely mass and charge to store the cosmic ray properties are added to the pHDU. The first column of the pHDU consists of the kinetic energy in GeV and the second column stores the calculated flux in m^{-2} sr^{-1} s^{-1} GeV^{-1} (if index is 0).

2. If the input is from a FITS file, depending on the value of the keyword modNumber one or all cosmic ray species in the file are modulated. All species are present in the output. If the species is not modulated the local interstellar flux is added to the output. In the case of text output, the file consists of one column with the kinetic energy in GeV and additional columns, one for the flux of each cosmic ray species in m^{-2} sr^{-1} s^{-1} GeV^{-1} (if index is 0).

   For FITS output, SOLARPROP uses the same format as given by the input. That means, if the input is in GALPROP like format the fluxes are stored, each in a separate column in the pHDU, whereas for a DRAGON like format every flux is stored in its own HDU. Please note that the unit of the stored fluxes in both cases is changed to the SOLARPROP standard m^{-2} sr^{-1} s^{-1} GeV^{-1}!

4.6. Inclusion of custom models

SOLARPROP can be easily extended by the user. The main classes for the physics of particle propagation are the interface (abstract base class) Iparticle.h, the derived abstract class particle.cc and its concrete implementations like standard2D.cc. If one wants to include an own model, it should inherit from particle.cc. Of course, all virtual methods of particle.cc have to be implemented and a constructor has to be added. The method void calculate() manages the computation of all quantities which are needed several times in a propagation step. This avoids unnecessary double computations. A basic implementation can already be found in particle.cc.

SOLARPROP already offers two files for the concrete implementation of a custom model, custom.h and custom.cc. The implemented model can be called from the control file with the name custom as described for the validation in import.h.

5. Examples and validation

5.1. Validation

We validate SOLARPROP against several models available in the literature. This approach has also been used previously to test Monte Carlo based solar modulation calculations [14]. A comparison of the result from SOLARPROP with the result of [10] and [26] is shown in Fig. 1. In the left panel a one dimensional model with plain diffusion and a diffusion coefficient $\kappa = 5 \cdot 10^{22}$ cm^2 s^{-1} GV^{-1} is used (matches figure 4 in [10]). The model is implemented for reference in SOLARPROP as model ref1. The corresponding SDEs can be found in Appendix A.1.1. The control file to reproduce this result is very simple:

```
model ref1
mass 0.938
charge 1
```

The corresponding program call is:

```
./solarprop -c control.dat -o example
```
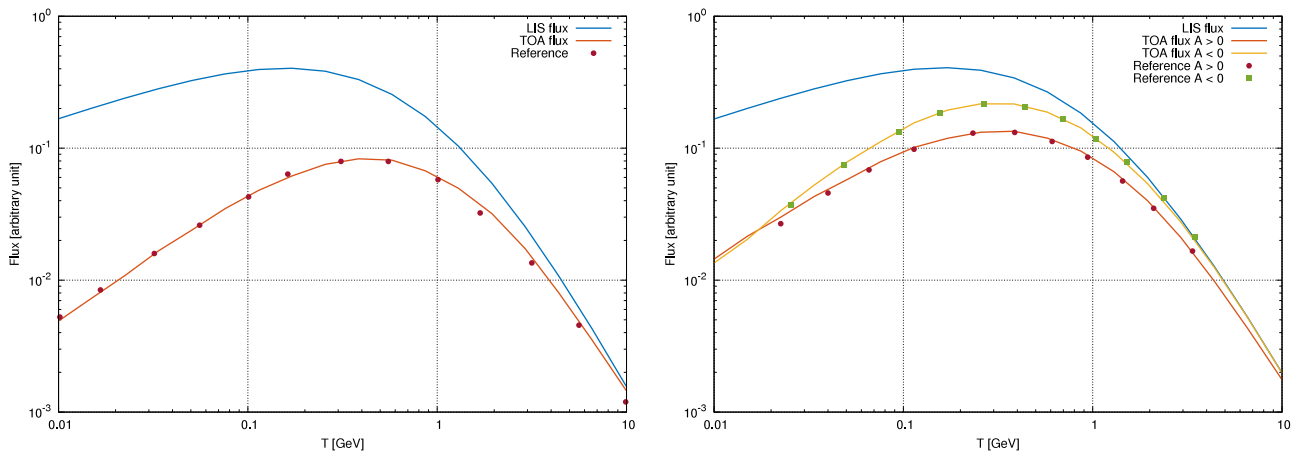
Fig. 1. Validation of `SOLARPROP` with the reference model `ref1` from [10] (left panel) and the model `ref2` from [26] (right panel).
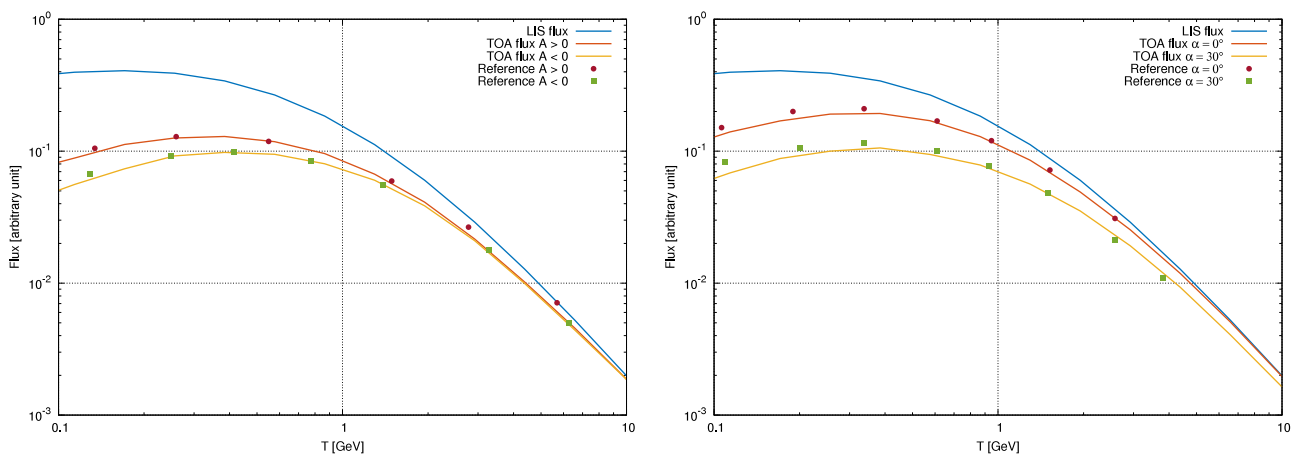

Fig. 2. Validation of `SOLARPROP` with the reference model `ref3` from [27] (left panel) and the model `ref4` from [28] (right panel).

In the right panel a two dimensional model without tilt angle dependence (flat HCS) is used. The parameters match the ones in table 1 and figure 2 of [26]. The model can be used in `SOLARPROP` with parameter `ref2` and the SDEs can be found in Appendix A.2.1. For negative polarity $A < 0$ the control file to reproduce the figure is:

```
model ref2
mass 0.938
charge 1
polarity -1
```

A comparison of more sophisticated models is displayed in Fig. 2. In the left panel a two dimensional model with a possibility to model a wavy HCS is used [27]. The plot displays a comparison between `SOLARPROP` and the result from figure 6 of [27]. The SDEs can be found in Appendix A.2.2. The corresponding control file for positive polarity $A > 0$ is:

```
model ref3
mass 0.938
charge 1
polarity 1
```

In the right panel the result of figure 5 from [28] is reproduced for two different values of the tilt angle α. The model is also two dimensional and able to describe a wavy HCS. As can be seen in Appendix A.2.3 the drifts are described in a different way than in [27]. The following options have been used to produce the result for the case $A < 0$ and $\alpha = 30°$.

```
model ref4
mass 0.938
charge 1
polarity -1
angle 30
```

The models of Fig. 2 are available in `SOLARPROP` under the name `ref3` and `ref4` respectively. The results from Figs. 1 and 2 nicely show that our Monte Carlo approach is viable and different models of the heliosphere can be easily implemented in `SOLARPROP`.
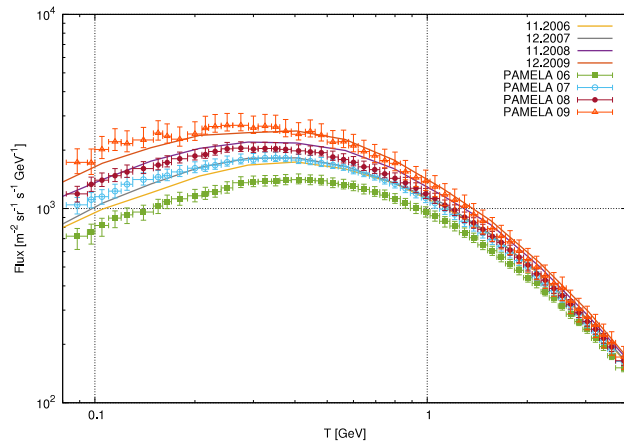
Fig. 3. Data for cosmic ray protons at different solar activities measured by PAMELA [29]. The result from `SOLARPROP` with the model `standard2D` is displayed as solid lines. The result is in good agreement, except for the data from 2006.

Table 5
Different solar activity dependent parameters as measured by [24,31–33] for a time of negative solar polarity $A < 0$. The increasing NM count rate indicates a decreasing solar activity.

Time	α_L	α_R	SSN	NM count rate (Newark)
11.2006	29.5	11.25	12.6	3539
12.2007	31.267	15.8	4.9	3591
11.2008	26	9.85	1.7	3678
12.2009	28.3	13.55	8.3	3732

5.2. Examples

We want to introduce in this section another model which is implemented in `SOLARPROP` and show its result with different data sets. This more sophisticated model, able to describe recent cosmic ray data is implemented in `SOLARPROP` under the name `standard2D`. The model is still very simple, but has several time dependent parameters to accurately describe recent data. A more complicated model, e.g. a three dimensional one, can easily be added by the user or may be part of the next version of `SOLARPROP`. A challenging data set for cosmic ray protons released by the PAMELA experiment [29] is displayed in Fig. 3. The proton flux increases with time which indicates a decreasing solar activity. This data set has been successfully described by solar modulation models [16,30]. These studies have shown that a simple model, where only one parameter (e.g. the tilt angle) is time dependent is insufficient to describe the data. This excludes all our discussed reference models. In Table 5 several measured solar activity dependent parameters are displayed. To accurately describe experimental data a tilt angle and sunspot number (SSN) dependent model was used in [12]. In [16,30] many more parameters, like the overall magnetic field strength have been adjusted. We use a tilt angle dependent model and vary in addition the normalization of the diffusion tensor κ_0, taking into account a more or less effective diffusion.

The variation of κ_0 is based on the calculations of [34,35] which average different neutron monitor (NM) count rate data. In [34,35] a force-field potential ϕ_{Usoskin} was determined from the neutron monitor data to describe charge-sign independent solar modulation. We relate the diffusion tensor normalization κ_0 to this quantity ϕ_{Usoskin}. By not directly using the neutron monitor data we avoid to redo the average procedure, already performed by [34,35]. We use the simple relation

$$\kappa_0 = \tilde{\kappa}_0 \begin{cases} \dfrac{137}{\phi_{\text{Usoskin}}} - 0.061, & qA < 0 \\ \dfrac{7}{100}\dfrac{137}{\phi_{\text{Usoskin}}} - 0.061, & qA > 0 \end{cases} \tag{7}$$

to model a solar activity dependent diffusion tensor normalization. $\tilde{\kappa}_0$ is just the usual diffusion constant. The formula depends on the polarity of the solar cycle A and the charge of the cosmic ray species q to take into account that drift effects have not been considered in [34,35]. This idea is also motivated by the approach in [12] where the normalization of the diffusion tensor is polarity and charge dependent. The detailed implementation of the drift effects is described in Appendix A.2.4. To reproduce the results in Fig. 3 e.g. for the date 11.2007 the control file is:

```
model standard2D
month 11
year 2007
mass 0.938
charge 1
```

Except for the data from 2006 the model `standard2D` is in good agreement with the data. To describe all four data sets a more complicated model is necessary.
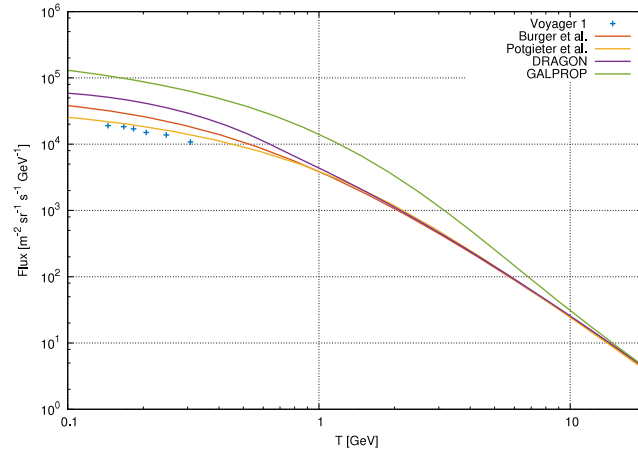
Fig. 4. Voyager 1 measurements for protons at 122 AU [36] and different popular local interstellar proton fluxes [16,37] including results from `GALPROP` and `DRAGON`. The fluxes from `GALPROP` and `DRAGON` are of course model dependent.
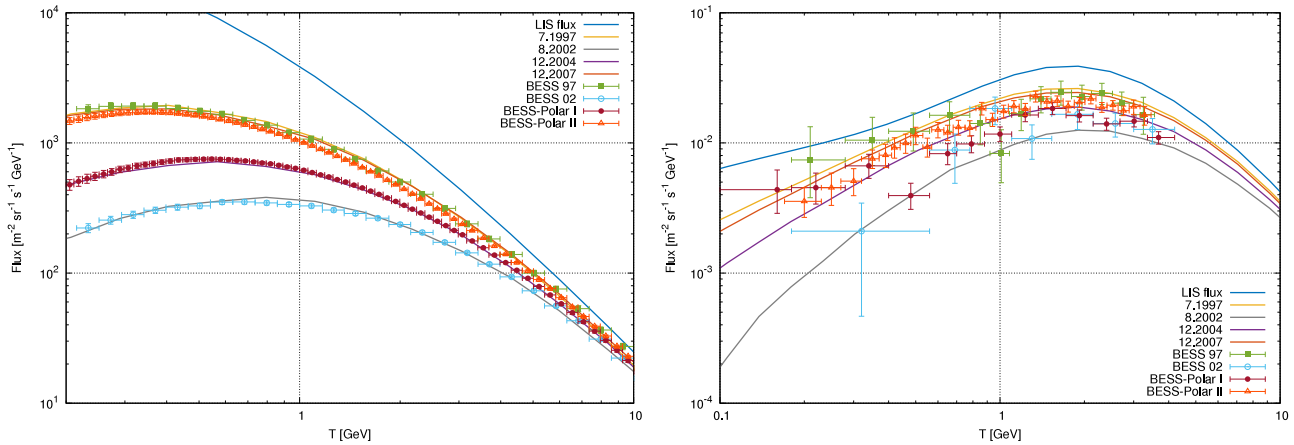

Fig. 5. Data for cosmic ray protons and antiprotons at different solar activities measured by BESS 97, BESS 2002 [38–40] and BESS-Polar I, BESS-Polar II [41–43].

We want to point out that the concrete form of a successful model for solar modulation strongly depends on the local interstellar flux. Obviously, a different shape of the diffusion tensor or a different approach to model the drift effects is possible if the local interstellar flux is adjusted. As the Voyager 1 spacecraft has recently passed the heliosphere and entered the heliopause [36], the local interstellar flux for some cosmic ray species is now measured directly for the first time. We show this data in Fig. 4 together with different local interstellar proton fluxes. We take the local interstellar proton flux from [37] as also used in [12,34,35] (notice the comment in [34] about the different forms of the equation). As Fig. 4 indicates, the flux from [16] seems to be more appropriate. Because we use results from [34,35], we restrict ourselves to the proton flux from [37]. This local interstellar flux is used for the result displayed in Fig. 3.

In Fig. 5 the result for cosmic ray modulation with the `standard2D` model of `SOLARPROP` and data from the BESS and BESS-Polar experiments [38–43] are displayed. The local interstellar flux for antiprotons is determined with recent results from the literature. The galactic propagation setup described in [44] together with the recently updated antiproton production cross sections [45] and the proton flux from [37] is used. Emphasizing that the model uses no fit parameters it describes the data very well. The only input is the time when the experiment took place. The result slightly overshoots the data from the BESS-Polar II experiment which took data in December 2007. As there is a deviation between the PAMELA proton data from the same time period and the BESS-Polar II data we do not worry about this result.

One can also describe leptons as can be seen in Fig. 6. This data (especially the positron fraction) has also been discussed in [49] as indication for charge-sign dependent solar modulation. The local interstellar positron flux seems to include an exotic primary contribution at higher energies, known as the positron excess (see e.g. [50] for a recent discussion of the possible primary positron contribution).

For the parametrization we use the secondary positron flux from `DRAGON` $\Phi^{\text{LIS}}_{\text{sec,DRAGON}}(T)$ and add a power law contribution $\Phi^{\text{LIS}}_{\text{prim}}(T) = 3.7 \cdot T^{-2.7}$ to model the primary contribution. This flux choice $\Phi^{\text{LIS}}_{e^+}(T) = \Phi^{\text{LIS}}_{\text{sec,DRAGON}}(T) + \Phi^{\text{LIS}}_{\text{prim}}(T)$ is in agreement with the high energy data by PAMELA [47] and AMS-02 [48]. The result for solar modulation with this local interstellar flux and the model `standard2D` is in poor agreement with the data. This shows that we need a refined model to describe also solar modulation for leptons in a reasonable way or the local interstellar flux for positrons is different. It has recently been shown that more sophisticated models [51] can well describe new electron data by the PAMELA experiment [52]. The inclusion of such models which seem to be able to describe solar modulation for leptons in `SOLARPROP` is left for future work. To reproduce e.g. the theoretical estimate for the positron flux of the AMS-02 experiment displayed in Fig. 6 the control file is:
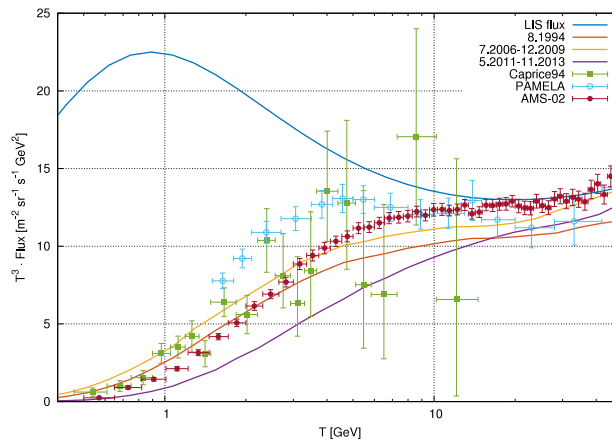
Fig. 6. Data for cosmic ray positrons at different solar activities measured by Caprice [46], PAMELA [47] and AMS-02 [48]. The model `standard2D` is in poor agreement with the data. The question if this indicates that the solar modulation model is too simple or the local interstellar positron flux is different is left for future work.

```
model standard2D
year 2011
month 5
yearEnd 2013
monthEnd 11
mass 0.000511
charge 1
index 3
```

together with the input flux as described above.

The examples in this section show that the simple `standard2D` model implemented in `SOLARPROP` is able to describe charge-sign dependent solar modulation. More sophisticated models which better describe the data can be implemented by the user.

6. Conclusions

We have introduced a tool called `SOLARPROP` to simulate charge-sign dependent solar modulation for cosmic rays. This is an advantage to the commonly used force-field approximation, as also drift effects which are charge-sign dependent are taken into account. We have validated the tool against several simple models from the literature, to show that the software works as expected. A more sophisticated two dimensional model has also been implemented. The only required input for this model is a date or a time period. With this information and a local interstellar flux, `SOLARPROP` calculates the top of the atmosphere flux. The results have been compared with several proton and antiproton data sets from BESS, BESS-Polar and PAMELA and are in a good agreement. The comparison with positron data from PAMELA and AMS-02 is more challenging. We find some tension with the data, which probably show that the implemented model is too simple to account for solar modulation for leptons. On the other hand also the local interstellar flux for positrons is rather uncertain due to its possible primary component.

`SOLARPROP` has the advantage that it can be easily extended by new models. The implementation of custom models will make it possible to test further models for leptons. This is left for future work. Another advantage of `SOLARPROP` is its FITS interface. With this interface, output from popular tools for the propagation of cosmic rays in the galaxy like `GALPROP` and `DRAGON` can directly be used as input for `SOLARPROP`. In the past, the free Fisk potential in the force-field method was often degenerate with fit parameters from galaxy propagation of cosmic rays like the strength of reacceleration. We hope that `SOLARPROP` can help to break this degeneracy and will lead in the combination with a detailed study of cosmic ray transport in the galaxy to a smaller propagation uncertainty at GeV kinetic energies. This is important for indirect dark matter detection with antiprotons and positrons and will hopefully be useful in the near future.

Appendix. Description of the models

We summarize the different SDEs which are implemented for the different models in `SOLARPROP`. The rigidity for a particle with momentum p, mass m and charge Z is defined as

$$\mathcal{R} = \frac{p}{|Z|} = \frac{\sqrt{T^2 + 2Tm}}{|Z|}. \tag{8}$$

A.1. One dimensional models

A.1.1. Model $\mathtt{ref1}$

The one dimensional model $\mathtt{ref1}$ [10] is described by

$$\Delta r = \left(-V + \frac{2\kappa_{rr}}{r}\right)\Delta t + \sqrt{2\kappa_{rr}\Delta t}\,dw_r, \tag{9}$$

$$\Delta T = \frac{2V}{3r}\frac{T^2 + 2Tm}{T + m}\Delta t. \tag{10}$$

The solar wind is assumed to be constant $V = 400$ km s^{-1} and dw_j is a Gaussian distribution of random numbers with mean zero and standard deviation of one $N(0, 1)$. The radial part of the diffusion tensor is given by

$$\kappa_{rr} = \kappa_0 \beta \mathcal{R} = \kappa_0 \beta \frac{\sqrt{T^2 + 2Tm}}{|Z|}. \tag{11}$$

β is the particle speed which can be linked to the kinetic energy.

A.2. Two dimensional models

We assume for all models that the heliospheric magnetic field is described by a Parker spiral (see [30] for a recent discussion of different magnetic fields)

$$\mathbf{B} = \frac{A}{r^2}(\mathbf{e_r} - \tan\psi\,\mathbf{e_\phi})\left(1 - 2H\left(\vartheta - \frac{\pi}{2}\right)\right), \qquad B = \frac{|A|}{r^2}\sqrt{1 + \tan^2\psi}. \tag{12}$$

A is a constant taking care that the magnetic field at the position of the earth matches its measured value ≈ 5 nT and the correct polarity of the solar cycle. We thus have usually $A \approx \pm 3.4$ nT AU^2. The factor $\pi/2$ in the Heaviside function is strictly valid only for a flat HCS and has to be corrected otherwise. The spiral angle ψ is defined as

$$\tan\psi = \Gamma = \frac{\Omega r \sin\vartheta}{V} \tag{13}$$

where $\Omega = 2.866 \cdot 10^{-6}$ rad s^{-1} is the average angular rotation of the sun. The radial and angular part of the diffusion tensor can be rewritten [53]

$$\kappa_{rr} = \kappa_\| \cos^2\psi + \kappa_\perp \sin^2\psi = \frac{1}{1 + \Gamma^2}(\kappa_\| + \kappa_\perp \Gamma^2), \tag{14}$$

$$\kappa_{\vartheta\vartheta} = \kappa_\perp. \tag{15}$$

For the SDEs we find

$$\Delta r = \left(-V - V_{\mathrm{D},r} - V_{\mathrm{HCS},r} + \frac{1}{r^2}\frac{\partial r^2 \kappa_{rr}}{\partial r}\right)\Delta t + \sqrt{2\kappa_{rr}\Delta t}\,dw_r, \tag{16}$$

$$\Delta\vartheta = \left(-\frac{V_{\mathrm{D},\vartheta}}{r} + \frac{1}{r^2 \sin\vartheta}\frac{\partial \sin\vartheta \kappa_{\vartheta\vartheta}}{\partial\vartheta}\right)\Delta t + \frac{\sqrt{2\kappa_{\vartheta\vartheta}\,\Delta t}}{r}dw_\vartheta, \tag{17}$$

$$\Delta T = \frac{2V}{3r}\frac{T^2 + 2Tm}{T + m}\Delta t. \tag{18}$$

A.2.1. Model $\mathtt{ref2}$

The two dimensional model $\mathtt{ref2}$ [26] is described by a r independent diffusion tensor

$$\kappa_\| = \kappa_0 \beta \sqrt{\mathcal{R}} = \kappa_0 \beta \frac{(T^2 + 2Tm)^{\frac{1}{4}}}{\sqrt{|Z|}}, \tag{19}$$

$$\kappa_\perp = 0.1\kappa_\| \tag{20}$$

which results in

$$\frac{1}{r^2}\frac{\partial r^2 \kappa_{rr}}{\partial r} = \frac{2}{r}\left(\kappa_{rr} + (\kappa_\perp - \kappa_\|)\frac{\Gamma^2}{(1 + \Gamma^2)^2}\right), \tag{21}$$

$$\frac{1}{r^2 \sin\vartheta}\frac{\partial \sin\vartheta \kappa_{\vartheta\vartheta}}{\partial\vartheta} = \frac{\kappa_\perp}{r^2}\cot\vartheta. \tag{22}$$

The HCS is assumed to be flat. To avoid a singular drift velocity due to the flat HCS we use a regularization proposed in [54]. In [26] a different approach is used, but our validation shows that the result does not depend on the details of the regularization. The drift velocities

are given by

$$V_{D,r} = q\frac{2\beta r}{3A}\sqrt{T^2 + 2Tm}\cot\vartheta\frac{\Gamma}{(1+\Gamma^2)^2}\left(1 - 2H\left(\vartheta - \frac{\pi}{2}\right)\right), \tag{23}$$

$$V_{D,\vartheta} = q\frac{2\beta r}{3A}\sqrt{T^2 + 2Tm}\frac{\Gamma(2+\Gamma^2)}{(1+\Gamma^2)^2}\left(1 - 2H\left(\vartheta - \frac{\pi}{2}\right)\right), \tag{24}$$

$$V_{HCS,r} = \begin{cases} -q\dfrac{A\beta}{|A|}\dfrac{\Gamma}{\sqrt{1+\Gamma^2}}\left(0.457 - 0.412\dfrac{d}{R_L} + 0.0915\dfrac{d^2}{R_L^2}\right), & d < 2R_L \\ 0, & d \geq 2R_L \end{cases} \tag{25}$$

where the Larmor radius R_L and the distance to the HCS d are given by

$$R_L = \frac{\mathcal{R}}{B} = \frac{\sqrt{T^2 + 2Tm}}{|Z|}\frac{r^2}{|A|\sqrt{1+\Gamma^2}}, \tag{26}$$

$$d = |r\cos\vartheta|. \tag{27}$$

A.2.2. Model ref3

Another two dimensional model is the one named ref3 [27]. We find for the diffusion tensor

$$\kappa_\parallel = \begin{cases} \kappa_{0\parallel}\beta\sqrt{0.4}(1+r^2), & \mathcal{R} < 0.4\,\text{GV} \\ \kappa_{0\parallel}\beta\sqrt{\mathcal{R}}(1+r^2) = \kappa_{0\parallel}\beta\dfrac{(T^2+2Tm)^{\frac{1}{4}}}{\sqrt{|Z|}}(1+r^2), & \mathcal{R} \geq 0.4\,\text{GV}, \end{cases} \tag{28}$$

$$\kappa_\perp = \kappa_{0\perp}\beta\mathcal{R}\frac{r^2}{\sqrt{1+\Gamma^2}} = \kappa_{0\perp}\beta\frac{\sqrt{T^2+2Tm}}{|Z|}\frac{r^2}{\sqrt{1+\Gamma^2}} \tag{29}$$

which gives us

$$\frac{1}{r^2}\frac{\partial r^2\kappa_{rr}}{\partial r} = \frac{2}{r}\left(\kappa_{rr} + (\kappa_\perp - \kappa_\parallel)\frac{\Gamma^2}{(1+\Gamma^2)^2} + \frac{r^2}{1+r^2}\frac{\kappa_\parallel}{1+\Gamma^2} + \kappa_\perp\frac{\Gamma^2(2+\Gamma^2)}{2(1+\Gamma^2)^2}\right), \tag{30}$$

$$\frac{1}{r^2\sin\vartheta}\frac{\partial\sin\vartheta\kappa_{\vartheta\vartheta}}{\partial\vartheta} = \frac{1}{1+\Gamma^2}\frac{\kappa_\perp}{r^2}\cot\vartheta. \tag{31}$$

The model simulates a wavy HCS and introduces a smooth function f and its derivative f' to model the HCS drifts. We find

$$f = \frac{1}{\alpha_{HCS}}\arctan\left(\left(1 - \frac{2\vartheta}{\pi}\right)\tan\alpha_{HCS}\right), \tag{32}$$

$$f' = -\frac{2}{\pi\alpha_{HCS}}\frac{\tan\alpha_{HCS}}{1 + \left(1 - \frac{2\vartheta}{\pi}\right)^2\tan^2\alpha_{HCS}}, \tag{33}$$

$$\alpha_{HCS} = \arccos\left(\frac{\pi}{2\vartheta_{1/2}} - 1\right) \tag{34}$$

where $\vartheta_{1/2}$ is a reference value for the waviness of the HCS. The drift velocities are

$$V_{D,r} = q\frac{2\beta r}{3A}\sqrt{T^2 + 2Tm}\cot\vartheta\frac{\Gamma}{(1+\Gamma^2)^2}f, \tag{35}$$

$$V_{D,\vartheta} = q\frac{2\beta r}{3A}\sqrt{T^2 + 2Tm}\frac{\Gamma(2+\Gamma^2)}{(1+\Gamma^2)^2}f, \tag{36}$$

$$V_{HCS,r} = q\frac{\beta r}{3A}\sqrt{T^2 + 2Tm}\frac{\Gamma}{1+\Gamma^2}f'. \tag{37}$$

A.2.3. Model ref4

The last two dimensional reference model implemented in SOLARPROP is model ref4 [28]. The diffusion tensor agrees with the one of model ref3. The drift effects are modeled in a different way. The regular part agrees with the one from model ref2 and is thus given by

$$V_{D,r} = q\frac{2\beta r}{3A}\sqrt{T^2 + 2Tm}\cot\vartheta\frac{\Gamma}{(1+\Gamma^2)^2}\left(1 - 2H\left(\vartheta - \frac{\pi}{2}\right)\right), \tag{38}$$

$$V_{D,\vartheta} = q\frac{2\beta r}{3A}\sqrt{T^2 + 2Tm}\frac{\Gamma(2+\Gamma^2)}{(1+\Gamma^2)^2}\left(1 - 2H\left(\vartheta - \frac{\pi}{2}\right)\right). \tag{39}$$

The model simulates a wavy HCS with tilt angle α by [55]

$$\vartheta_\Delta = \frac{2\mathcal{R}V}{|A|\Omega\cos\alpha} = \frac{2\sqrt{T^2 + 2Tm}V}{|A||Z|\Omega\cos\alpha} \tag{40}$$

and

$$V_{HCS,r} = \begin{cases} 2\sin\vartheta \dfrac{\Gamma}{\sqrt{1+\Gamma^2}} \dfrac{\beta}{6} \dfrac{\vartheta_\Delta \cos\alpha}{\sin(\alpha+\vartheta_\Delta)}, & \dfrac{\pi}{2} - \alpha - \vartheta_\Delta < \vartheta < \dfrac{\pi}{2} + \alpha + \vartheta_\Delta \\ 0, & \text{else.} \end{cases} \tag{41}$$

A.2.4. Model standard2D

The model `standard2D` incorporates a tilt angle dependent wavy HCS and is tuned against recent cosmic ray data. The diffusion tensor is

$$\kappa_\parallel = \begin{cases} \kappa_0 \dfrac{|A|}{3B}\beta \cdot 0.1 = \kappa_0 \beta \dfrac{1}{30} \dfrac{r^2}{\sqrt{1+\Gamma^2}}, & \mathcal{R} < 0.1 \text{ GV} \\ \kappa_0 \dfrac{|A|}{3B}\beta\mathcal{R} = \kappa_0 \beta \dfrac{\sqrt{T^2+2Tm}}{3|Z|} \dfrac{r^2}{\sqrt{1+\Gamma^2}}, & \mathcal{R} \geq 0.1 \text{ GV}, \end{cases} \tag{42}$$

$$\kappa_\perp = 0.02\kappa_\parallel \tag{43}$$

which results in

$$\frac{1}{r^2}\frac{\partial r^2 \kappa_{rr}}{\partial r} = \frac{2}{r}\left(\kappa_{rr} + (\kappa_\perp - \kappa_\parallel)\frac{\Gamma^2}{(1+\Gamma^2)^2} + \kappa_{rr}\frac{(2+\Gamma^2)}{2(1+\Gamma^2)} \right), \tag{44}$$

$$\frac{1}{r^2\sin\vartheta}\frac{\partial \sin\vartheta \kappa_{\vartheta\vartheta}}{\partial \vartheta} = \frac{1}{1+\Gamma^2}\frac{\kappa_\perp}{r^2}\cot\vartheta. \tag{45}$$

The normalization of the diffusion tensor κ_0 is time dependent (see also Eq. (7)). For the drift effects the model is inspired by [12] and thus similar to model `ref3`. The only difference is that $\vartheta_{1/2}$ is related to the tilt angle of the current sheet α by [28]

$$\vartheta_{1/2} = \frac{\pi}{2} - \frac{1}{2}\sin\left(\alpha + \frac{2R_L}{r}\right) = \frac{\pi}{2} - \frac{1}{2}\sin\left(\alpha + \frac{\sqrt{T^2+2Tm}}{|Z|}\frac{2r}{|A|\sqrt{1+\Gamma^2}}\right). \tag{46}$$

References

[1] D. Maurin, F. Donato, R. Taillet, P. Salati, Astrophys. J. 555 (2001) 585–596. astro-ph/0101231.
[2] GALPROP, http://galprop.stanford.edu.
[3] A.W. Strong, I.V. Moskalenko, Astrophys. J. 509 (1998) 212–228. astro-ph/9807150.
[4] DRAGON, http://dragonproject.org.
[5] C. Evoli, D. Gaggero, D. Grasso, L. Maccione, J. Cosmol. Astropart. Phys. 0810 (2008) 018. arXiv:0807.4730.
[6] E. Parker, Planet. Space Sci. 13 (1) (1965) 9–49.
[7] L.J. Gleeson, W.I. Axford, Astrophys. J. 154 (1968) 1011.
[8] H. Gast, S. Schael, Proceedings of the 31st ICRC Lodz, 2009.
[9] I. Cholis, D. Hooper, T. Linden, 2015, arXiv:1511.01507.
[10] Y. Yamada, S. Yanagita, T. Yoshida, Geophys. Res. Lett. 25 (1998) 2353–2356.
[11] M. Zhang, Astrophys. J. 513 (1999) 409–420.
[12] P. Bobik, G. Boella, M. Boschini, C. Consolandi, S. Della Torre, et al., Astrophys. J. 745 (2012) 132. arXiv:1110.4315.
[13] R.D. Strauss, M.S. Potgieter, I. Büsching, A. Kopp, Astrophys. J. 735 (2011) 83.
[14] C. Pei, J.W. Bieber, R.A. Burger, J. Clem, J. Geophys. Res.: Space Phys. 115 (A14) (2010) 12107.
[15] K. Alanko-Huotari, I.G. Usoskin, K. Mursula, G.A. Kovaltsov, J. Geophys. Res.: Space Phys. 112 (2007) 8101.
[16] M.S. Potgieter, E.E. Vos, M. Boezio, N. De Simone, V. Di Felice, V. Formato, Sol. Phys. 289 (2014) 391. arXiv:1302.1284.
[17] SOLARPROP, http://www.th.physik.uni-bonn.de/nilles/people/kappl/.
[18] R. Kappl, M.W. Winkler, Phys. Rev. D 85 (2012) 123522. arXiv:1110.4376.
[19] J.R. Jokipii, E.N. Parker, Astrophys. J. 160 (1970) 735.
[20] C.W. Gardiner, Handbook of Stochastic Methods, Springer Verlag, Berlin, 1989.
[21] A. Kopp, I. Büsching, R.D. Strauss, M.S. Potgieter, Comput. Phys. Comm. 183 (2012) 530–542.
[22] S. Della Torre, et al., 2015, arXiv:1511.00541.
[23] GALPROP Manual, http://galprop.stanford.edu/download/manuals/galprop_v54.pdf.
[24] WSO, http://wso.stanford.edu/Tilts.html.
[25] I.G. Usoskin, et al., http://cosmicrays.oulu.fi/phi/Phi_mon.txt.
[26] J.R. Jokipii, D.A. Kopriva, Astrophys. J. 234 (1979) 384–392.
[27] M.S. Potgieter, H. Moraal, Astrophys. J. 294 (1985) 425–440.
[28] R.A. Burger, M.S. Potgieter, Astrophys. J. 339 (1989) 501–511.
[29] O. Adriani, et al., Astrophys. J. 765 (2013) 91. arXiv:1301.4108.
[30] J.L. Raath, M.S. Potgieter, R.D. Strauss, A. Kopp, 2015, arXiv:1506.07305.
[31] J.T. Hoeksema, Space Sci. Rev. 72 (1995) 137–148.
[32] SILSO World Data Center, International Sunspot Number Monthly Bulletin and online catalogue, http://www.sidc.be/silso/.
[33] Neutron monitors of the Bartol Research Institute, http://neutronm.bartol.udel.edu.
[34] I.G. Usoskin, K. Alanko-Huotari, G.A. Kovaltsov, K. Mursula, J. Geophys. Res.: Space Phys. 110 (A9) (2005) 12108.
[35] I.G. Usoskin, G.A. Bazilevskaya, G.A. Kovaltsov, J. Geophys. Res.: Space Phys. 116 (2011) 2104.
[36] E.C. Stone, A.C. Cummings, F.B. McDonald, B.C. Heikkila, N. Lal, W.R. Webber, Science 341 (2013) 150–153.
[37] R.A. Burger, M.S. Potgieter, B. Heber, J. Geophys. Res. 105 (2000) 27447–27456.
[38] BESS, S. Orito, et al., Phys. Rev. Lett. 84 (2000) 1078–1081. astro-ph/9906426.
[39] S. Haino, K. Abe, T. Fuke, T. Maeno, Y. Makida, et al., International Cosmic Ray Conference 3 (2005) 13.
[40] Y. Shikaze, et al., Astropart. Phys. 28 (2007) 154–167. astro-ph/0611388.
[41] K. Abe, et al., Phys. Lett. B 670 (2008) 103–108. arXiv:0805.1754.
[42] K. Abe, et al., Phys. Rev. Lett. 108 (2012) 051102. arXiv:1107.6000.
[43] K. Abe, et al., 2015, arXiv:1506.01267.
[44] R. Kappl, A. Reinert, M.W. Winkler, J. Cosmol. Astropart. Phys. 1510 (10) (2015) 034. arXiv:1506.04145.

[45] R. Kappl, M.W. Winkler, J. Cosmol. Astropart. Phys. 1409 (2014) 051. arXiv:1408.0299.
[46] G. Barbiellini, G. Basini, R. Bellotti, M. Bocciolini, M. Boezio, et al., Astron. Astrophys. 309 (1996) L15–L18.
[47] PAMELA, O. Adriani, et al., Phys. Rev. Lett. 111 (2013) 081102. arXiv:1308.0133.
[48] AMS, M. Aguilar, et al., Phys. Rev. Lett. 113 (2014) 121102.
[49] L. Maccione, Phys. Rev. Lett. 110 (8) (2013) 081101. arXiv:1211.6905.
[50] M. Di Mauro, F. Donato, N. Fornengo, A. Vittino, 2015, arXiv:1507.07001.
[51] M.S. Potgieter, E.E. Vos, R. Munini, M. Boezio, V. Di Felice, Astrophys. J. 810 (2) (2015) 141.
[52] O. Adriani, et al., Astrophys. J. 810 (2) (2015) 142.
[53] R.A. Burger, T.P.J. Krüger, M. Hitge, N.E. Engelbrecht, Astrophys. J. 674 (2008) 511–519.
[54] R.A. Burger, H. Moraal, G.M. Webb, Astrophys. Space Sci. 116 (1985) 107–129.
[55] R.A. Burger, M. Hattingh, Astrophys. Space Sci. 230 (1995) 375–382.