Chapter 4

Simulation Program Description

- Project's name: **sphwind**.
- Purpose:

  To study the cosmic ray transport and acceleration nearby a spherical shock in various magnetic field situations.

This project is basically modified from the earlier, prototype "wind" project (Ruffolo 1995). The word "sphwind" indicates a spherically symmetric solar wind circumstance. It contains several separate source code files which were written in the C language. The reason to have several files is that each file can be developed, modified and tested independently without reconstructing the whole program (a modular approach). In the final step, we use a "make" utility program to compile all parts of the program together and to link the program to the required libraries, providing a ready executable file

named "sphwind" to run. The *sphwind* project consists of nine files:

1. sphwind.c
2. stream.c
3. initial.c
4. field.c
5. inject.c
6. tridag.c
7. decel.c
8. printout.c
9. nrutil.c,

and each file has explicit purposes as we will describe in more detail as follows:

1. **sphwind.c:** This is the main program in which everything we need is controlled. This program obtains some necessary input variables, such as the simulation time span, Δt, $\Delta\mu$, the momentum values of particles, *etc.* The important subroutines in this main program are:

   - elements(): This subroutine calculates elements of matrices used in step().
   - step(): This subroutine controls the time evolution of the simulation program, *i.e.*, the whole μ-, r- and p-transport (adiabatic deceleration) calculated in each time step.

In addition, in the main program at each appropriate time interval, as specified by the user, the program will call the printout() routine for data output.

2. **stream.c:** This program contains a large routine called stream(). Its function is to evaluate spatial transport due to the z-advection term in Equation (2.12). The subroutines called by stream() are:

   - tvd(): the purpose is to perform spatial transport by using the concept of the generalized TVD method that we developed (Nutaro, Riyavong, and Ruffolo, 2000).
   - gen_gam(): this routine provides the Courant number, γ, for each grid point in our simulation space (p, μ, z). Recall that the Courant number is defined by $\gamma \equiv v_z \Delta t / \Delta z$.
   - lorentz(): this routine performs a Lorentz transformation used by the shock treatment process in the stream() routine.

   Furthermore, boundary conditions and particle transport nearby a shock will be treated inside the stream() routine.

3. **initial.c:** This program provides the desired initial condition which is the distribution function F at the initial time of the simulation.

4. **field.c:** This program has several routines concerning the magnetic field configuration and fluid speed flow at various distances. The subroutines

inside are:

- dzz(): This routine calculates an effective spatial diffusion coefficient, expressing the strength of the pitch angle scattering. It uses the limit of diffcoeff() in the absence of focusing.
- diffcoeff(): This routine calculates a modified scattering coefficient to account for a singularity at $\mu = 0$ (Ng and Wong, 1979; Ruffolo, 1991).
- mudot(): This routine calculates the rate of increase of μ due to adiabatic focusing (Equation (2.15)).
- arclength(): This routine converts r to z, where here $z = r - r_{sh}$.
- radius(): This routine converts z to r.
- cospsi(): This routine calculates $\cos\psi$, where ψ is the angle between the magnetic field line at a desired point to the radius vector (see Figure (2.3)).
- dzdt(): This routine calculates the Fokker-Planck coefficient following the term $\langle \Delta r/\Delta t \rangle$ (Equation (2.14)).
- relvwf(): This routine gives the solar wind velocity relative to the fixed frame.
- relvws(): This routine gives the solar wind velocity relative to the solar wind frame.
- decelrate(): This routine concerns the term $\langle \Delta p \rangle/\Delta t$, which is the

momentum loss rate (Equation (2.16)).

- enratio(): This routine calculates the ratio of the particle energy in the local fluid frame to the energy in the fixed frame (used in the diffusion term).
- f_length(): Calculates the magnetic focusing length L.
- r_updown(): Calculates the characteristic winding radius of the Archimedean spiral in the upstream or downstream region.

5. **inject.c:** This program provides a desired injection of particles during the course of the simulation, if need in the simulation (not used in this work).

6. **decel.c:** This program accommodates a deceleration process by calling routine decelrate() in field.c.

7. **printout.c:** This program provides desired output data in several formats for use in other visualizing programs. The appropriate printout of the output data format is crucial in interpreting our simulation results.

8. **tridag.c:** This program was copied from the Numerical Recipes in C (Press *et al.*, 1991), to perform a trigonal band matrix solution.

9. **nrutil.c:** This file is a collection of several useful utility routines. It was slightly modified from Numerical Recipes in C (Press *et al.*, 1991). Several routines are used to allocate and deallocate memory for vectors and matrices. It helps us to conserve memomory and increase flexibility in managing

vectors, matrices or higher-dimension arrays, since in the typical C language, the index will always start from 0. By the contribution of these routines, we can easily assign a desired range of indices. Furthermore, the routine, nrerror(), is invoked to terminate program execution – with an appropriate message – when a fatal error is encountered.