CRPropa 3.1 — a low energy extension based on stochastic differential equations

Lukas Merten,\textsuperscript{a,d} Julia Becker Tjus,\textsuperscript{a,d} Horst Fichtner,\textsuperscript{b,d} Björn Eichmann\textsuperscript{a,d} and Günther Sigl\textsuperscript{c}

\textsuperscript{a}Theoretische Physik IV: Plasma-Astroteilchenphysik, Ruhr-Universität Bochum, Universitätstrasse 150, 44801 Bochum, Germany
\textsuperscript{b}Theoretische Physik IV: Weltraum- und Astrophysik, Ruhr-Universität Bochum, Universitätstrasse 150, 44801 Bochum, Germany
\textsuperscript{c}II Institut für Theoretische Physik, Universität Hamburg, Luruper Chaussee 149, 22761 Hamburg, Germany
\textsuperscript{d}Ruhr Astroparticle- and Plasmaphysics Center (RAPP Center), 44801 Bochum, Germany

E-mail: lukas.merten@rub.de, julia.tjus@rub.de, hf@tp4.rub.de, eiche@tp4.rub.de, guenter.sigl@desy.de

Received April 26, 2017
Revised May 22, 2017
Accepted June 5, 2017
Published June 23, 2017

Abstract. The propagation of charged cosmic rays through the Galactic environment influences all aspects of the observation at Earth. Energy spectrum, composition and arrival directions are changed due to deflections in magnetic fields and interactions with the interstellar medium. Today the transport is simulated with different simulation methods either based on the solution of a transport equation (multi-particle picture) or a solution of an equation of motion (single-particle picture).

We developed a new module for the publicly available propagation software CRPropa 3.1, where we implemented an algorithm to solve the transport equation using stochastic differential equations. This technique allows us to use a diffusion tensor which is anisotropic with respect to an arbitrary magnetic background field. The source code of CRPropa is written in C++ with python steering via SWIG which makes it easy to use and computationally fast.

In this paper, we present the new low-energy propagation code together with validation procedures that are developed to proof the accuracy of the new implementation. Furthermore, we show first examples of the cosmic ray density evolution, which depends strongly on the ratio of the parallel $\kappa_{\parallel}$ and perpendicular $\kappa_{\perp}$ diffusion coefficients. This dependency is systematically examined as well the influence of the particle rigidity on the diffusion process.

Keywords: cosmic ray theory, galactic magnetic fields, ultra high energy cosmic rays

ArXiv ePrint: 1704.07484
1 Introduction

While the field of astroparticle physics has advanced significantly during the past decade coming closer to an identification of the sources, the origin of both Galactic and extragalactic cosmic rays is still not fully resolved. Modern experiments like IceCube [1, 2], Imaging Air Cherenkov Telescopes (H.E.S.S. [3], MAGIC [4], VERITAS [5]), the Pierre Auger Observatory (PAO) [6], the Telescope Array (TA) [7, 8] or AMS-02 [9] challenge theories and simulations with a high accuracy of the observed data. Especially the transition between the Galactic and extragalactic part of the energy spectrum is not fully understood. There are different simulation frameworks available for low-energy, Galactic and high-energy extragalactic propagation that meet a high standard already today [10–12]. One unified simulation framework, however, that can broadly and consistently describe all three major characteristics of
the observations — namely chemical composition, energy spectrum and anisotropy level of the arrival directions — is not yet developed.

In particular, there are two different fundamental approaches for the modeling of cosmic ray (CR) transport: (1) The solution of the transport equation enables a approximate description in different environments, and thus has the major disadvantage that it is necessary to model the diffusion of the particles via a diffusion tensor, often reduced to a one dimensional diffusion coefficient. Here, also a variety of (semi-)analytic theories exist. These are e.g. the leaky box or more complex models (see e.g. [13–16]). These models are able to describe the global shape of the spectrum but cannot fully reproduce the latest precision measurements. (2) The solution of the equation of motion, which is a very precise, but also a very CPU-time-consuming ansatz, only being usable at the highest energies, where it is feasible with respect to the computation times.

**Transport equation.** On Galactic scales, the solution of a set of transport equations is commonly used, describing the flux for the different particle species and the change of their energy spectrum through processes like diffusion in space and momentum, advection and adiabatic cooling. A typical representation of the transport equation is given here:

\[
\frac{\partial n}{\partial t} + \vec{u} \cdot \nabla n = \nabla \cdot (\hat{\kappa} \nabla n) + \frac{1}{p^2} \frac{\partial}{\partial p} \left( p^2 \kappa_{pp} \frac{\partial n}{\partial p} \right) + \frac{1}{3} (\nabla \cdot \vec{u}) \frac{\partial n}{\partial \ln p} + S(\vec{x}, p, t) .
\]

(1.1)

This is the so called Parker transport equation (augmented with a term describing momentum diffusion) which is a simplified version of Fokker-Planck transport equation. It is believed to be a good description of the particle transport problem in diffusive regimes (e.g. [10]). Here, \(n\) is the particle density, \(\vec{u}\) is the advection speed, \(\hat{\kappa}\) is the spatial diffusion tensor, \(p\) is absolute momentum, \(\kappa_{pp}\) is the momentum diffusion coefficient used to describe re-acceleration and \(S(\vec{x}, p, t)\) describes the sources of cosmic rays. For the case of an entire nuclear network, as it is present in the cosmos, eq. (1.1) can be extended to include catastrophic losses and gains from spallation (interaction with background matter) and nuclear decay. Furthermore, formulations of continuous losses from interactions with the magnetic field (e.g. synchrotron radiation) can be developed also for single species simulations.

Nowadays, there exist three major simulation tools for the Galactic cosmic ray transport: GALPROP [10], DRAGON [11] and PICARD [12]. GALPROP is probably the most widely used program but does not provide a method to use anisotropic, space dependent diffusion tensors, yet. Nonetheless, it is an excellent instrument to describe spallation and a variety of loss processes. DRAGON, which is based on the same principles as GALPROP, addresses the anisotropic diffusion problem among other things. It is known that the diffusive behavior along and perpendicular to the mean magnetic field line differ from each other (e.g. [17]). A more sophisticated numerical approach to solve the transport equation is PICARD. The biggest difference is that it uses multi-grids and provides a solver explicitly designed to yield the stationary solution of the transport equation.

Another technically different approach to solve the transport equation, commonly used for the propagation of cosmic rays inside the heliosphere [18, 19], makes use of the equivalence of parabolic partial differential equations and a corresponding set of stochastic differential equations, known from stochastic integrals and Ito calculus (see section 2 and [20]). First attempts to apply this method also to the Galactic propagation of cosmic rays are made in [21–24]. This concept is part of the methods introduced for Galactic propagation in this paper and will be discussed in detail in section 2.
**Equation of motion.** In CRPropa [25, 26] a single-particle approach is used for description of the transport of extra-galactic CRs. CRPropa3.0 propagates single particles forward in time through the numerical integration of the equation of motion [26]. This opens up many new possibilities for the simulation. For example, no assumptions on the diffusion tensor have to be made, but arbitrary magnetic field configurations can be used. This makes the simulation more fundamental. In addition things like tracing of distinct particle paths and a stochastic treatment of interactions is possible. However, this method is not applicable to Galactic transport problems, because it is computationally too time consuming for simulations using particles with energies below a few 10 PeV.

The energy range between the knee at $10^{15}$ eV [27] and the ankle at $10^{18.5}$ eV [6] in the cosmic ray spectrum is called transition region because there the source distribution shifts probably from Galactic to extragalactic sources (see e.g. [28] for a recent review). It is unclear where the influence of the extragalactic part of the spectrum begins to dominate. Furthermore, it would be interesting to decompose these two compositions in the knee-to-ankle region to learn more about possible sources with energies up to EeV energies. Up to now no open-access program exists which models the transition region as a combination of Galactic and extragalactic influences. Individual models have been presented describing a possible contribution of Galactic and extragalactic cosmic rays in this transition region [29–33]. In this paper, we introduce an extension of the public CRPropa code that enables the user to perform cosmic ray simulations at hundreds of PeV to Zev energies with the single-particle propagation method, but also the modeling of TeV-PeV energies with the low energy add-on that we present in this paper using stochastic differential equations for the multi-particle propagation. This code is open-access\(^1\) and can be used by the entire community [26].

1.1 The structure of CRPropa

In this section we explain the program structure of CRPropa 3.1 and how the new DiffusionSDE module is related to the other modules. CRPropa 3.1 is build of several independent modules which can be used to create a user defined simulation setup. This structure makes it easy to extend CRPropa with new modules, e.g. with new interaction modules.

The central data structure in CRPropa is the so called candidate. This object holds all necessary information about the (pseudo-) particle such as energy, direction, particle type, position and many more attributes. These attributes are altered and processed by the ModuleList during the propagation procedure. The ModuleList is basically a list containing all simulation modules chosen for a specific simulation. The user can load different kinds of modules into the ModuleList:

1. **Boundary modules:** deactivate (stop simulation) the candidate if the boundary condition is met.

2. **Source modules:** create new candidates and inject them into the simulation.

3. **Interaction modules:** take interactions into account, e.g. proton-photon-interaction, radioactive decay, etc.

4. **Deflection modules:** change the position of the candidate according to the equation of motion (PropgationCK) or transport equation (DiffusionSDE).

\(^1\)https://github.com/CRPropa/CRPropa3/releases.
5. **Observer modules**: detect particle if a certain condition is fulfilled.

6. **Output modules**: save the simulation results to a text- or hdf-file.

Figure 1 shows the simulation process. The modules, except for the source module, are called successively and act on a candidate with each integration time step $h$ as long as the candidate is active.

The new module DiffusionSDE can replace the module PropagationCK as needed. In particular, we recommend to use the DiffusionSDE module for propagation with an energy below 100 PeV, while PropagationCK should be used for samples that are dominated by energies range above 100 PeV. While these numbers are a good estimate for the Galaxy they may differ significantly in other surroundings.

It cannot be generally answered which module is better. The two modules simulate two fundamentally different descriptions of the propagation process. Which module suits best depends on the specific simulation setup. In the case of weak magnetic fields with gyro radii of the order of the coherence length of the magnetic field a description by the transport equation is not possible. Also in the case of vanishing turbulent magnetic fields the particle transport is not diffusive and therefore can not be described by eq. (1.1). In general, the distribution function $n$ has to be isotropic in momentum space to be described by the diffusion approximation.

On the other hand, the propagation of particles in very turbulent magnetic fields, where the gyro radius is smaller than the coherence length of the field, is computing time intensive. This may prevent the generation of sufficient statistics to draw significant conclusions.

In addition, we want to give a short overview of the different output formats and how they can be used to create observables which can be compared to measurements. CRPropa outputs can be generated in a tracking mode, where every (pseudo-) particle position during
the simulation process is recorded. This is very useful if the concrete particle trajectory is of interest, but obviously yields huge amounts of data if it used for a full scale simulation with billions of candidates and hundred thousands of integration time steps for each candidate. In contrast to that, most other output modes save only those candidates which are detected by an ObserverModule. Here, the cosmic ray properties (position, momentum, etc.) at the origin and at detection are stored. With the DiffusionSDE module we developed also a new ObserverTimeEvolution module which takes snapshots of the total cosmic ray distribution for user defined time points.

Furthermore, not only cosmic rays but also neutral particles like \( \gamma \) rays or neutrinos, which are created as secondaries during interactions, can be propagated and stored.

2 Transport model and stochastic differential equations

In order to extend CRPropa toward lower energies and by that to a dedicated Galactic propagation tool, we decided to use stochastic differential equations. This ansatz is most compatible with the single particle tracking used for the extra-galactic propagation.

Furthermore, it has some advantages over classical grid based methods. Since the calculation is not restricted to a grid, adaptive algorithms can be applied to vary the integration time step which depends on the local magnetic field geometry. This leads to a simulation as accurate as needed and as effective as possible. To take all possible interaction channels into account the grid based methods have to calculate the whole (DRAGON and GALPROP) or parts (PICARD) of the reaction network twice. That is not necessary following the stochastic approach, where all secondary particles are propagated as new particles.

The diffusion equation is solved in the local frame of the magnetic field line (see section 2.3) which implies nearly no restriction on the allowed magnetic field set up. This means that no analytical description of the magnetic field is needed. In contrast to the approach in [22] the magnetic field direction is calculated on the fly and there is no need for derivatives of the vector field.

In times of huge CPU clusters the potential of parallelization is a crucial factor for a simulation program. The solution of the stochastic differential equation (SDE) is trivial to parallelize because the different phase space elements (hereafter called pseudo-particles) are independent of each other. The computation time for pure propagation of pseudo-particles scales linearly with the number of cores.

Furthermore, the simulation results can be reweighed after the actual simulation is finished. The pseudo-particles can for example be weighted according to their energy at the source to change effectively the energy spectrum of the source distribution without the need of a new full size simulation. This makes it easy and computationally cheap to do parameter studies. It has to be mentioned that a change of the propagation parameters, such as diffusion coefficient or magnetic field configuration cannot be changed after the simulation.

2.1 Stochastic differential equations

In this section we briefly summarize the mathematical background of stochastic differential equations. Here, we only discuss the one-dimensional case which can in principle be generalized to more dimensions. Furthermore, it should be noted that the rigorous mathematical description is always just an approximation of a real physics problem, especially with respect to correlation times etc.
Stochastic differential equations have been examined since more than one hundred years ago. Starting with studies on the Brownian motion by Einstein and Langevin who derived the so-called Langevin equation independently:

$$\frac{dx}{dt} = a(x, t) + b(x, t)\xi(t) , \quad (2.1)$$

where $\xi(t)$ is a “rapidly [in time] fluctuating random term”, meaning $\xi(t)$ and $\xi(t')$ are statistically independent for $t \neq t'$ [20]. That means the random term is fully uncorrelated which can be described as:

$$\langle \xi(t + \tau)\xi(t) \rangle = \lim_{T \to \infty} \frac{1}{T} \int_0^T \xi(t)\xi(t + \tau) \, dt \quad (2.2)$$

$$= \delta(\tau) .$$

Furthermore, we require the time average to vanish ($\langle \xi(t) \rangle = 0$) because any drift term can be described by $a(x, t)$. Equation (2.1) can consistently be interpreted by the corresponding integral equation (see e.g. [20]):

$$x(t) = x(0) + \int_0^t a[x(s), s] \, ds + \int_0^t b[x(s), s] \, dW(s) . \quad (2.3)$$

Here, we used the fact that the integral of $\xi(t)$ is a Wiener process, $\int_0^t \xi(t') \, dt' = W(t)$, which allows for the replacement of $\xi(s) \, ds = dW(s)$. A Wiener Process is a Markov process and it can be interpreted as the solution of the diffusion equation with zero drift and a constant diffusion coefficient that is equal to 1. The sample paths are continuous but not differentiable. Equation (2.3) is the prototype of a stochastic integral equation. The quantity $x(t)$ is determined by the initial condition $x(0)$ a deterministic term (first integral) and a stochastic term (second integral). The details of the existence and uniqueness of solution of these types of equations is beyond the scope of this paper and we refer the reader to the detailed studies in the textbook by Gardiner [20]. We just want to emphasize that the solution is not fully determined by the starting condition — as it is the case for ordinary differential equations — but also depends on the particular realization of the Wiener process.

In addition it should be mentioned that the last term of eq. (2.3) — the stochastic integral — can be interpreted in two ways, which differ in the choice where between the grid points $t_i$ and $t_{i-1}$ with a time ordering $t_0 \leq t_1 \leq \cdots \leq t_n = t$, the stochastic part of the integral $[b[x(s), s]]$ is evaluated. It can be shown that the stochastic differential equation is independent of the two, namely Itô and Stratonovich, interpretations of the stochastic integral [20].

If a stochastic quantity $x(t)$ is described by eq. (2.3) for all $t$ and $t_0$ it obeys the Itô SDE given by:

$$dx(t) = a[x(t), t]dt + b[x(t), t]dW(t) . \quad (2.4)$$

It can be concluded that the solution of the Itô SDE eq. (2.4) is described by taking the limit $(t_{i+1} - t_i) = \Delta t_i \to 0$ for the discretized version of eq. (2.4):

$$x(t_{i+1}) = x(t_i) + a(x(t_i), t_i)\Delta t_i + b(x(t_i), t_i)\Delta W_i , \quad (2.5)$$

where $\Delta W_i = W(t_{i+1}) - W(t_i)$ and the grid points are time ordered $t_0 \leq t_1 \leq \cdots \leq t_n = t$. 

– 6 –
2.2 Connection between Fokker-Planck and stochastic differential equations

The Fokker-Planck equation (FPE) is a parabolic partial differential equation and can be seen as a special case of the differential form of the Chapman-Kolmogorov equation with vanishing jump probability (e.g. [20]). In general, a FPE can be written as:

\[
\frac{\partial n(x,t; y, t')}{\partial t} = -\sum_i \frac{\partial}{\partial x_i} [A_i(x,t)n(x,t; y, t')] + \frac{1}{2} \sum_{i,j} \frac{\partial^2}{\partial x_i \partial x_j} [B_{ij}(x,t)n(x,t; y, t')] ,
\]

where \(A_i(x,t)\) is the drift vector, \(B_{ij}(x,t)\) is the diffusion tensor and \(n(x,t; y, t')\) is the density at place \(x\) and time \(t\) depending on the density at place \(y\) and time \(t'\). Here, \(x\) and \(y\) are in principle higher dimensional phase space vectors. The propagation of the phase space elements is described by the corresponding stochastic differential equation (here for the four dimensional case with three spatial and one momentum dimension):

\[
dr_\nu = A_\nu \, dt + D_{\nu \mu} \, d\omega_\mu ,
\]

where \(dt\) is the time increment, \(r_\nu\) is a 4-dimensional vector (\(\vec{r}, ||\vec{p}||\)) and \(d\omega_\mu = \sqrt{dt} \, \eta_\mu\) stands for a 4-dimensional Wiener process with Gaussian noise. The square-root proportionality of \(d\omega\) to \(dt\) is easily connected to the general behavior of normal diffusion \(\langle x^2 \rangle/t \propto D\). \(A_\nu\) is a transport vector, \(D_{\nu \mu}\) is a 4x4 tensor representing the diffusion and \(\eta_\mu\) is a 4-vector of independent normal distributed random variables with zero mean and unity variance.

The derivation of eq. (2.7) from eq. (2.6) is beyond the scope of this paper and the interested reader is referred to textbooks like e.g. [20]. The connection between the FPE in eq. (2.6) and the SDE in eq. (2.7) can be compared to Liouville’s equation and the ordinary differential equation which describes a deterministic motion:

\[
\frac{\partial n(x,t; y, t')}{\partial t} = -\sum_i \frac{\partial}{\partial x_i} [A_i(x,t)n(x,t; y, t')] , \quad \frac{dx}{dt} = A[x(t)] .
\]

Indeed Liouville’s equation is only a special case of the Fokker-Planck equation for vanishing stochastic motion.

Since the spatial and momentum operators in eq. (1.1) decouple for most Galactic applications, eq. (2.7) can be split into two parts:

\[
d\vec{r} = \vec{A} \, dt + D_r \, d\vec{\omega}_r \quad \text{(2.9)}
\]

\[
dq = A_q \, dt + D_q dq \, d\omega_q \quad \text{(2.10)}
\]

where the absolute value of the momentum is denoted as \(q = ||\vec{p}||\), see e.g. [34] for a more detailed explanation.

The remaining problem is the calculation of the tensor \(D_r\) and scalar \(D_q\) in the SDE from the given spatial diffusion tensor \(\kappa_r\) and the given momentum diffusion scalar \(\kappa_{qq}\), respectively, according to eq. (1.1). In general, one has to find the square root of the diffusion tensor \(\kappa\), which is the solution of \((\kappa + \kappa^t) = DD^\dagger\). In our case this reduces to a simple uncoupled set of equations, because we neglect drift terms and solve the equation in the local frame of the magnetic field line. Hence, the diffusion tensor is always diagonal which leads two the following relations:

\[
D_{ij} = \delta_{ij} \sqrt{2\kappa_{ij}} , \quad D_{qq} = \sqrt{2\kappa_{qq}} .
\]
Note that while the SDE method allows for the computation of ‘quasi-trajectories’ of particles and thereby, as mentioned above, makes it most compatible to the single particle tracking used for extragalactic cosmic rays, it nonetheless requires the knowledge of the diffusion tensor. This is opposite to so-called full-orbit simulations e.g. [35, 36] where the elements of the diffusion tensor are determined from the particle trajectories obtained from a direct solution of their equations of motion.

2.3 Numerical implementation

In this section we will briefly explain the developed algorithm which calculates the next pseudo-particle position. First, we discuss the Euler-Mayurama scheme which can be used to integrate the SDE when the local orthonormal basis is known. In section 2.3.2 the implemented adaptive method for the calculation of the local trihedron — orthonormal basis defined by the Frenet equation — is explained.

2.3.1 The Euler-Maruyama scheme

To explain the diffusion code all terms except spatial diffusion are neglected in this section. Following eq. (2.10) and eq. (2.9) the so called Euler-Maruyama scheme (e.g. [37]) can be derived:

\[
\vec{x}_{n+1} = \vec{x}_n + D_r \Delta \vec{\omega}_r \\
= \vec{x}_n + \left( \sqrt{2\kappa_{||}} \eta_1 \vec{e}_t + \sqrt{2\kappa_{\perp,1}} \eta_{\perp,1} \vec{e}_n + \sqrt{2\kappa_{\perp,2}} \eta_{\perp,2} \vec{e}_b \right) \sqrt{h},
\]

(2.12)

where \( h = t_{n+1} - t_n \) is the integration time step. Further, the orthonormal basis\(^2\) \( \{\vec{e}_t, \vec{e}_n, \vec{e}_b\} \) is generally defined by the Frenet-Serret-equation (e.g. [38]) which can be rewritten for magnetic field lines (e.g. [19]) as:

\[
\vec{e}_t = \vec{B}_{\text{reg}} / B_{\text{reg}} \\
\vec{e}_n = (\vec{e}_t \cdot \nabla) \vec{e}_t / k \\
\vec{e}_b = \vec{e}_t \times \vec{e}_n,
\]

(2.13)

where \( \vec{B}_{\text{reg}} \) is the regular or coherent background field vector and \( k = |d^2 \vec{r}/ds^2| \) is the curvature of the parametrized field line \( \vec{r}(s) \).\(^3\) Generally, all three vectors of the local trihedron \( \{\vec{e}_t, \vec{e}_n, \vec{e}_b\} \) have to be calculated on the fly at each time step. Although the two perpendicular diffusion coefficients differ in principle, we state for the Galactic scenario that they can be assumed equal \( \kappa_{\perp,1} = \kappa_{\perp,2} \). Anisotropic perpendicular diffusion can, in general, either result from anisotropic (or non-axisymmetric) turbulence perpendicular to the mean magnetic field [39], which is not the case for the Galactic magnetic field (see the turbulent component of the JF12 field [40]). Otherwise the decoupling of the perpendicular directions is caused by a large curvature of the background field on scales of the mean scattering length of the particles. This is also not observed for the Galactic magnetic field. This is in contrast to simulations in the heliosphere, where the consideration of anisotropic perpendicular diffusion can be crucial (see e.g. [41]).

\(^2\)The orthonormal basis is the local trihedron of the magnetic field line \( \vec{B}(\vec{x}_n) \) when it is interpreted as a three-dimensional curve.

\(^3\)In this paper the term “field line” always refers to the coherent background field vector \( \vec{B}_{\text{reg}} \).
2.3.2 Adaptive calculation of the local trihedron

The tangential vector of the magnetic field line $\vec{e}_t$ is calculated via the so called Cash-Karp (CK) algorithm \cite{42}, an adaptive Runge-Kutta (RK) algorithm. With this algorithm we are able to adapt the time integration step to minimize the number of steps. The local truncation error for the field line integration can be set by the user. In this way the overall computation time is reduced without any losses in accuracy.

The tangential vector is approximated by the difference of two points on the magnetic field line $\vec{e}_t = \vec{r}_{\text{end}} - \vec{r}_{\text{start}}$. Here, the starting point is given by the old pseudo-particle position $\vec{r}_{\text{start}} = \vec{x}_n$ (see eq. (2.12)) and the end position $\vec{r}_{\text{end}}$ is calculated with a field line integration:

$$\vec{r}_{\text{end}} = \vec{r}_{\text{start}} + \int_0^L \vec{B}/B \, ds , \quad (2.14)$$

where $L = \sqrt{2\kappa \parallel \eta \parallel} \sqrt{\tau} h$ is the arc length of the field line elements and has the length of the diffusion step in parallel direction. The CK algorithm uses a fourth and fifth order RK algorithm to calculate two solutions of the initial condition problem given in eq. (2.14). The norm of the difference of the vectors is used as a measure for the local truncation error. The step is accepted if:

$$m = \parallel \vec{r}_{\text{end},4} - \vec{r}_{\text{end},5} \parallel \leq \xi \cdot \text{kpc} , \quad (2.15)$$

where $\xi$ is the user set precision and $\vec{r}_{\text{end},4}$ and $\vec{r}_{\text{end},5}$ are the 4th- and 5th-order solution, respectively.

In the case of a rejected step integration length $L$ is bisected until the condition in eq. (2.15) is fulfilled or the minimum step is reached. Following this procedure the end position vector $\vec{r}_{\text{end}}$, used for the calculation of the tangential vector $\vec{e}_t$, is derived by $2^n$ consecutive solutions of eq. (2.14) if $(n-1)$ step attempts are rejected:

$$\vec{r}_{\text{end}} = \vec{r}_{\text{start}} + \sum_{j=0}^{2^n-1} \int_{L_j}^{L_{j+1}} \vec{v}(s) \, ds , \quad (2.16)$$

where $L_j = 2^{-n}L_j$ and $\vec{v} = \vec{B}/|\vec{B}|$ is the normalized magnetic field direction. This procedure is not as efficient as the conventional step adaption ($h_{\text{next}} = 0.95 h \cdot m^{-0.2}$ see \cite{42}) but is necessary. If the step size is reduced as conventional, large values of $\eta$ will be rejected more frequently as compared to small values. This would lead to an effective underestimation of the tails of the Wiener process $d\omega$. Furthermore, this would not just be an inaccurate description of the diffusion process but is simply not an approximation of the diffusion. The error is hard to quantify since it depends on the global magnetic field structure, the diffusion tensor and also on the simulation set-up such as minimum/maximum step size and the precision. With the procedure of eq. (2.16) we ensure that the particles are always propagated over the full drawn diffusion step but use an adequate number of intermediate steps.

The suggested next integration time $h_{\text{next}}$ is:

$$h_{\text{next}} = h \cdot 2^{(-2n)} \quad \text{if } n \geq 0 \quad (2.17)$$

$$h_{\text{next}} = h \cdot 4 \quad \text{else} \quad (2.18)$$

When the current step is not decreased the next step should be increased. The next time integration time step $H_{\text{next}}$ is increased by a factor 4, which leads on average to a doubled parallel step length $L$, due to the square root dependence of the step length from the integration time.
3 Validation of the algorithm

The new code is validated in two ways. Firstly, we proof that our code reproduces the correct solution for a given diffusion coefficient. In doing so, we compare in section 3.1 the simulation with simple analytical expectations. Further, we validate the adaptive field line integration in section 3.2.

All tests that are developed for this paper test a mathematical problem. That means they are not designed to describe the physical reality. So before the software is used for physical simulations it should be carefully examined if the necessary assumptions made for the tests do also hold (at least approximately) in the physical simulation.

3.1 Homogeneous background field

3.1.1 Green’s function

For the first test we simulate protons under the influence of diffusion in a homogeneous magnetic field \( \vec{B} = B_0 \cdot \hat{e}_z \). We use a diffusion tensor that is anisotropic with respect to the background field \( \hat{\kappa} = \text{diag}(\kappa_\perp, \kappa_\perp, \kappa_\parallel) \) with \( \kappa_\parallel = 10 \kappa_\perp \). Here, the probability distribution function (PDF or density \( n \)) of a pseudo particle is known and given by the Green’s function of the following equation:

\[
\frac{\partial n}{\partial t} = \nabla \cdot (\hat{\kappa} \nabla n) + \delta(\vec{r})\delta(t) ,
\]

which is solved by

\[
n(r_i, t) = \frac{1}{(4\pi\kappa_{ii}t)^{d/2}} \cdot \exp \left( -\frac{r_i^2}{4\kappa_{ii}t} \right) .
\]

Here, \( \kappa_{ii} \) is the diffusion coefficient, \( d \) is the dimension (\( d = 1 \) in this test case, because the three directions are decoupled) and \( r_i \) is the \( i \)th component of the spatial vector. In other words, we expect that the distributions of the three vector components \( (x(t), y(t), z(t)) \) of the pseudo-particles follow normal distributions with zero means and variances \( \sigma_i = \sqrt{2\kappa_{ii}t} \) since that is the analytical solution as given in eq. (3.2).

Figure 2 shows the density distributions for all three components at three different times. The difference between the parallel (\( z \)-) and the perpendicular (\( x \)-, \( y \)-) direction is clearly visible. Furthermore, it can be seen that the width of the distribution increases with increasing time and that the speed of this process depends on the diffusion coefficient. To emphasize this diffusive behavior we plot the absolute distance from the origin for all three components in combination with the analytical solution in figure 3. A difference between the analytical solution (line) and the simulated data (histogram) cannot be seen by eye. To quantify these results a statistical test is applied to the data.

Since we expect that the differences per bin between the simulated and expected distribution are normally distributed we use a \( \chi^2 \)-test of the probability distribution function (PDF) of the particle range (norm of the end position vector components). So here, \( P(r) \) is tested rather than \( P(\vec{r}) \). This PDF is described by:

\[
\tilde{P}(R_t) = \frac{2}{\sqrt{4\pi\kappa_{ii}t}} \cdot \exp \left( -\frac{R_t^2}{4\kappa_{ii}t} \right) .
\]

Here, \( R_t = |r_i| \) is the absolute value of the vector components. The distance is sorted into bins while it is ensured that the minimal count per bin is greater than 5. This is done by a
consecutive reduction of the number of bins. These bin counts are compared to the expected values derived from the integration of eq. (3.3) over the bin edges. From that the $\chi^2$ value is calculated as:

$$\chi^2 = \sum_{i=1}^{n} \frac{(O_i - E_i)^2}{E_i},$$  \hspace{1cm} (3.4)

where $n$ denotes the number of bins and $O_i$ and $E_i$ the observed and expected count of events in the $i^{th}$ bin, respectively. Using the $\chi^2$ test result according to eq. (3.4) we calculate the p-value:

$$p(\chi^2) = \int_{\chi^2}^\infty P_k(\chi^2) \, d\chi^2,$$  \hspace{1cm} (3.5)

where $p$ is the p-value\(^4\) and $P_k(\chi^2)$ is the $\chi^2$-distribution with $k$ degrees of freedom.

\(^4\)Although the p-value is seen as deprecated by the American Statistical Association among others it is not critical to use it here.

---

**Figure 2.** Diffusion in a homogeneous magnetic field for different times. The magnetic field is aligned with the $z$-axis.

**Figure 3.** Diffusion in a homogeneous magnetic field for different times. The distance from the origin is shown for all three components. The simulated data are shown as a histogram and the analytical expectations as marker.
The Null hypothesis, that the sample is drawn from eq. (3.3), is rejected if the p-value is not in the range \(0.005 < p < 0.995\). We choose this broad range of acceptance because we want to minimize the number of false rejections. Since problems with the algorithm, at least to our experience, lead to very large deviations we decided to use this robust acceptance range. This also allows us to use this test for the automated testing of CRPropa (see below). Here, a large number of statistical tests is applied and the test suite has to be rerun if one of them fails, so a very robust test setup is desired.

According to the p-values summarized in table 1 and table 2, respectively, we conclude that the algorithm reproduces the diffusive behavior correctly in the case of a homogeneous background field using an anisotropic diffusion tensor.

Table 1. P-values for diffusion in homogeneous background field.

| Coordinate | Time [s] | \(10^{12}\) | \(5.2 \cdot 10^{12}\) | \(10^{13}\) |
|------------|----------|-------------|-----------------|-------------|
| X          | 0.64     | 0.95        | 0.06            |
| Y          | 0.94     | 0.85        | 0.02            |
| Z          | 0.23     | 0.58        | 0.30            |

Table 2. \(\chi^2\)/dof for diffusion in homogeneous background field.

| Coordinate | Time [s] | \(10^{12}\) | \(5.2 \cdot 10^{12}\) | \(10^{13}\) |
|------------|----------|-------------|-----------------|-------------|
| X          | 0.78     | 0.37        | 1.80            |
| Y          | 0.38     | 0.48        | 2.22            |
| Z          | 1.29     | 0.84        | 1.19            |

Unittest in CRPropa 3.1. A slightly different version of the tests described above is implemented in the unit test framework of CRPropa 3.1 which can be run within the installation process of the software. In this way it is ensured that the code is executed as expected. If all tests are passed, the user will be sure that no known software problems occur after the installation.

Here, two statistical tests are implemented. First, the Anderson-Darling test [43] is applied on the simulated data. This test can be compared to the commonly known Kolmogorov-Smirnov test but gives more weight to the tails of the distribution. The Anderson-Darling test is very sensitive to outliers of the data but cannot be used to test against a specific normal distribution. The Anderson-Darling turned out to be a very good test to find problems in the numerical implementation of the diffusion algorithm. Since it is sensitive to outliers the test detects even a small number of failed pseudo-particle propagations which was helpful during the development of the code.

If the Anderson-Darling test does not find any significant deviations from a normal distribution we can assume that the errors of the simulated data compared with the analytic results will follow also a normal distribution. This allows us to apply the \(\chi^2\)-test to verify that the width of the normal distribution does not deviate significantly from the expected one.
3.1.2 Stationary test solution

This test, as the one before, is not a physical model but rather a artificial problem specifically designed to test the algorithm. Therefore a comparison with real physical problems can be misleading.

In this section the test setup for a more complex simulation which is also used in the testing of the PICARD code [12] is explained. Here, the stationary solution of a simple diffusion equation:

\[-\nabla \cdot (\hat{\kappa} \nabla n(\vec{r})) = s(\vec{r}) ,\]  

(3.6)

with \(\hat{\kappa} = \text{diag}(\kappa_{xx}, \kappa_{xx}, \kappa_{zz})\) and a source term \(s(\vec{r})\) is tested against the known analytical solution. With the boundary condition \(n(x = \pm R, y = \pm R, z = \pm H) = 0\) and the source term defined by

\[s(\vec{r}) = \frac{4}{\pi^2} \left( \frac{2\kappa_{xx}}{4R^2} + \frac{\kappa_{zz}}{2H^2} \right) \cdot \cos \left( \frac{x\pi}{2R} \right) \cos \left( \frac{y\pi}{2R} \right) \cos \left( \frac{z\pi}{2H} \right) ,\]  

(3.7)

where the solution is given by:

\[n_{\text{ana}}(\vec{r}) = \cos \left( \frac{x\pi}{2R} \right) \cos \left( \frac{y\pi}{2R} \right) \cos \left( \frac{z\pi}{2H} \right) .\]  

(3.8)

For the test procedure we inject pseudo-particles using a simple rejection sampling method. In that way we make sure that the source distribution follows eq. (3.7) ignoring the constant factor for now. Afterward they are propagated forward in time and the position of the particles is recorded at \(N\) consecutive times \(t_1, t_2, \ldots, t_N\). The boundary condition is implemented simply by deactivating the candidates when they reach the boundary. This removal of the pseudo-particles from the simulation leads automatically to \(n(x = \pm R, y = \pm R, z = \pm H) = 0\) as required.

The time dependent particle distribution \(n(t_i)\) corresponds to the solution of the following equation:

\[\frac{\partial n}{\partial t} = \nabla \cdot (\hat{\kappa} \nabla n) + s(\vec{r})\delta(t - t_i) ,\]  

(3.9)

which differs from eq. (3.6) since it is not a stationary equation. We assume that the particle distribution \(n(t_i)\) does not vary significantly in the time interval \(h_i = t_{i+1} - t_i\) so we can calculate the stationary solution from our particle distribution as follows:

\[n_{\text{sim}}(\vec{r}) = \sum_{i=1}^{N_{\text{snap}}} n(t_i) h_i w .\]  

(3.10)

Here, \(w\) is a weight according to the source function, where \(s(\vec{r}) = w \cdot n_{\text{ana}}(\vec{r})\). In general, eq. (3.10) is an infinite sum but the density \(n(t)\) decreases with time \(\lim_{t \to \infty} n(t) = 0\). It follows that there exists a time \(t = T_{\text{max}}\) with \(\int_V n(T_{\text{max}}) dV \leq \beta N\) where only a mere fraction \(\beta\) of the particle remain in the simulation volume. The maximum integration time \(T_{\text{max}}\) needs to be chosen such that \(\beta\) is small enough for the aspired accuracy. In other words, since the time integration of eq. (3.9) leads to eq. (3.6), the discrete solution in eq. (3.10) is an approximation for the solution of the original stationary transport equation.

This test is designed to answer several questions:
1. How many snapshots \( N_{\text{snap}} \) are needed to resolve the solution sufficiently? The number of snapshots \( N_{\text{snap}} \) determines the number of summands in eq. (3.10).

2. What is a reasonable size of the integration time step \( h \)?

3. What is the integration time \( T_{\text{max}} \) to reach the stationary solution?

In this test we implemented the parameters as \( \kappa_{zz} = 10^{24} \text{ m}^2/\text{s}, \kappa_{xx} = 0.1 \kappa_{zz}, R = H = 0.5 \text{ kpc}, T_{\text{max}} = 50 \text{ kpc}/c \) and \( N = 10^6 \). We used two different approaches: in the first one we fixed the integration time step \( h = 1 \text{ pc}/c \) and varied the number of snapshots \( N_{\text{snap}} \). To answer the second question we fixed the number of snapshots \( N_{\text{snap}} = 500 \) and varied the integration time step \( h \). In both setups we analyzed the integrated number density depending on the maximum integration time (see paragraph ‘Total particle number’). After that we examined the accuracy of the spatial density distribution \( n_{\text{sim}}(\vec{r}) \) for the latest time point \( T_{\text{max}} = 50 \text{ kpc}/c \) (see paragraph ‘Spatial accuracy’).

**Total particle number.** Figure 4 and figure 5 each show the simulated integrated particle density \( \int_V n_{\text{sim}}(\vec{r}) \, dV \) depending on the maximum integration time. In general both figures show that the approximated solution asymptotically increases to a value close to the expectation.

Figure 4 shows that the estimated density increases with an increasing number of snapshots. This is reasonable because a higher number of snapshots approximates the time evolution better. This gives effectively more weight to the early time points with higher densities as compared to the case with a low time resolution, where the first time snapshot is taken when already a significant number of particles is lost. The error for a sufficient number of snapshots \( (N > 500) \) is below one percent for \( T_{\text{max}} > 50 \text{ kpc}/c \). Furthermore, it is visible that the integrated particle number does not converge to a completely flat distribution, due to the fact that \( \beta > 0 \). That means that not all pseudo-particles have time to leave the simulation volume before the end of the simulation time. A longer time integration can fix this error but is not computing-time efficient. Moreover, the density is slightly over-estimated. This is presumably caused by problems of the numerical implementation of the boundary conditions. To estimate the statistical error we repeated the simulation ten times and calculated the mean and standard deviation from these sets.

Figure 5 shows the results for different integration time steps \( h \). Here, an increased accuracy (decreased time step) leads to a lower integrated particle number, where \( h = 0.1 \text{ pc}/c \) gives an approximation smaller than the expected one. From that we conclude that an integration time step of \( h = 0.1 \text{ pc}/c \) is small enough to fully resolve the boundary. In addition, we allowed the adaptive algorithm to choose the step size depending on the pseudo-particle position. In doing so, the algorithm becomes significantly faster than it would have been using the minimal allowed integration step only. On the other hand, it is a lot more accurate than the solution for the maximum allowed step.

**Spatial accuracy.** After that the accuracy of the spatial density distribution \( n_{\text{sim}}(\vec{r}) \) is examined for \( T_{\text{max}} = 50 \text{ kpc}/c \). In doing so, the data are binned in a three-dimensional histogram with ten bins in each direction to ensure sufficient statistics in each bin. Furthermore, the expected number of particles per bin is calculated using the bin edges and the analytic solution given in eq. (3.8).

A first analysis showed that the error depends on the bin position. More precisely we can identify two groups of bins: the two outer layers in z-direction (orange bins in figure 6(a)) and the other bins (green bins in figure 6(b)). This anomaly of the errors is an effect of the
Figure 4. Integrated number density depending on the maximum integration time. Different colors show different numbers of snapshots $N_{\text{snap}}$. The shaded color bands correspond to a statistical uncertainty of one standard deviation.

Figure 5. Integrated number density depending on the maximum integration time. Different colors show different integration time steps $h$. The shaded color bands correspond to one standard deviation statistical uncertainty.

anisotropic diffusion tensor. Since the mean spatial step is larger in z-direction than in the x-y-plane the boundary condition is resolved worse in these bins. In general, all simulations could reproduce the shape of the analytical solution. Figure 6(b) displays exemplary that the density distribution for $N_{\text{snap}} = 500$ and $\Delta h = 0.1 \, \text{pc}/c$ reproduces the expected shape of eq. (3.8).
Table 3. Mean relative error for different simulation setups with fixed integration time $h = 0.1$ pc/c.

| Bins         | $N_{\text{snap}}$ | 100  | 500  | 1000 | 2000 |
|--------------|-------------------|------|------|------|------|
| Z-boundary   | 0.023             | 0.045| 0.048| 0.049|
| Inner        | -0.019            | 0.002| 0.006| 0.006|
| All          | -0.011            | 0.011| 0.014| 0.015|

Table 3 shows the relative errors of setup 1 (fixed integration step) broken down for the different bin groups. The table gives also the overall mean per simulation setup. Evidently, the increase from 1000 to 2000 snapshots has nearly no effect on the results but it doubles the amount of data being processed. Consequently, a good guess for the required accuracy for the problem saves a lot of memory and processing time. The decreasing accuracy from 500 to 1000 snapshots can be explained by the fact that the approximated solution is not yet fully stationary, meaning the maximum integration time $T_{\text{max}}$ is not long enough.

Table 4 clearly exposes that a decreased integration time step $h$ decreases the relative error effectively. The good performance of the adaptive algorithm is confirmed also in the spatial analysis. The algorithm uses larger time steps in the middle of the simulation and decreases the integration time at the boundary. This results in a better accuracy than reached using the upper integration time limit at low computational costs. However, the accuracy of the lower time integration limit is not met.

A more detailed analysis, including full error resolution per bin and spread of the error in the different groups, is given in appendix A.
Table 4. Mean relative error for different simulation setups with fixed number of snapshots $N_{\text{snap}} = 500$.

| Bins | Time step | 100  | 0.1–100  | 10  | 0.01–10  | 1   | 0.1       |
|------|-----------|------|-----------|-----|-----------|-----|-----------|
| Z-boundary |          | 0.554| 0.188     | 0.172| 0.12      | 0.045| 0.008     |
| Inner  |          | 0.113| 0.033     | 0.036| 0.019     | 0.002| -0.005    |
| All    |          | 0.201| 0.064     | 0.063| 0.039     | 0.01 | -0.003    |

3.2 Field line integration

As it is shown in section 3.1 the code reproduces the solution of the transport equation for a homogeneous magnetic field very well. The next step is to check whether the field line integration or derivation of the tangent vector $\vec{e}_t$ works. Therefore, we developed two different tests. The first problem has a simple analytic solution which enables us to calculate errors very carefully (see section 3.2.1). The second one uses a realistic magnetic field as it may be chosen for a full simulation setup but does not provide easy analytic solutions (see section 3.2.2).

As explained in section 2.3.1 the algorithm is designed to adapt the integration time to follow the magnetic field line with the user defined precision. The basic assumption of this test is pure parallel diffusion, i.e. $\kappa_\parallel = 0$. Since magnetic field lines do not intersect, particles cannot leave their original field line. We make use of this fact in the following tests as we use the deviation from the field line as a measure for the accuracy of the algorithm.

3.2.1 Spiral line

In this section we explain the test of the field line integration of the developed algorithm. A simple curve $\vec{r}_{\text{spiral}}$ of the form:

$$\vec{r}_{\text{spiral}}(z) = \begin{pmatrix} \frac{z \cdot \cos(2\pi z/s)}{z} \\ \frac{z \cdot \sin(2\pi z/s)}{z} \end{pmatrix}$$

is used as the field line. The curve is parametrized in $z$ and the constant $s$ determines the windings per height. The form of the field line is shown in figure 7. Due to the varying curvature radius of the spiral eq. (3.11) this curve is ideal to test the adaptive algorithm. In doing so, we inject 10,000 particles at the origin $\vec{r}_0 = 0$ into the simulation and propagate them forward for a time span of $T_{\text{max}} = 100$ kpc/c. We used a minimum/maximum integration step of $h_{\text{min}} = 10^{-5}$ kpc/c and $h_{\text{max}} = 1$ kpc/c, respectively. In different simulations the local truncation error $\xi$ is varied. The diffusion coefficient in parallel direction is fixed to $\kappa_\parallel \approx 6.4 \cdot 10^{25}$ m$^2$/s and the diffusion in perpendicular direction is neglected $\kappa_\perp = 0$. The winding constant is fixed at $s = 0.02$ kpc.

In addition to the position of the pseudo-particles we also tracked the arc length $L_{\text{sim}}$ of the particles trajectories. The simulated arc length $L_{\text{sim}}$ is calculated as the sum of the propagation steps considering the sign and not just the absolute value. From this variable we are able to reconstruct the true particle position. We calculate the arc length $L$ for an
Figure 7. End positions of 500 pseudo-particles (blue spheres) and field line (orange line). Particles were injected at the middle of the double spiral at $\vec{r}_0 = 0$.

The end position $z_{\text{max}} = z(T_{\text{max}})$ and fixed start position $z_0 = 0$:

$$L_{\text{ana}} = \int_0^{z_{\text{max}}} \sqrt{\frac{d r_{\text{spiral}}(z)}{dz}} dz = \int_0^{z_{\text{max}}} \left( 2 + \left( \frac{2\pi}{s} z \right)^2 \right)^{0.5} dz = \frac{1}{2} z_{\text{max}} \sqrt{a^2 z_{\text{max}}^2 + 2} + \frac{\sinh^{-1} \left( \frac{a z_{\text{max}}}{\sqrt{2}} \right)}{a}.$$  \hspace{1cm} (3.12)

Since eq. (3.12) is not easily invertible we use a numerical solver to minimize $L_{\text{sim}} - L_{\text{ana}}(z)$.

The best fitting $z_{\text{ana}}$ is used for the error estimation. Using $z_{\text{ana}}$ we can calculate the remaining two coordinates via eq. (3.11). This deviation from the analytic position $\Delta_1 = |\vec{r}_{\text{sim}} - \vec{r}_{\text{ana}}|$ is shown on the bottom left in figure 8.

Furthermore, we calculate the absolute distance from the field line as an alternative measure of the algorithm accuracy. In doing so, we simply minimize the distance between the simulated particle end position $\vec{r}_{\text{sim}}$ and the field line. This leads to the second error $\Delta_2 = \min(|\vec{r}_{\text{sim}} - \vec{r}_{\text{spiral}}|)$ which is shown on the right of figure 8.

We repeated the test for precisions $\xi$ from 1 to $10^{-9}$ and recorded the computation time for the total simulation.\footnote{The simulation was done on a single core of the Intel® Core i7-6800 chip at 3.4 GHz.} Figure 8 shows the results of this test. In the upper left the computation time is shown and it is clearly visible that there is no linear correlation between the precision and the computation time. This is due to the adaptive algorithm which uses small integration steps at low $|z|$ and increases the integration time in regions where the curvature radius is larger. In contrast, the deviation from the field line is nearly linearly correlated to the precision in the region where it is not bound by the maximum integration step (precision of 0.01 – 1). The deviation from position $\Delta_1$ saturates not only at the low precisions but also at very high precisions. This is most likely due to the restricted resolution.
Figure 8. (Upper left) Computation time needed for 10,000 trajectories. (Lower left) Deviation from the expected position of the pseudo-particles per precision. (Right) Deviation from the spiral line. It is calculated as the minimum distance between particle position and the field line. The horizontal line marks the mean, the box indicates the range of fifty percent of the particles and the whiskers give the total range.

of the numerical minimizer. This test clearly shows that the new adaptive algorithm is able to follow a field line with changing curvature radius with a user defined precision. Since the curvature radii of the Galactic magnetic field are in most cases much larger than in this test a precision of order $10^{-4}$ is most likely sufficient to receive sub-parsec errors from the propagation.

In general, it should be noted that such a large range of possible integration steps is not the most effective way to use an adaptive algorithm. A very high accuracy of order $10^{-7}$ or higher will probably lead to many (up to the order of 20 or more) consecutive step refinements. This can lead to millions of field line integrations for a single diffusion step (see section 2.3.1 for details). The user should carefully choose a feasible range of steps and a reasonable precision. Good parameters are hard to predict but small benchmark simulations will likely help to find a useful set.

3.2.2 Galactic magnetic field

From the tests described in section 3.2.1 it is known that the field line integration works. But what does that mean with respect to the Galactic environment? It is the aim of the proposed test here to evaluate whether the adaptive integration methods works on realistic Galactic field parameterizations.

The accuracy of the algorithm is supposed to be strongly correlated to the underlying magnetic field model. A simple continuously differentiable magnetic field is supposed to lead to a better result than a partly discontinuous field or one with poles. To get an estimate of the problems occurring from realistic Galactic magnetic fields, the magnetic field by Janson and Farrar (JF12-field) [40] was chosen for this test.

The magnetic field line is to be approximated with trajectory points along the magnetic field, again assuming $\kappa_\perp = 0$. In doing so, magnetic field lines of 20 kpc length in 17 pc steps have been created for 100 randomly chosen emission points in the Galaxy.
In the following, a total number of $10^5$ particles are injected into the simulation. The start positions of the particles are randomly chosen among the 100 different sources locations for which ‘best fit’ field lines exist. The end position of the particles are recorded. Now the deviation from the field line is approximated by the minimum of the differences of field line points and end positions. In this way, 100 sets of minimum distances are created with a sample size of around 1000 each. This procedure is done for five different rigidities $\rho = (10^{13} - 10^{17})$ V.

To analyze the data the mean distance $<\Delta R>$ is calculated for each rigidity. Table 5 shows that the error increases slightly with increasing rigidity. Higher rigidities imply increased mean spatial steps for the same time interval. The adaptive algorithm normally accounts for this problem by a reduction of the integration time. When the minimum integration time is reached the error cannot be reduced to the optimum which is visible here for the highest rigidities. The error does not converge to zero due to the finite resolution of the ideal magnetic field lines which leads to expected minimal error of $\Delta R_{\text{ideal}} = 4.6 - 4.7$ pc.

Although the current implementation of the JF12 field is not completely continuously differentiable the adaptive algorithm works well.

4 Cosmic ray density evolution

There are many possible scenarios to use the new code, starting with the simulation of nearby known supernova remnants (SNRs) or a backtracking of particles, originating in the solar system to identify possible source regions inside the Galaxy. We decided to perform a different approach and simulate the diffusion of cosmic rays in a global sense. In order to get an estimate of the overall cosmic ray distribution we use a continuous source distribution following the SNR distribution given in [44]. Furthermore, the realistic JF12 field, explained in section 3.2.2, is used as the regular background field for the diffusion. Different values for the diffusion parameter $\epsilon$ with $\kappa_{\perp} = \epsilon \kappa_{\parallel}$ are tested, as well as the influence of the rigidity on the density evolution.

Since we fix the energy for each simulation we do not have to include any process that might change the energy of the pseudo-particles. For a realistic, full-scale simulation of the CRs in the Galaxy this should of course be included. These processes are for example energy losses by interaction but also diffusive re-acceleration, which could in principle be implemented as a diffusion process in momentum also using stochastic differential equations.

Before we explain the results in section 4.3 the source distribution and the simulation setup are explained in section 4.1 and in section 4.2, respectively.

4.1 Source distribution

We use the same supernova distribution as in [44] where the radial distribution is originally given by Case and Bhattacharya [45]. Although this distribution just gives a rough estimate it should reflect the general features of the SNR distribution inside our Galaxy well enough.
The radial distribution is described by:

\[ f_R(r) = \frac{A_R}{R_0^2} \left( \frac{r}{R_0} \right)^2 \exp \left( -\frac{\beta r - R_0}{R_0} \right), \quad \text{with} \quad 1 = \int_0^\infty 2\pi r \cdot f_R(r) \, dr , \quad (4.1) \]

where \( \beta = 3.53 \) and \( R_0 = 8.5 \text{kpc} \) are taken from [44] and \( A_R = \beta^4 \exp(-\beta)/(12\pi) \) is the normalization constant. The distribution in z-direction is defined by:

\[ f_Z(z) = \frac{A_Z}{z_g} \exp \left( -\frac{|z|}{z_g} \right) , \quad (4.2) \]

where \( z_g = 0.3 \text{kpc} \) is chosen as the scale height of the distribution and \( A_Z = 1 \) is once more a normalization constant.

To generate the pseudo-particle start position we use a rejection sampling for \( r \) and \( z \) from eq. (4.1) and eq. (4.2), respectively. The third coordinate in cylinder coordinates, the angle \( \phi \), is drawn from a uniform distribution with \( \phi \in (0, 2\pi] \). After that the coordinates are transformed into cartesian coordinates \( x = r \cos(\phi), \quad y = r \sin(\phi) \) and \( z = z \). Our approach is different from the one in [44] since we do not define a given number of SNR positions drawn from the distribution but every particle has its own, unique origin. The sampling technique implemented for the source generation is very simple but fast enough. We can generate one million source positions in a few seconds, meaning that the generation of sources is not a significant part of the overall simulation time.

Figure 9 shows the distribution of SNRs used in the simulation. It is visible that the density has a minimum at \( r_{\text{min}} = 0 \) and maximum around \( r_{\text{max}} = 2R_0/\beta \approx 4.8 \text{kpc} \). After that the density decreases exponentially and is cut off at the boundary of the simulation volume at \( r > 20 \text{kpc} \) (see section 4.2). The rotation symmetry is clearly visible which should be kept in mind to interpret the results in section 4.3.

### 4.2 Simulation setup

In this section the other simulation parameters, beside the source distribution, are discussed in detail.

As mentioned above, the regular magnetic field component of the JF12 field [40] is used as the background field. Due to the fact that we only use the regular component we have to slightly adapt the field. In the original implementation of the field the regular field component of the field vanishes for \( |\vec{r}| \leq 1 \text{kpc} \), where the origin \( \vec{r}_0 = 0 \) is at the Galactic center (meaning \( B_{\text{reg}} = 0 \) in this region). Since this zero-field sphere has no physical reason but is due to a lack of precise observational data we continue the field to \( \vec{r}_0 = 0 \). In doing so, we neglect the hard cut off at \( |\vec{r}| = 1 \text{kpc} \) which is possible since the analytic form of the field is continuous up to \( \vec{r}_0 \).

This simulation is primarily designed to study the influence of the diffusion parameter \( \epsilon \) and rigidity \( \rho \) on the cosmic ray distribution. The diffusion tensor \( \hat{\kappa} \) is diagonal in the local frame of the magnetic field line, \( \hat{\kappa} = \text{diag}(\kappa_{\perp}, \kappa_{\perp}, \kappa_{\parallel}) \) with the parallel diffusion coefficient defined as in [10]:

\[ \kappa_{\parallel} = A\kappa_0 \left( \frac{\rho}{4\text{GV}} \right)^\alpha , \quad (4.3) \]

where \( \kappa_0 = 6.1 \cdot 10^{24} \text{m}^2/\text{s}, \alpha = 0.3 \) and \( A(\epsilon) = 1.02/(1 + 2\epsilon) \) is a normalization constant.

The normalization \( A \) is chosen to keep the trace of the diffusion tensor constant for different
values of the diffusion parameter $\epsilon$. This is necessary since the value of the trace has a huge impact on the loss time scale. Using this normalization we exclude the influence of a varying trace on the time scale and are able to analyze the influence of the diffusion parameter $\epsilon$ or rigidity $\rho$.

In every simulation setup we calculate the trajectories for about 15 million pseudo-particle forward in time until the maximum integration time $T_{\text{max}} = 100$ Mpc$/c$ or the escaping from the simulation volume at $|\vec{r}_{\text{max}}| = 20$ kpc. We use adaptive steps with $h_{\text{min}} = 0.1$ pc$/c$, $h_{\text{max}} = 1.0$ kpc$/c$ and precision $\xi = 10^{-5}$. In that way we ensure the desired accuracy for the averaging process (see below). The particle position is recorded at 1000 points in time $t_n = n\Delta t$ with $\Delta t = 100$ kpc$/c$ and $n = 1\ldots1000$. For the first analysis we use protons with a rigidity of $\rho = 10$ TV and for the second analysis we iteratively increase the proton rigidity by an order of magnitude up to $\rho = 100$ PV.

The simulation is performed on a cluster with 256 cores and takes about 3200 CPU hours per simulation set. The raw output files take a total memory storage of up to 300 GB per simulation set, depending on the escape time scale. To handle this huge amount of data we bin the pseudo-particle positions into a three-dimensional histogram with 1 kpc bin edges for each time step $t_n$ and simulation setup. This concurrently ensures that the particle density is averaged over a sufficiently large number of pseudo-particles.

4.3 Results

The first results show that the diffusion along the magnetic field lines affect the cosmic ray density evolution significantly. We can show that even a nearly unstructured source distribution is transformed into a complicated density distribution following the magnetic field structure of the Galaxy. This is in contrast to earlier simulations where the source distribution itself is already shaped in a spiral structure e.g. [10, 12]. We discuss the results for varying diffusion parameter $\epsilon$ (section 4.3.1) and rigidity $\rho$ (section 4.3.2) separately.

Figure 9. The distribution of $10^6$ random source positions. On top and on the right side the corresponding one-dimensional histograms are displayed. There is no dependence of the scale height $z_g$ on $r$ but this is just a projection effect in figure (b).
Figure 10. Face on view of the time evolution of the Galactic cosmic ray density. Here, only particles inside the Galactic disc ($-1\text{kpc} \leq z \leq 1\text{kpc}$) are displayed. From left to right the diffusion parameter $\epsilon$ is decreased. The time evolution is shown from top (early) to bottom (late). The density is given in arbitrary units on a log scale. Here, only protons with a rigidity of $E = 10\text{TV}$ are shown.

4.3.1 Diffusion parameter

Figure 10 shows different face-on views (anti-parallel to the $z$-axis) of the Galactic disc. In order to highlight the Galactic structure we display only the particles near the Galactic plane with height $|z| \leq 1\text{kpc}$. The difference in the diffusion parameter $\epsilon$ is clearly present. As expected, a decreased perpendicular diffusion coefficient emphasizes the structure of the magnetic field. In the case of very strong perpendicular diffusion $\kappa_\perp = \kappa_\parallel$ almost any structure, besides a fall off in the direction of the boundary, can be found in the density distribution.
Figure 11. Edge on view of the time evolution of the Galactic cosmic ray density. Here, all particles inside the simulation volume are displayed. From left to right the diffusion parameter $\epsilon$ is decreased. The time evolution is shown from top (early) to bottom (late). The density is given in arbitrary units on a log scale. Here, only protons with a rigidity of $E = 10$ TV are shown.

Another interesting fact is the difference in the cosmic ray density at the Galactic center. A strong perpendicular diffusion $\epsilon \geq 0.1$ leads to an increase of the Cosmic ray density. Finally, the maximum density region is shifted into the Galactic center. In contrast, strong parallel diffusion prevents the pseudo-particles from diffusion into the central region.

In addition to the analysis of the Galactic disc, we examined the cosmic ray density in the halo as well. In doing so, we combined in figure 11 all particles to produce edge-on representations of the Galaxy. As in the face-on projection the unstructured state of the cosmic ray density for (very) strong perpendicular diffusion is clear. The general trend of higher escape rates in the case of increasing perpendicular diffusion is apparent and quantified in figure 12.
Figure 12. The total particle number relative to the number of injected particles is shown for the different simulation setups. Although the total diffusion strength $\text{tr}(\kappa)$ is constant a significant difference in the escape time scale is visible.

Furthermore, especially for strong parallel diffusion, a remarkable difference between the northern and southern hemisphere can be noticed. In the North the cosmic rays are concentrated in the center for lower Galactic latitudes. Perpendicular outflows are produced as clearly visible in the figures by the halo component of the magnetic field. These outflows have shapes that resemble the x-shape structure of the magnetic field that is detected for external galaxies (e.g. [46, 47]) and their winds (e.g. [48]). In this paper, we do not perform a dedicated analysis in this direction and cannot draw any physics conclusions from these first findings. Detailed studies on the role of diffusion and advection for galactic outflows are planned in the future, but go beyond the scope of this paper. A similar structured outflow is not observed for the southern part of the Galaxy. This difference is an effect of the different magnetic field structure of the JF12 field in both hemispheres because the radial transition width of the toroidal field cannot be constrained by data for the southern hemisphere (see [40, 49] for details). This broken symmetry is not necessarily a real effect: the lack of observations for the halo field in the southern hemisphere leads to an approach to use the most simple magnetic field representation. Thus, even in the south, there might exist an x-shape-like structure, but it cannot be resolved by data yet at this point.

The escape time scale of the different diffusion configurations was mentioned above and is quantified in figure 12. This is not surprising, although the different setups have equal overall diffusion strength, but describe a completely different morphology. This difference is mainly caused by the spiral magnetic field component which is strongly aligned in the Galactic plane. Since large perpendicular diffusion coefficients $\kappa_\perp \approx \kappa_\parallel$ allow a fast escape into the halo these configuration have shorter loss times. In contrast, a dominant parallel diffusion component $\kappa_\parallel \gg \kappa_\perp$ binds the pseudo-particles strongly inside the Galactic disc preventing a fast escape. We define the effective escape time scale $T_{\text{loss}}$ by

$$\int_V n(T_{\text{loss}}) \, dV = \int_V n_0 \cdot \exp(-1) \, dV ,$$

(4.4)

as the time scale on which the total particle density decreases by $1/e$. The different time scale are given in table 6.
\[
\begin{array}{c|cccc}
\epsilon & 1.0 & 0.1 & 0.01 & 0.001 \\
T_{\text{loss}} \, [\text{Mpc}/c] & 19.2 & 23.3 & 29.4 & 32.6 \\
\end{array}
\]

Table 6. Effective escape time scale for different diffusion parameter \( \epsilon \).

**Figure 13.** Face on view of the time evolution of the Galactic cosmic ray density. Here, only particles inside the Galactic disc \((-1 \text{kpc} \leq z \leq 1 \text{kpc})\) are displayed. From left to right the rigidity \( \rho \) is increased. The time evolution is shown from top (early) to bottom (late). Here, the diffusion parameter \( \epsilon = 0.01 \) is fixed. The density is given in arbitrary units on a log scale. Please notice the different points in time and density scales compared to figure 10.

### 4.3.2 Rigidity dependence

In this section the influence of the particle rigidity on the evolution of the cosmic ray density is discussed. We fix the diffusion parameter \( \epsilon = 0.01 \) and vary the rigidity on a logarithmic scale from \( \rho = 10 - 10^5 \text{TV} \). The analysis is the same as described in section 4.3.1. High rigidities lead to a very fast particle loss, so we decided to show the time evolution on a logarithmic scale rather than on a linear scale (as in section 4.3.1).

Figure 13 shows the density distribution in the Galactic disc at \( t = (0.1, 1, 10, 100) \, \text{Mpc}/c \) for different rigidities. The very left column may help to compare the two analyses as the data for \( \rho = 10 \text{TV} \) was already used in the previous analysis. It is
visible that the simulation for different rigidities do not differ much from each other when the general topology of the density distribution is compared. But it is obvious that an increase in the particle rigidity leads to an acceleration of the escape process. The different simulations are very similar to each other but on very different time scales.

The effect of different time scales becomes even more prominent in the edge-on view shown in figure 14. At the highest rigidity the particles fill up nearly the whole halo in just a few hundred thousand years. The early time evolution reveals a feature of the cosmic ray transport that was not visible in the first analysis. The x-shape like structure which is observed for the northern hemisphere at later time is also visible in the southern hemisphere but on much shorter time scale. After a couple million years, depending on the rigidity, the diffusion in the toroidal magnetic field washes the x-shape structures out.

Figure 15 emphasizes the huge differences in the escape time scales. for rigidities above $\rho = 1$ PV it takes less than a hundred million years to lose nearly all particles to the extragalactic medium. In table 7 the escape time scales are listed for a quantified analysis. Com-
Figure 15. The total particle number relative to the number of injected particles is shown for the different simulation setups. As expected the escape time scale is strongly correlated to the rigidity of the particles.

| ρ [TV] | 10 | 100 | 1000 | $10^4$ | $10^5$ |
|--------|----|-----|------|-------|-------|
| $T_{\text{loss}}$ [Mpc/c] | 29.4 | 14.1 | 6.9 | 3.6 | 2.0 |

Table 7. Effective escape time scale for different rigidities $\rho$ and $\epsilon = 0.01$.

paring the time scales we obtain that

$$T_{\text{loss}} \propto \rho^{-0.29},$$

which reflects the rigidity dependence of the diffusion coefficient given in eq. (4.3).

5 Summary and outlook

In this section we emphasize once more the advantages of the new algorithm and recapitulate what is already possible using SDEs for Galactic propagation. In addition, we give a short outlook into the further development of simulation tasks and possible improvements of the status quo.

In principle the diffusion approach as it is implemented here is also able to simulate CRs with energies below $E \leq 10\, \text{TeV}$. The problem here is that CRPropa uses approximations for very high energies when it comes to interactions processes. For example secondary particles are ejected parallel to the parent nucleus. This makes a low energy approach not impossible but surely the interaction modules have to revised.

5.1 Summary

In this paper we show that stochastic differential equations provide an efficient mathematical tool to describe the cosmic ray transport in the diffusive regime. This method is not a completely new approach but is already established in a few propagation codes used for the simulation in the heliosphere (e.g. [19, 34]). For Galactic propagation only first examples
exist (e.g. [22, 23]). But in contrast to the approaches mentioned above we do not solve the transport equation in the global or laboratory frame. As explained in detail in section 2.3 we take advantage of the fact that the diffusion tensor is diagonal in the frame of the coherent magnetic background field. This allows us to decouple the diffusion process for the three distinct directions of the orthonormal basis defined by the comoving trihedron \( \{ \vec{e}_t, \vec{e}_n, \vec{e}_b \} \) (see eq. (2.13)). In doing so, the diffusion tensor is completely defined by two parameters, the parallel diffusion coefficient \( \kappa_\parallel \) and the diffusion ratio \( \epsilon \).

We prove that the adaptive field line integration works in simple fields (section 3.2.1) as well as realistic field descriptions (section 3.2.1). Furthermore, a method to yield stationary solutions of the transport equation by averaging the time-dependent solutions is given and explained in section 3.1.2.

In section 4 first basic simulation examples show the potential of this new simulation software. Already this very simple source distribution leads to interesting insights of the time evolution of the cosmic ray density in the Galactic halo. These need to be quantified and analyzed in future work.

We conclude that this software may help to develop a code to describe the transition region consistent in a single propagation framework. In this way systematic differences between the treatment of Galactic and extra-galactic cosmic rays can be minimized.

Finally, we want to emphasize that this propagation tool is restricted to Galactic propagation but can be used in any environment where the diffusion approximation holds. In principle, this technique may also be used for the simulation of reacceleration by the implementation of momentum diffusion.

5.2 Outlook

In this section we will give a brief outlook on the future development and possible applications of the described tool.

In the case of our own Galaxy we will examine the cosmic ray density evolution further. In doing so, we will use different source scenarios, e.g. applying discrete SNRs and not a continuous source distribution or more sophisticated source models taking the spiral structure into account (e.g. the distribution of isolated radio pulsars [50]).

In the future this new program will contribute to the solution of the gradient problem of the observed gamma-rays (e.g. [51]) with the possibility of combining dedicated propagation tools with precise interaction models. The density maps shown in section 4 are first step into the modeling of the cosmic ray gradients. The implemented anisotropic diffusion tensor will also help to understand the anisotropy of cosmic rays better.

From the technical point of view, several major extensions of the existing software are planned. The new propagation software has some advantages over the grid-based methods as mentioned above. But up to now the software is not able to take all kinds of interactions, like proton-proton or proton-nucleon inelastic scattering processes into account. Only proton-photon interaction are already implemented very efficiently in the framework. This means in effect, a complete description of the transport of cosmic rays in a dense environment like our Galaxy is not yet possible. The implementation of the missing interaction processes is one of the next steps to improve the code. In doing so, we will adapt the methods already used and tested in CRPropa for proton-photon scattering. Here, we will use Monte-Carlo methods to generate secondary particles depending on the specific cross sections (see e.g. [52, 53]) and inject them as new candidates into the simulation chain. Furthermore, density maps of the Galactic mass distribution have to implemented for a proper, location-dependent simulation.
of the spallation processes. Beside the implementation of the scattering processes three
dimensional density maps of the target material (e.g. HI and HII maps from [54]) are crucial
to derive realistic maps of neutral secondaries, such as gamma rays and neutrinos. We want
to emphasise that implementation of interactions is relevant to model the fluxes of secondary
particles like lithium, beryllium, boron, nitrogen, electrons, positrons and neutrinos. On the
other hand the flux of primaries, such as protons, will not be affected much by interactions
because the interaction probability is much smaller than unity for the majority of the CR
population. An exception would be the combined case of strong parallel diffusion ($\epsilon \lesssim 0.01$)
and low rigidities ($\rho \lesssim 100$ TV) where primary fluxes can be significantly influenced by
interactions.

To take Galactic winds or in general advection processes into account an additional term
of the transport equation has to be factored in. The advection is described by the linear term
$A$ in eq. (2.7) which means that the Euler-Maruyama Scheme in eq. (2.12) is modified by a
further term proportional to the integration time step $h$.

Up to now the diffusion tensor is locally independent in the frame of the magnetic field
line. This means that the diffusion tensor, transformed in the lab frame, has constant eigen-
values. Recent studies by Snodin and others have shown that the calculation of the diffusion
tensor depending on the local magnetic field structure and the particle’s gyro radius is possi-
bile [55]. They used test particle simulation to derive the diffusion tensor for a homogeneous
background field and different models for the random magnetic field component. They give
analytic formulas for the parallel and the perpendicular diffusion coefficient which might
serve as a good starting point for a spatially varying implementation of the diffusion tensor
in CRPropa 3.1.

The technically most challenging problem we try to solve is the development of a dy-
namic switching between the two different propagation models. As mentioned in section 1.1
the definition of parameters for the change between the propagation modes is non trivial. We
will use machine learning algorithms to find the best simulation procedure depending on the
current candidate properties, like particle type, position and rigidity. First studies may for
example compare the escape times in equal environments using the different modules. The
long term goal is to develop a software which is able to dynamically choose the best propa-
gation module depending on user set criteria. These criteria should take limited computing
resources as well as the specific physical simulation into account.

Acknowledgments

The authors would like to thank the whole CRPropa development team, especially G. Müller,
A. Dundovic and D. Walz, for their persistent help with the implementation of the algo-

rithm. We also appreciate discussions with A. Kopp and M.S. Potgieter. Some of this work
was made possible by the following python packages: NumPy and SciPy [56], Pandas [57],
matplotlib [58] and IPython [59]. We would further like to acknowledge support from the
RAPP Center (Ruhr Astroparticle and Plasmaphysics Center sponsored by the MERCUR
project St-2014-0040) and from the Research Department of Plasmas with Complex Inter-
actions (Bochum). GS is supported by the DFG through collaborative research centre SFB
676, by the Helmholtz Alliance for Astroparticle Physics (HAP) funded by the Initiative and
Networking Fund of the Helmholtz Association and by the Bundesministerium für Bildung
und Forschung (BMBF) through grant 05A14GU1.
A Spatial error resolution stationary solution

Figure 16. Relative error of the simulated density for different number of snapshots $N_{\text{snap}}$. The deviations are split into groups of bins at the top and bottom of the simulation box (orange, circle) and the inner part of the volume (green, triangles) (see figure 6(a)). Orange and green lines indicate the mean of the groups and the blue one the overall mean.
Figure 17. Relative error of the simulated density for different integration time steps $\Delta h$. The deviations are split into groups of bins at the top and bottom of the simulation box (orange, circle) and the inner part of the volume (green, triangles) (see figure 6(a)). Orange and green lines indicate the mean of the groups and the blue one the overall mean.
References

[1] IceCube collaboration, A. Achterberg et al., First Year Performance of The IceCube Neutrino Telescope, Astropart. Phys. 26 (2006) 155 [astro-ph/0604450] [SPIRE].

[2] IceCube collaboration, M.G. Aartsen et al., Observation of High-Energy Astrophysical Neutrinos in Three Years of IceCube Data, Phys. Rev. Lett. 113 (2014) 101101 [arXiv:1405.5303] [SPIRE].

[3] K. Bernloehr et al., The optical system of the HESS imaging atmospheric Cherenkov telescopes, Part 1: Layout and components of the system, Astropart. Phys. 20 (2003) 111 [astro-ph/0308246] [SPIRE].

[4] C. Baixeras et al., Commissioning and first tests of the MAGIC telescope, Nucl. Instrum. Meth. A 518 (2004) 188 [SPIRE].

[5] Pierre Auger collaboration, J. Abraham et al., Measurement of the energy spectrum of cosmic rays above $10^{18}$ eV using the Pierre Auger Observatory, Phys. Lett. B 685 (2010) 239 [arXiv:1002.1975] [SPIRE].

[6] Telescope Array collaboration, T. Abu-Zayyad et al., Energy Spectrum of Ultra-High Energy Cosmic Rays Observed with the Telescope Array Using a Hybrid Technique, Astropart. Phys. 61 (2015) 93 [arXiv:1408.1726] [SPIRE].

[7] R.U. Abbasi et al., Study of Ultra-High Energy Cosmic Ray composition using Telescope Arrays Middle Drum detector and surface array in hybrid mode, Astropart. Phys. 64 (2015) 49 [arXiv:1305.7273] [SPIRE].

[8] Z. Weng and V. Vagelli, Ams-02 measurement of cosmic ray positrons and electrons, Nucl. Part. Phys. Proc. 273 (2016) 466.

[9] A.W. Strong and I.V. Moskalenko, Propagation of cosmic-ray nucleons in the galaxy, Astrophys. J. 509 (1998) 212 [astro-ph/9807150] [SPIRE].

[10] E. Evoli, D. Gaggero, D. Grasso and L. Maccione, Cosmic-Ray Nuclei, Antiprotons and Gamma-rays in the Galaxy: a New Diffusion Model, JCAP 10 (2008) 018 [arXiv:0807.4730] [SPIRE].

[11] R. Schlickeiser and F. Jenko, Cosmic ray transport in non-uniform magnetic fields: consequences of gradient and curvature drifts, J. Plasma Phys. 76 (2010) 317.

[12] A. Shalchi, Astrophysics and Space Science Library. Vol. 362: Nonlinear Cosmic Ray Diffusion Theories, Springer, Heidelberg Germany (2009).

[13] I. B"usching, A. Kopp, F. Effenberger, R. Strauss, M. Potgieter and H. Fichtner, A stochastic approach to galactic propagation, in International Cosmic Ray Conference. Vol. 6, Beijing China (2011), pg. 252.

[14] F. Effenberger, H. Fichtner, K. Scherer, S. Barra, J. Kleiman and R.D.T. Strauss, A Generalized Diffusion Tensor for Fully Anisotropic Diffusion of Energetic Particles in the Heliospheric Magnetic Field, Astrophys. J. 750 (2012) 108 [arXiv:1202.6319] [SPIRE].
[20] C. Gardiner, *Stochastic Methods: A Handbook for the Natural and Social Sciences*, Springer Series in Synergetics, Springer, Heidelberg Germany (2009).

[21] H. Muraishi, S. Miyake and S. Yanagita, *A stochastic view of the propagation of galactic cosmic rays*, in *Proceedings of the 31st ICRC Lodz 2009*, Lodz Poland (2009).

[22] F. Effenberger, H. Fichtner, K. Scherer and I. Büsing, *Anisotropic diffusion of Galactic cosmic ray protons and their steady-state azimuthal distribution*, *Astron. Astrophys.* **547** (2012) A120.

[23] S. Miyake, H. Muraishi and S. Yanagita, *A stochastic simulation of the propagation of Galactic cosmic rays reflecting the discreteness of cosmic ray sources Age and path length distribution*, *Astron. Astrophys.* **573** (2015) A134.

[24] A. Kopp, I. Büsing, M.S. Potgieter and R.D. Strauss, *A stochastic approach to Galactic proton propagation: Influence of the spiral arm structure*, *New Astron.* **30** (2014) 32 [rsSPIRE].

[25] K.-H. Kampert et al., *CRPropa 2.0 — a Public Framework for Propagating High Energy Nuclei, Secondary Gamma Rays and Neutrinos*, *Astropart. Phys.* **42** (2013) 41 [arXiv:1206.3132] [rsSPIRE].

[26] R. Alves Batista et al., *CRPropa 3 — a public astrophysical simulation framework for propagating extraterrestrial ultra-high energy particles*, *JCAP* **05** (2016) 038 [arXiv:1603.07142].

[27] B. Wiebel-Sooth, P.L. Biermann and H. Meyer, *Cosmic rays vii. individual element spectra: prediction and data*, *Astron. Astrophys.* **330** (1998) 389 [astro-ph/9709253] [rsSPIRE].

[28] D. Allard, E. Parizot and A.V. Olinto, *On the transition from galactic to extragalactic cosmic-rays: spectral and composition features from two opposite scenarios*, *Astropart. Phys.* **27** (2007) 61 [astro-ph/0512345] [rsSPIRE].

[29] T. Stanev, P.L. Biermann and T.K. Gaisser, *Cosmic rays 4. The Spectrum and chemical composition above 10^4-GeV*, *Astron. Astrophys.* **274** (1993) 902 [astro-ph/9303006] [rsSPIRE].

[30] P.L. Biermann, J.K. Becker, A. Meli, W. Rhode, E.S. Seo and T. Stanev, *Cosmic Ray Electrons and Positrons from Supernova Explosions of Massive Stars*, *Phys. Rev. Lett.* **103** (2009) 061101 [arXiv:0903.4048] [rsSPIRE].

[31] J.R. Hördandel, *Models of the knee in the energy spectrum of cosmic rays*, *Astropart. Phys.* **21** (2004) 241 [astro-ph/0402356] [rsSPIRE].

[32] G. Giacinti, M. Kachelrieß and D. V. Semikoz, *Explaining the spectra of cosmic ray groups above the knee by escape from the Galaxy*, *Phys. Rev. D* **90** (2014) 041302.

[33] S. Thoudam et al., *Cosmic-ray energy spectrum and composition up to the ankle: the case for a second Galactic component*, *Astron. Astrophys.* **595** (2016) A33.

[34] A. Kopp, I. Büsing, R. Strauss and M. Potgieter, *A stochastic differential equation code for multidimensional Fokker-Planck type problems*, *Comput. Phys. Commun.* **183** (2012) 530.

[35] J. Giacalone and J.R. Jokipii, *The Transport of Cosmic Rays across a Turbulent Magnetic Field*, *Astrophys. J.* **520** (1999) 204.

[36] G. Qin and A. Shalchi, *Numerical Test of Different Approximations Used in the Transport Theory of Energetic Particles*, *Astrophys. J.* **823** (2016) 23.

[37] D.J. Higham., *An algorithmic introduction to numerical simulation of stochastic differential equations*, *SIAM Rev.* **43** (2001) 525.

[38] A. Marris and S. Passman, *Vector fields and flows on developable surfaces*, *Archive Rational Mech. Anal.* **32** (1969) 29.

[39] D. Ruffolo, P. Chuychai, P. Wongpan, J. Minnie, J.W. Bieber and W.H. Matthaeus, *Perpendicular Transport of Energetic Charged Particles in Nonaxisymmetric Two-Component Magnetic Turbulence*, *Astrophys. J.* **686** (2008) 1231.
[40] R. Jansson and G. R. Farrar, A New Model of the Galactic Magnetic Field, *Astrophys. J.* 757 (2012) 14 [arXiv:1204.3662] [InSPIRE].

[41] F. Effenberger, H. Fichtner, K. Scherer and I. Büsching, Anisotropic diffusion of Galactic cosmic ray protons and their steady-state azimuthal distribution, *Astron. Astrophys.* 547 (2012) 201.

[42] J.R. Cash and A.H. Karp, A variable order runge-kutta method for initial value problems with rapidly varying right-hand sides, *ACM Trans. Math. Softw.* 16 (1990) 193.

[43] T.W. Anderson and D.A. Darling, Asymptotic theory of certain “Goodness of fit” criteria based on stochastic processes, *Ann. Math. Statist.* 23 (1952).

[44] P. Blasi and E. Amato, Diffusive propagation of cosmic rays from supernova remnants in the Galaxy. I: spectrum and chemical composition, *JCAP* 01 (2012) 010.

[45] G. Case and D. Bhattacharya, Revisiting the galactic supernova remnant distribution, *Astron. Astrophys. Suppl.* 120 (1996) 437.

[46] V. Heessen, M. Krause, R. Beck and R.-J. Dettmar, The magnetic field structure in NGC 253 in presence of a galactic wind, in IAU Symposium. Vol. 259: Cosmic Magnetic Fields: From Planets, to Stars and Galaxies, K.G. Strassmeier, A.G. Kosovichev and J.E. Beckman eds., Tenerife Spain (2008), pg. 509.

[47] K. Ferrièrre and P. Terral, Analytical models of X-shape magnetic fields in galactic halos, *Astron. Astrophys.* 561 (2014) A100.

[48] C. López-Cobá et al., Star formation driven galactic winds in UGC 10043, *Mon. Not. Roy. Astron. Soc.* 467 (2016) 4951.

[49] G.R. Farrar, The Galactic magnetic field and ultrahigh-energy cosmic ray deflections, *Compt. Rend. Phys.* 15 (2014) 339.

[50] C.-A. Faucher-Giguère and V.M. Kaspi, Birth and evolution of isolated radio pulsars, *Astrophys. J.* 643 (2006) 332 [astro-ph/0512585] [InSPIRE].

[51] Fermi-LAT collaboration, M. Ackermann et al., Constraints on the Cosmic-ray Density Gradient Beyond the Solar Circle from Fermi γ-ray Observations of the Third Galactic Quadrant, *Astrophys. J.* 726 (2011) 81 [arXiv:1011.0816] [InSPIRE].

[52] M.N. Mazziotta, F. Cerutti, A. Ferrari, D. Gaggero, F. Loparco and P.R. Sala, Production of secondary particles and nuclei in cosmic rays collisions with the interstellar gas using the FLUKA code, *Astropart. Phys.* 81 (2016) 21 [arXiv:1510.04623] [InSPIRE].

[53] W.R. Webber, A. Soutoul, J.C. Kish and J.M. Rockstroh, Updated Formula for Calculating Partial Cross Sections for Nuclear Reactions of Nuclei with $Z \leq 28$ and $E \geq 150$ MeV Nucleon$^{-1}$ in Hydrogen Targets, *Astrophys. J. Suppl. Ser.* 144 (2003) 153.

[54] H. Nakanishi and Y. Sofue, Three-dimensional distribution of the ISM in the Milky Way galaxy. III. The total neutral gas disk, *Publ. ASJ* 68 (2016) 5.

[55] A.P. Snodin, A. Shukurov, G.R. Sarson, P.J. Bushby and L.F.S. Rodrigues, Global diffusion of cosmic rays in random magnetic fields, *Mon. Not. Roy. Astron. Soc.* 457 (2016) 3975 [arXiv:1509.03766] [InSPIRE].

[56] S. van der Walt, C.S. Colbert and G. Varoquaux, The numpy array: A structure for efficient numerical computation, *Comput. Sci. Eng.* 13 (2011) 22.

[57] W. McKinney, *Python for Data Analysis*, first edition, O’Reilly, Sebastopol U.S.A. (2012).

[58] J.D. Hunter, *Matplotlib: A 2d graphics environment*, *Comput. Sci. Eng.* 9 (2007) 90.

[59] F. Pérez and B.E. Granger, *Ipython: A system for interactive scientific computing*, *Comput. Sci. Eng.* 9 (2007) 21.