Chapter 4
Transport Simulation Methodologies

The equations explaining particle transport in our models were introduced in Chapter 3. In this chapter, the methodologies used for solving the equations will be shown. Since the equations are quite complicated, especially the pitch-angle transport equation, numerical methods are used. The pitch-angle transport equation is solved by a finite-difference method, while the diffusion-convection equation is solved by a shooting method.

4.1 Transport Simulation by Solving the Diffusion-Convection Equation

Before solving the problem in complicated cases, we would like to begin with solving the diffusion-convection-acceleration equation in the case of a shock, which is the simplest case that can be solved analytically.

In the case of a shock, equation (3.59) can be solved directly without the assumption of $F \propto p^{-\gamma}$ that simplifies equation (3.59) to equation (3.61). In equation (3.59), there is a term $(\nabla \cdot \mathbf{U})$, which is zero except at the shock. This means that in a steady state, the total z-flux is constant throughout the downstream side and the throughout the upstream side (though the downstream and upstream values are different):

$$\frac{dS_z}{dz} = 0 \quad \text{if} \quad z \neq 0, \tag{4.1}$$

where

$$S_z = U_z F_0 - D\frac{dF_0}{dz} \tag{4.2}$$

is the total z-flux (convective flux + diffusive flux). Since S_z is constant, the solution of equation (4.2) should be in the form of

$$F_0 = c_1 + c_2 \exp\left(\frac{U_z}{D} z\right), \tag{4.3}$$

where c_1 and c_2 are constants. Since in our model plasma flows from upstream ($z > 0$) to downstream ($z < 0$), U_z has a negative value. Because the particle density needs to be a finite value, in the downstream region c_2 needs to be zero to avoid the divergence of the exponential term. This means that F_0 is a constant in the downstream region, so dF_0/dz is equal to zero. Therefore, there is only convective flux in the downstream region, $S_z = U_2 F_{down}$, where U_2 is the downstream plasma flow speed and F_{down} is the downstream omnidirectional particle density. Particles are convected from upstream to downstream, and although the particles can diffuse back upstream, they cannot diffuse back very far. Assuming that the upstream high-energy particle background is very small compared with the accelerated particle population, we can set a steady-state condition $F_0 = 0$ at far upstream. By this condition, c_1 needs to be zero, so F_0 decreases exponentially in z. Furthermore, when this condition is used with equation (4.2), it also means that the total z-flux in the upstream region is equal to zero, i.e., the convective flux is balanced by the diffusive flux.

At the shock, $\nabla \cdot \mathbf{U}$ is not zero. The diffusion-convection equation for the shock region is

$$-\frac{dS_z}{dz} + \frac{(U_1 - U_2)}{3}\delta(z)\frac{\partial(pF_0)}{\partial p} = 0, \tag{4.4}$$

where U_1 is the upstream plasma flow speed and $\delta(z)$ is the delta function. Since the downstream flux is $U_2 F_{down}$ while the upstream flux is zero, we have

$$\frac{dS_z}{dz} = -U_2 F_{down} \delta(z). \tag{4.5}$$

Because F_0 is constant throughout the downstream region, we can assume that F_0 at the shock is still the same as F_{down}. After substituting equation (4.5) into equation (4.4) and integrating over z, we get

$$\frac{\partial(pF_{down})}{\partial p} = -\frac{3U_2}{U_1 - U_2} F_{down}. \tag{4.6}$$

Next, we use the product rule on the term $\partial(pF_0)/\partial p$ and integrate over p, and then we get

$$F_{down} \propto p^{-\gamma}, \tag{4.7}$$

where γ is a constant given by

$$\gamma = \frac{3U_2}{U_1 - U_2} + 1. \tag{4.8}$$

Since γ is independent of p, we can evidently see that shock acceleration yields a power-law spectrum!

In the case of a compression region, for simplicity, we use the assumption of $F \propto p^{-\gamma}$ in equation (3.59), yielding equation (3.61), but in this case $\nabla \cdot \mathbf{U}$ is not constant in z. This makes our problem more complicated, so a numerical method is used to solve it. Since equation (3.61) is an ordinary differential equation, together with z-boundary conditions far upstream and far downstream that should be the same as in the case of a shock, this equation can be solved by a shooting method (Press et al. 1988).

Beginning with rewriting equation (3.61) in terms of the flux,

$$\frac{dS_z}{dz} = -\frac{\gamma - 1}{3} (\nabla \cdot \mathbf{U}) F_0, \tag{4.9}$$

then reforming equation (4.2) yields

$$\frac{dF_0}{dz} = \frac{-S_z + U_z F_0}{D}. \tag{4.10}$$

Along with the condition $S_z = U_2 F_0$ far downstream, here we normalize F_0 far downstream to be 1 (actually this is arbitrary). When a γ value is given, equations (4.9) and (4.10) are solved to obtain F_0 and S_z for every z-step from downstream to far upstream by a fourth-order Runge-Kutta method (Press et al. 1988). As part of the shooting method to find the solution of the diffusion-convection equation, the γ value is optimized to satisfy the condition $S_z = 0$ far upstream by the secant method (Press et al. 1988).

4.2 Transport Simulation by Solving the Pitch-Angle Transport Equation

4.2.1 Numerical Method of Solving the Pitch-Angle Transport Equation for a Compression Region

In order to solve equation (3.24), which is a complicated partial differential equation, we use methods constructed on the finite-difference approximation, which approximates continuous derivatives of functions as slopes of straight lines between two certain discrete points representing the functions. To find the steady state solution, we start with an initial distribution function, $F(z, p, \mu, t = 0)$. Afterward, the distribution will be updated to be a distribution function for the next time step, $F(z, p, \mu, t = \Delta t)$. Then, we will keep updating to new time steps until the new distribution function is nearly the same as the previous distribution function.

Ideally, F would be simultaneously updated due to the processes leading to fluxes in the z, p, and μ directions. Solving that problem would be quite

difficult even if the finite-difference approximation is used. The problem can be simplified by using a technique called "operator splitting." This technique approximates a simultaneous update of F as a sequential update. If the time step for this approximation is small enough it can give an accurate result. In practice, the series of updating F from t to $t + \Delta t$ is as follows:

STEP 1: Update F due to processes leading to a flux in the μ-direction, i.e., focussing, differential convection, and pitch-angle scattering, over a time $\Delta t/2$.

STEP 2: Update F due to the process leading to a flux in the p-direction, i.e., acceleration, over a time Δt.

STEP 3: Update F due to processes leading to a flux in the z-direction, i.e., streaming and convection, over a time Δt.

STEP 4: Update F due to processes leading to a flux in the μ-direction over a time $\Delta t/2$ again to complete a time step update due to μ processes.

Note that the update due to μ-processes is divided into two parts because this treatment can give more accurate results compared with no dividing, for the same time-step size. Moreover, we already tried to split streaming and convection into a first-half time step before acceleration and a second-half time step after acceleration, but the results were the same as without splitting. Therefore, the splitting of z-processes is not needed. The reason is the streaming, convection, and acceleration might approximately commute with each other, so the p-process splitting is not needed either.

After the overview about our updating steps discussed above, some details about each step are unveiled here.

STEP 1 & STEP 4: Update F due to μ-processes over a half time step

When the entire process leading to the F update is split, we will get the process leading to F updating due to the flux in the μ-direction, which can be explained by

$$\left(\frac{\partial F}{\partial t}\right)_{\mu} = \frac{\partial S_{\mu}}{\partial \mu}, \tag{4.11}$$

where S_{μ} is the μ-flux, which is a diffusive-convective flux, given by

$$S_{\mu} = \frac{1-\mu^2}{2}\left[v\nabla\cdot\hat{\mathbf{l}} + \mu\nabla\cdot\mathbf{U} - 3\mu\, l_i\, l_j \frac{\partial U_j}{\partial x_i} - \frac{\mu v^2 \mathbf{U}\cdot\hat{\mathbf{l}}}{c^2}\nabla\cdot\hat{\mathbf{l}}\right]F$$

$$+ \frac{\varphi(\mu)}{2}\frac{\partial}{\partial\mu}\left(1 - \frac{\mu v\,\mathbf{U}\cdot\hat{\mathbf{l}}}{c^2}\right)F. \tag{4.12}$$

At $\mu \to 1$, the μ-flux is zero. That means that particle cannot flow or diffuse into the region with $|\mu| > 1$ ($\cos\theta > 1$), which is the nonphysical region. For our problems, this condition is used to be the boundary condition of the μ domain.

In this work, the equation (4.11) is numerically solved by using a finite difference method. In particular, for each z and p value we use the Crank-Nicolson method, which updates F by solving the explicit equations,

$$F(\mu)^* \leftarrow \left[F(\mu) - \left(\frac{\Delta t}{4n}\right)\left(\frac{S_{\mu}(\mu+\Delta\mu/2) - S_{\mu}(\mu-\Delta\mu/2)}{\Delta\mu}\right)\right]_{old}, \tag{4.13}$$

for each grid point of μ at the start of a calculation step, where F^* is the intermediate value of F between the old time step and the new time step. Then, for ending the calculation step, the updated F is calculated by solving the coupled implicit equations,

$$\left[F(\mu) + \left(\frac{\Delta t}{4n}\right)\left(\frac{S_{\mu}(\mu+\Delta\mu/2) - S_{\mu}(\mu-\Delta\mu/2)}{\Delta\mu}\right)\right]_{new} \leftarrow F(\mu)^*, \tag{4.14}$$

of all μ-grid points at fixed values of z and p. For both STEP 1 and STEP 4, these processes are repeated for n explicit and n implicit substeps, where n

is repeatedly doubled until the resulting $F(\mu)$ changes by less than the preset absolute tolerance or the preset relative tolerance. Each substep represents a time increment of $\Delta t/(4n)$. Hence, we can note that for each STEP 1 or STEP 4 dealing with the time evolution due to μ-process in a time interval $\Delta t/2$, the μ-process is calculated for $2n$ substeps.

STEP 2: Update F due to p-process over a time step

To update F due to our p-changing process (acceleration), the split equation explaining the evolution of F due to this p-process can be written

$$\left(\frac{\partial F}{\partial t}\right)_p = -\frac{\partial}{\partial p} p \left[\frac{1-3\mu^2}{2} l_i l_j \frac{\partial U_j}{\partial x_i} - \frac{1-\mu^2}{2} \nabla \cdot \mathbf{U}\right] F. \tag{4.15}$$

When we denote the terms in the bracket as $1/\tau_a$, which is a constant in p, the above equation (4.15) can be rewritten with shorter notation,

$$\left(\frac{\partial F}{\partial t}\right)_p = -\frac{1}{\tau_a}\frac{\partial}{\partial p} pF, \tag{4.16}$$

where τ_a, called the "acceleration time," is given by

$$\frac{1}{\tau_a} = \frac{1-3\mu^2}{2} l_i l_j \frac{\partial U_j}{\partial x_i} - \frac{1-\mu^2}{2} \nabla \cdot \mathbf{U}. \tag{4.17}$$

Moreover, when F is multiplied by p on both sides of equation (4.16), the equation can be expressed in another form,

$$\left(\frac{\partial}{\partial t} pF\right)_p = -\frac{1}{\tau_a}\frac{\partial}{\partial \ln(p/p_0)} pF, \tag{4.18}$$

where p_0 is a fixed reference momentum. This equation (4.18) is in the form of a convection equation. Therefore, the value pF is constant along characteristics, which are straight lines of slope $1/\tau_a$ in the graph of $\ln(p)$ versus t (see Figure 4.1). Mathematically, the solution of equation (4.18) can be written as

$$pF(t+\Delta t, z, \mu, p) = pe^{-\Delta t/\tau_a} F(t, z, \mu, pe^{-\Delta t/\tau_a}). \tag{4.19}$$

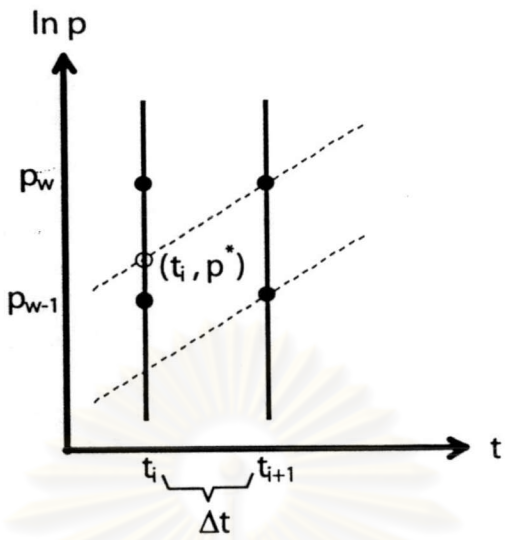

Figure 4.1: The interpolation scheme. The solid lines are at constant t. The dashes lines are characteristics, along which pF is constant. The value of $F(t_{i+1}, p_w)$ can be found from $F(t_i, p^*)$, which can be estimated by geometric interpolation between $F(t_i, p_{w-1})$ and $F(t_i, p_w)$.

Therefore, in order to find the updated F of a momentum of interest, $F(t_{i+1}, p_w)$, where i and w are the time step index and momentum step index, we need to find $F(t_i, p^*)$, where p^* can be found from p_w and the slope of the characteristics in the graph of $\ln(p)$ versus t,

$$\ln p^* = \ln p_w - (\Delta t/\tau_a), \tag{4.20}$$

$$p^* = p_w e^{-\Delta t/\tau_a}, \tag{4.21}$$

demonstrated in Figure 4.1. Then, $F(t_i, p^*)$ is evaluated by using linear interpolation between $F(t_i, p_w)$ and $F(t_i, p_{w-1})$ in log-log scale (geometric interpolation). In particular, $F(t_i, p^*)$ can be found by this relation:

$$\log F(t_i, p^*) = (1 - f_p) \log F(t_i, p_{w-1}) + f_p \log F(t_i, p_w), \tag{4.22}$$

where

$$f_p = \frac{\log p^* - \log p_{w-1}}{\log p_w - \log p_{w-1}}. \tag{4.23}$$

When considered in a normal scale, equation (4.22) can be rewritten in another form,

$$p^* F(t_i, p^*) = p_{w-1} F(t_i, p_{w-1}) \left(\frac{p^*}{p_{w-1}} \right)^{expon+1}, \tag{4.24}$$

where

$$expon = \frac{\log F(t_i, p_{w-1}) - \log F(t_i, p_w)}{\log p_{w-1} - \log p_w}. \tag{4.25}$$

Then set $p_w F(t + \Delta t, p_w, z, \mu)$ to be equal to the evaluated $p^* F(t, p^*, z, \mu)$, and the F update due to the p-process in a time interval Δt is finished.

For the step of F updating due to the p-process, our program has been improved. The acceleration time is set to be calculated once for every z and μ, and these values are stored in a 2-dimensional array instead of being calculated at each time step. This modification reduced the program running time to about 1/8 of the running time of the previous unmodified code.

STEP 3: Update F due to z-processes over a time step

When the overall process is split, the particle density evolution due to the streaming and convection processes, making particles move in the z-direction, is governed by

$$\left(\frac{\partial F}{\partial t} \right)_z = -\frac{\partial}{\partial z} \left[U_z + \mu v l_z - \frac{\mu^2 v^2 U_z l_z}{c^2} l_z \right] F. \tag{4.26}$$

With boundary conditions carefully designed for consistency with those used for solving the diffusion-convection equation, the above equation can be solved by the "total variation diminishing (TVD) method," which was first presented by Harten (1983), where the "total variation" (TV) of a function $F(z, t)$

at time t is defined by

$$TV(F(z,t)) = \int \left| \frac{\partial F(z,t)}{\partial z} \right| dz, \tag{4.27}$$

or in the form of finite differences,

$$TV(F^n) = \sum_l \left| F^n_{l+1} - F^n_l \right|, \tag{4.28}$$

where l and n are indices of space (z) and time (t) respectively. Ideally, for a convection equation, $TV(F^{n+1})$ should be equal to $TV(F^n)$; however, in a numerical simulation, numerical oscillations, which make $TV(F^{n+1}) > TV(F^n)$, might occur. In the TVD method, there is a principle that the TV of the next time step must not be greater than the TV of the previous time step:

$$TV(F^{n+1}) \leq TV(F^n). \tag{4.29}$$

This means that the TVD method does not permit any numerical oscillations that create new minima and maxima, but it still allows some numerical diffusion, which makes $TV(F^{n+1}) < TV(F^n)$.

The TVD scheme used in this work is based on Roe's superbee limiter (Roe 1983), which is a flux limiter. Among various flux limiters, this flux limiter is the one that provides the most aggressive anti-diffusion (Sweby 1984), corresponding to our goal to avoid numerical diffusion. However, usual TVD methods allow a parameter, γ, called the Courant number, only in the range of $0 \leq \gamma \leq 1$, where the Courant number is defined by $\gamma = V\Delta t/\Delta z$ and V is equal to the terms in the bracket in equation (4.26), i.e., dz/dt of particles at the μ and p of interest. In this work, we use a version of TVD that is generalized to allow $\gamma > 1$ or $\gamma < 0$ and also γ varying with position z (Nutaro et al. 2001). Therefore, with

this generalized TVD method, Δt can be set to be bigger than it was in previous TVD methods while the accuracy is still the same; thus the simulation time can be greatly reduced, approximately a hundred times faster.

In the generalized TVD method, γ can be greater than 1. This means F can be convected (actually, by both convection and streaming processes) more than one grid point in a time interval Δt. In the strategy of generalized TVD, the parameter γ is split into two parts: an integer g and a remainder γ' such that $0 \leq \gamma' < 1$. For example, if $\gamma = 2.8$, meaning F is convected for 2.8 z-grid points, here g is 2, so F is first moved forward by 2 grid points, and the remainder $\gamma' = \gamma - g$ (in this case 0.8), which is between 0 and 1, can be accounted for by the usual TVD method. Hence, F can be updated by

$$F_l \leftarrow F_{l-g} - \frac{\Delta t}{\Delta z} S'_{l+\frac{1}{2}} + \frac{\Delta t}{\Delta z} S'_{l-\frac{1}{2}}, \tag{4.30}$$

where $S'_{l+\frac{1}{2}}$ is the flux from the spatial cell l to $l+1$ due to γ', which can be calculated by

$$S'_{l+\frac{1}{2}} = V'_{l+\frac{1}{2}} F_{l-g} + \frac{1}{2} V'_{l+\frac{1}{2}} (1 - \gamma'_{l+\frac{1}{2}})(F_{l-g+1} - F_{l-g})\varphi_{l-g}, \tag{4.31}$$

where $\gamma'_{l+1/2} = \gamma'(z_l + \Delta z/2)$, $V'_{l+1/2} = \gamma'_{l+1/2}\Delta z/\Delta t$, the index $l + 1/2$ refers to the position of the cell boundary at $z_l + \Delta z/2$, and φ_l is Roe's superbee limiter given by

$$\varphi_l = \begin{cases} 0 & r_l \leq 0 \\ 2r_l & 0 \leq r_l \leq 0.5 \\ 1 & 0.5 \leq r_l \leq 1 \\ r_l & 1 \leq r_l \leq 2 \\ 2 & r_l > 2 \end{cases}, \tag{4.32}$$

and r_l is defined by

$$r_l = \frac{F_l - F_{l-1}}{F_{l+1} - F_l}. \tag{4.33}$$

4.2.2 Orbit-Tracing Treatment for Shocks

For the cases of shocks, the pitch-angle transport equation is not used for the whole range of simulation. It is used to explain particle transport just within the downstream or upstream region. For particle transport across the shock, which is a discontinuity plane, we use the particle-orbit-tracing treatment developed by Sanguansak and Ruffolo (1999).

In the simulation strategy, the particle distribution function is transformed from our mixed frame to the pure shock frame when the particle arrives at the region where $|z| \leq 2v/\omega$, where the shock is at $z = 0$, and ω is the frequency of gyration. Then with an assumption of particle gyrotropy, we simulate particle orbits across the shock. With the condition in the pure shock frame that particles have the same vector momentum immediately before and after crossing the shock, we can find the new pitch angle (θ_n) and the new gyrophase (ϕ_n) of a particle that just crossed the shock by these relations:

$$\cos\theta_n = \cos\theta_o \cos(\psi_o - \psi_n) + \sin\theta_o \cos\phi_o \sin(\psi_o - \psi_n) \tag{4.34}$$

$$\sin\theta_n \cos\phi_n = \cos\theta_o \sin(\psi_n - \psi_o) + \sin\theta_o \cos\phi_o \cos(\psi_n - \psi_o) \tag{4.35}$$

$$\sin\theta_n \sin\phi_n = \sin\theta_o \sin\phi_o, \tag{4.36}$$

where θ_o is the old pitch angle and ϕ_o is the old gyrophase of the particle just before crossing the shock, and here the new and old field angles are denoted by ψ_n and ψ_o respectively (see Figure 4.2). Afterward, when particles move out from this region, transformation to the mixed frame is used, and then the particle transport will be governed by the pitch-angle transport equation again.

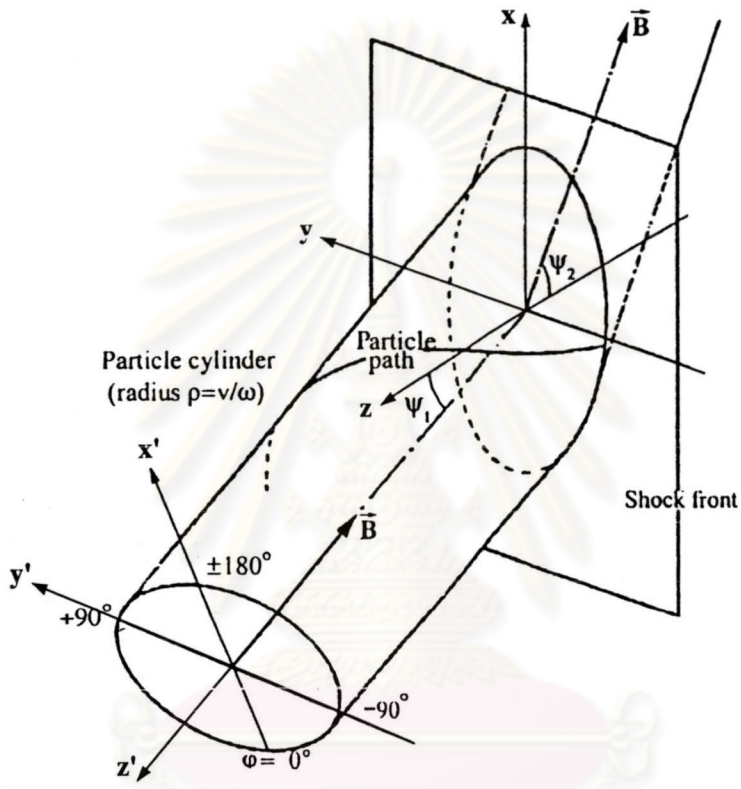

Figure 4.2: The particle helical orbit path near an oblique shock, along with the relevant definitions of coordinates. The primed coordinates will change when particles cross the shock front. (Picture credit: Leerungnavarat et al. 2000, adapted from Teresawa 1979)

4.2.3 Simulation Procedure

After the discussion about the main numerical methods, which are used to solve the pitch-angle transport equation, we would like to present our procedure, which is used throughout our simulation and shown in steps as follows:

1. Choosing Cases to Study

In this work, particle transport is simulated in many cases classified by an upstream magnetic field angle and compression width. Commonly used expressions are "parallel" (meaning $\theta_1 = \theta_2 = 0°$) and "perpendicular" ($\theta_1 = \theta_2 = 90°$). Here, when a case is specified, an upstream angle is chosen from three cases: $\tan\theta_1 = 4$ called "quasi-perpendicular", $\tan\theta_1 = 1$ called "oblique", and $\tan\theta_1 = 0.01$ called "quasi-parallel", and for each angle, a compression width is selected from $b/\lambda_{\parallel} = 0.2$, 0.5, 1.0, or 2.0 or the shock case is chosen (corresponding to a zero width compression).

2. Finding Parameters

After a case of study is chosen, some parameters for the chosen case are assessed by satisfying shock jump conditions summarized by Ruffolo (1999), based on de Hoffmann and Teller (1950):

$$\left[U_n + \frac{3}{5}\frac{u_s^2}{U_n} + \frac{u_{an}^2 t^2}{2U_n}\right] = 0, \tag{4.37}$$

$$\left[\left(U_n - \frac{u_{an}^2}{U_n}\right)t\right] = 0, \tag{4.38}$$

$$\left[\frac{3}{2}u_s^2 + \frac{1}{2}U_n^2(1+t^2)\right] = 0, \tag{4.39}$$

$$\left[\frac{u_{an}}{\sqrt{U_n}}\right] = 0, \tag{4.40}$$

where a bracketed quantity refers to the difference between that quantity on either side of the shock, U_n and u_{an} are the plasma flow speed and Alfvén speed in the shock normal direction, u_s is the sound speed, and $t = \tan\theta$, where θ is the field angle. For the case of a compression, these shock jump conditions (conservation laws) still apply when comparing conditions far upstream and far downstream. Here, we specify $u_{s1} = u_{a1} = 50$ km/s (note that $u_{an} = u_a \cos\theta$), which are typical for near-Earth interplanetary space, $U_{1n} - U_{2n} = 400$ km/s, which is typical for a near-Earth interplanetary shock, and t_1 by the tangent of the chosen upstream angle. Then we can get U_1, U_2, t_2 and the magnetic compression ratio B_2/B_1 (via t_2/t_1), which are used in our simulation. Note that some important parameters for various field angles are tabulated in Appendix C.

3. Finding the Appropriate Step Sizes

For this step of the work, we have to find the step sizes used in our simulation. They should be small enough to give accurate results, but unnecessarily small step sizes waste computer time. However, $\Delta\mu = 2/N_\mu$, where $N_\mu = 15$ is the number of μ-steps, has been tested. Even if we increase N_μ to more than 15, the results are still the same. For Δp, we use the value corresponding to $\Delta(v/U_{1n}) = 5$. This value has been also checked. Some smaller Δp values were used but they gave similar results. Moreover, an excessively small Δp can lead to some problems involving boundary conditions and numerical oscillations, while $\Delta(v/U_{1n}) = 5$ does not for any cases of interest. Therefore, the problem of finding step sizes now reduces to finding Δz and Δt for each case.

Since, for a given p, the change in F due to μ, p, and z processes, described by equations (4.11), (4.15), and (4.26), respectively, can be analytically calculated for a short-time interval when the initial $F(\mu, z)$ is set to 1 for all μ and z, we can

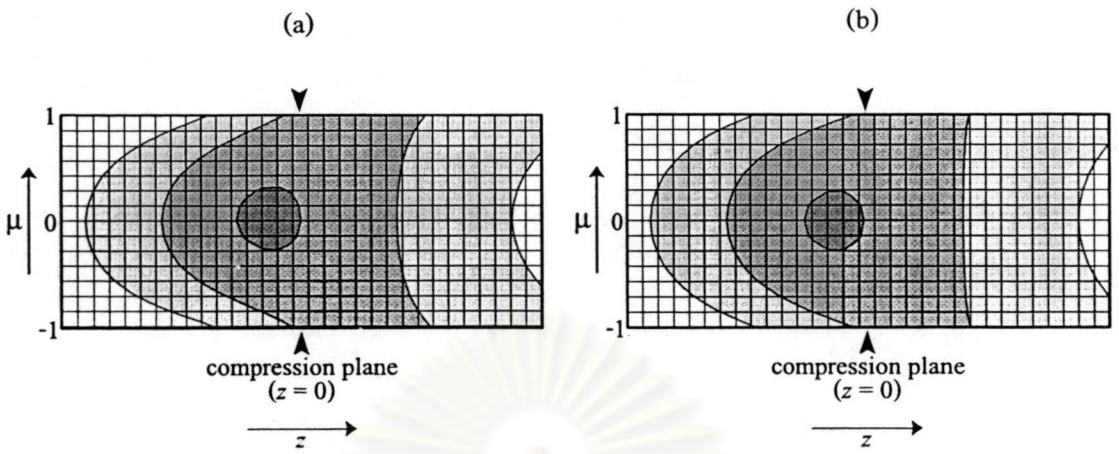

Figure 4.3: (a) An acceptable numerical ΔF (contour plot on μ-z plane) over a short time Δt, compared with (b) an analytical ΔF (contour plot on μ-z plane)

use this case to check our numerical simulations for a given Δz and a given Δt. After simulating particle transport for the case with the given Δz and Δt for a single time step, plots of ΔF from the numerical method and analytical method are compared. This method is much faster than performing a full steady-state simulation. If the given Δz and Δt yield similar results (see Figure 4.3), this should mean that Δz and Δt are accurate enough, but if not (see Figure 4.4) the Δz or Δt value will be reduced until a satisfactory comparison is obtained. In some cases we have also directly verified that reducing Δz and Δt yields the same steady state results.

4. Finding an appropriate p-boundary condition

In §4.2.1, the boundary conditions were discussed for the $\mu-$ and z-directions, but not for the p-direction because we do not exactly know what F should be for a momentum at the boundary of the simulation region. However, an approximate boundary condition can be found, and the boundary is set far

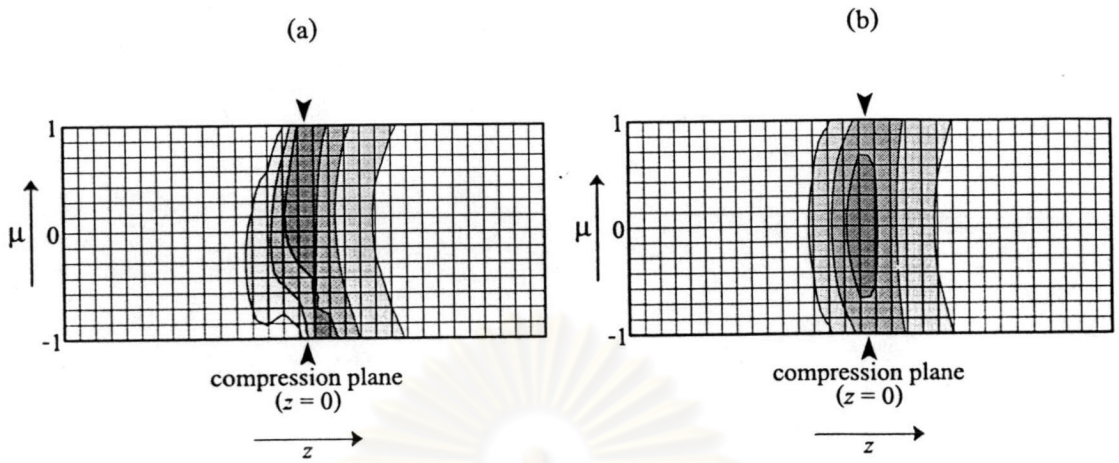

Figure 4.4: (a) An unacceptable numerical ΔF (contour plot on μ-z plane) over a short time Δt, compared with (b) an analytical ΔF (contour plot on μ-z plane)

enough that the results are not sensitive to the details of the boundary condition.

From the diffusion-convection equation, we know that in a steady state, the upstream convective flux and diffusive flux are balanced, while there is only convective flux ($S_{down} = U_2 F_{down}$) downstream. For a shock, the shock discontinuity between upstream and downstream is the only place where a p-flux exists, and the p-flux should be balanced by the convective flux flowing downstream as shown in Figure 4.5.

To find an approximate boundary condition at low p, representing particles that will be accelerated to higher p, we assume a condition $F \propto p^{-\gamma}$ at the boundary ("peg") momentum and optimize γ, called γ_p where p stand for "peg," to yield an equilibrium in the steady state. That means we find a γ_p value that gives not too much and not too little p-flux to balance the downstream convective flux. The optimized γ_p is called γ_{eq}. Afterward, $F(z, \mu)$ at the peg momentum is "pegged" to this approximate solution, which is used for the p-boundary condition.

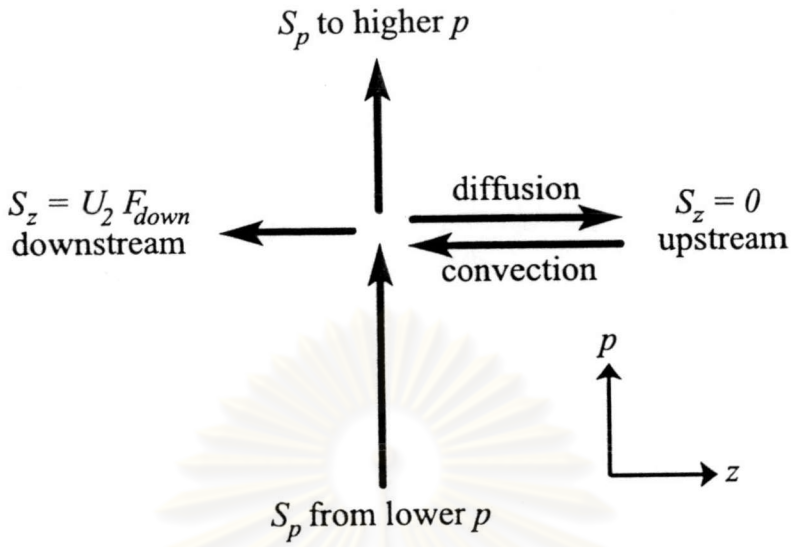

Figure 4.5: Schematic illustration of the balanced fluxes at a shock in a steady state (Picture credit: Ruffolo 1999)

5. Full Particle Transport Simulation

After the peg boundary condition is set, now we can fully simulate particle transport for many momenta without the assumption of $F \propto p^{-\gamma}$.

6. Check the Obtained Results for the Upstream Decay Rate and Anisotropy

When the full simulation results are obtained, the upstream decay rate vs. z and anisotropy at far upstream (see Appendix D) are checked against analytic results for the diffusion-convection equation for regions far from shock (or compression center), where the diffusion-convection equation should yield accurate results.

7. Boundary Dependence Checking

We do not exactly know the true boundary condition at low p, so we use an approximate condition and set the p-boundary low enough that it does not significantly affect the results. This means our simulation results should be insensitive to small changes in this particular boundary condition.

In this work, the boundary dependence of the simulation results is checked by changing γ_p in the fourth simulation step from γ_{eq} to $\gamma_{eq} \pm 0.3$, and we found that the changed boundary-condition simulation yields quite the same results as the results yielded by γ_{eq}, so the results (shown in Appendix D) should be correct. The only exception is the case of a quasi-perpendicular shock with $\gamma_p = \gamma_{eq} + 0.3$, in which the results do not converge as a consequence of too much p-flux reaching the steady state. However, the simulation results with $\gamma_p = \gamma_{eq}$ and $\gamma_p = \gamma_{eq} - 0.3$ are quite similar, so we still believe they are correct.

8. Redo the Above Seven Steps for Every Case of Interest