Astroparticle Physics 55 (2014) 37–50

Contents lists available at ScienceDirect

Astroparticle Physics

journal homepage: www.elsevier.com/locate/astropart

PICARD: A novel code for the Galactic Cosmic Ray propagation problem

R. Kissmann *

Institut für Astro- und Teilchenphysik, Leopold-Franzens-Universität Innsbruck, A-6020 Innsbruck, Austria

ARTICLE INFO

Article history:
Received 10 September 2013
Received in revised form 16 January 2014
Accepted 3 February 2014
Available online 15 February 2014

Keywords:
Cosmic Rays
Methods: numerical
Diffusion

ABSTRACT

In this manuscript we present a new approach for the numerical solution of the Galactic Cosmic Ray propagation problem. We introduce a method using advanced contemporary numerical algorithms while retaining the general complexity of other established codes. In this paper we present the underlying numerical scheme in conjunction with tests showing the correctness of the scheme. Finally we show the solution of a first example propagation problem using the new code to show its applicability to Galactic Cosmic Ray propagation.

1. Introduction

The Galactic Cosmic Ray propagation problem, i.e., the question how Cosmic Rays are transported from their sources to arbitrary locations in the Galaxy, becomes ever more relevant with recent advances in observational techniques. Such observations yield the flux of primary Cosmic Rays (see, e.g., [9,12,2,3]) or also of secondaries at Earth. For neutral secondary particles also directional information can be extracted from the data (see, e.g., [1]). Together with a physical description of the transport process of Cosmic Rays these data should allow a better understanding of the physics involved in Cosmic Ray transport.

The transport of Galactic Cosmic Rays is a diffusion-loss problem (see [15]). That is we have to find a solution of the partial differential equation:

$$\frac{\partial\psi}{\partial t} - \nabla\cdot(\mathcal{D}\nabla\psi) + \nabla\cdot(\vec{u}\psi) - \frac{\partial}{\partial p}\left(p^2 D_{pp}\frac{\partial}{\partial p}\frac{\psi}{p^2}\right) + \frac{\partial}{\partial p}\left(\dot{p}\psi - \frac{p}{3}(\nabla\cdot\vec{u})\psi\right) = s(\vec{r},p,t) - \frac{1}{\tau}\psi \quad (1)$$

with

$$\frac{1}{\tau} = \frac{1}{\tau_f} + \frac{1}{\tau_r} \quad (2)$$

where the first term on the right hand side represents the sources of Cosmic Ray species ψ, the second term gives the spatial diffusion, the third represents the energy losses and the fourth term gives losses by fragmentation and radioactive decay for the current Cosmic Ray species.

This partial differential equation has been solved using different numerical codes or analytical approximations or a mixture of both. Use of analytical solutions or approximations within a numerical code decreases the numerical cost to find a solution and gives a more direct idea of the underlying dependence of the solution on different parameters. Analytical methods, however, are not suited to investigate the Cosmic Ray propagation problem in a realistic environment, i.e., an environment, where all functions that determine the final outcome of Eq. (1) are allowed to vary arbitrarily in configuration- and momentum-space.

With the increasing precision of Galactic Cosmic Ray measurements an analytical approach is far from being able to explain the fine details in the measurements. Also a discussion of >1 TeV Cosmic Rays would necessitate consideration of the Cosmic Ray transport from individual sources. Therefore we will only discuss fully numerical methods in the paper, thus also omitting references to such numerical codes like Usine (see [11]) that use analytical approximations to improve the performance of the code. Such codes aim at finding the best values for the variables in Eq. (1) which, however, are assumed constant in the space.

For the full numerical solution of the Galactic Cosmic Ray propagation problem there are mainly two publicly available codes: Galprop (see [14]) and Dragon (see [4]). Galprop is a very sophisticated framework that tries to include all relevant physics for the propagation problem with a high complexity. The Dragon code emerged from an earlier Galprop version and has been continuously enhanced since. In particular Dragon allows a significantly more complex description for some of the transport parameters, like e.g. fully anisotropic spatial diffusion, than currently available in

* Tel.: +43 51250752069.
E-mail address: ralf.kissmann@uibk.ac.at

http://dx.doi.org/10.1016/j.astropartphys.2014.02.002
0927-6505/

Galprop – see, e.g., [6], where also the effort in establishing the transition to spatially three-dimensional simulations is shown. There are indeed some issues with the representation of the physical parameters in Galprop as is discussed in [8]. This will not be subject of the present paper. Here we rather diagnose the problem that there was far less attention directed to the numerical solver in Galprop than to other aspects of the code. This led to the fact that the solver is rather outdated regarding the numerical methods employed.

Consequently we will discuss the implementation of an up to date numerical solver within a code that can adopt the same transport parameters as Galprop, using initialisation via Galdef files. In Section 2 we describe the new numerical scheme. Corresponding tests will be discussed in Section 3 and we will show a typical example of a Galactic Cosmic Ray propagation problem in Section 4. Finally we will conclude with an outlook on ongoing development of the code.

2. A new numerical approach

As mentioned in the introduction the presently most widely used code for the solution of the Galactic Cosmic Ray transport problem is Galprop. This code was introduced some 20 years ago (see [17]) where the numerical solver has only been marginal altered since that time.

The solution in the Galprop code is computed from a Crank–Nicolson discretisation of the partial differential equation Eq. (1), where the authors use operator splitting by which they can apply the updating scheme to each spatial or momentum dimension separately. To avoid the problem of having to solve a prohibitively large amount of timesteps Galprop additionally uses a procedure where a range of different timestep sizes is used for the time integration beginning with very large steps and ending at a user-specified smallest timestep. By this the solution can reach a steady state faster than for a constant timestep method (for further details see the appendix of [13]).

This solution scheme, however, has some severe shortcomings. The first issue is that the numerical integration scheme depends on parameters to be set by the user. Such parameters are, e.g., the largest and smallest timestep, and the number each timestep size is supposed to be used for the integration. The final solution of a simulation then depends on the correct choice of these parameters. While the standard parameters might suffice for the standard Galprop runs a significant change in the configuration might lead to the necessity to come up with a corresponding new set of integration parameters. To investigate the steady state solution that has been found Galprop offers some diagnostic tools. These, however, have explicitly to be administered and also interpreted by the user. Therefore, when finding new time-integration parameters, several simulations will have to be done with different parameters until it is certain that a steady state is reached from a comparison of the results.

Most of these issues arise from the fact that a time integration scheme is used where a steady state solution is searched for. Therefore, we are using two different approaches depending on the question whether the parameters in Eq. (1) are time dependent or not. In the former case the solution is obtained by integrating Eq. (1) from some initial conditions up to the time of interest. Whenever the source term $s(\vec{r}, p, t)$, the diffusion tensor $\mathcal{D}$, the momentum loss rate $\dot{p}$ and the catastrophic loss times τ are time independent we are using a solver that yields a steady state solution without any integration in time instead. In this section we will discuss both approaches – keeping in mind that it is also a viable option to use a steady state solution to compute an initial condition for the time dependent problem.

Looking at the transport equation Eq. (1) shows that when re-acceleration is not taken into account we only have to deal with first order derivatives in momentum space. If additionally the energy changes universally occur in the same direction, the momentum space transport problem becomes particularly simple. This motivates the choice of a dedicated solver. Even though this might seem to be a special case it is a very common application in Galactic Cosmic Ray propagation simulations. By comparison with the more general solver we will later find that the solver adapted to this particular case is indeed more efficient than the general one. In the following we will refer to the different solvers as the *re-acceleration scheme* for the general solver and the *energy-loss scheme* for the special case without re-acceleration. We will introduce adapted solvers for both situations. We start by discussing the steady state problem.

2.1. The steady state problem

Looking for a steady state solution of Eq. (1) means that we are looking for a solution where the time-derivative goes to zero. Therefore a popular option is to do a time integration instead and integrate until a steady state solution is reached, i.e. until the solution does not change anymore. Depending on the choice of variables in Eq. (1) this, however, can take quite some time and also a good criterion is needed to check whether the code has found a steady state solution. In particular, one has to ask how small a change would have to be in order to indicate such a steady state. This is a particular issue for Cosmic Ray transport were the solution varies over orders in magnitude especially as a function of energy.

Here we are therefore using a different approach, where we explicitly make use of the fact that the time derivative is supposed to be zero. That is we are solving the equation

$$\begin{aligned} &-\nabla\cdot(\mathcal{D}\nabla\psi) + \nabla\cdot(\vec{u}\psi) - \frac{\partial}{\partial p}\left(p^2 D_{pp}\frac{\partial}{\partial p}\frac{\psi}{p^2}\right) \\ &+\frac{\partial}{\partial p}\left(\dot{p}\psi - \frac{p}{3}(\nabla\cdot\vec{u})\psi\right) + \frac{1}{\tau}\psi \\ &= s(\vec{r}, p, t) \end{aligned} \tag{3}$$

instead. For this equation it is not possible to use dimensional splitting anymore like, e.g., employed in the Galprop code. Solving Eq. (3) requires solving the whole equation at once. To find a numerical solution we need to discretise this equation on a grid – here we use the same approach as in Galprop, i.e., a linear spatial grid and a logarithmic grid in momentum space. Using such a discretisation the above PDE is transformed into a coupled system of algebraic equations. In 1D such a system can directly be solved by inverting the corresponding matrix (that motivates the dimensional splitting used, e.g., in Galprop), which usually is just a tridiagonal matrix. In the present case with three spatial and one momentum dimension, however, a direct solution is not efficient to compute anymore.

Therefore we are using an iterative method that relies heavily on the application of multigrid methods, which turned out to lead to excellent convergence in this case. As indicated above, we will now discuss two different implementations of the numerical solver.

2.2. Energy-loss scheme

Neglecting re-acceleration and provided that energy losses always dominate gains by adiabatic energy changes it is possible to derive an extremely efficient solution scheme for the Cosmic Ray transport problem. Due to the fact that spatial advection can