7.5 Project No. 5: Perpendicular diffusivity in a uniform Lorenz plasma

Introduction.-

One of the more important problems of fusion plasmas is to estimate the perpendicular (to the magnetic field) transport coefficients (diffusivities, conductivities, etc.) that may allow us to tell how fast particles or energy would leak out of a given magnetic configuration like a tokamak or stellarator. This is a extremely difficult calculation, especially if one wants to keep the mutual interactions between all electrons and ions in the plasma on top of the effect on transport of the confining magnetic field. This can be turned into an easier problem by substituting the mutual interaction by probabilistic mean-field model [2]. This simplified problem allows us to concentrate then on the effect of the magnetic geometry on transport by following a single particle at a time. At the end, we can construct a transport coefficient by averaging the behavior of the individual particle results using the expected particle distribution function (pdf) for that problem. In general this pdf is unknown but, if the system is strongly collisional, we may expect it to be a local Maxwellian and we use it to average the results[3].

The simplest problem for which this program can be carried out is a **Lorenz plasma** in a uniform magnetic field $\mathbf{B} = B_0\mathbf{k}$. The Lorenz plasma is a fully ionized, single ion species plasma. Therefore, in equilibrium quasineutrality implies that $n_i = Zn_e$, being n_s the density of electrons ($s = e$) or ions ($s = i$) and Z the ion charge number. The ions are assumed infinitely heavy and thus fixed at their positions. Electrons, however, may move and are *allowed to collide only with the fixed ions* not among themselves. In these collisions, electrons do not interchange energy with the ions (since they have infinite mass) and therefore it is only the direction of the electron velocity vector $\mathbf{v}^e$ which is changed. Typically, the (mean-field) representation of the ion-electron interaction is modeled through an $e - i$ collision frequency that depends on the electron velocity magnitude v^e as:

$$\nu_{ei}(v^e) \equiv \frac{n_i}{4\pi(v^e)^3}\left(\frac{Z^2e^4\ln\Lambda}{m_e^2\epsilon_0^2}\right), \tag{7.34}$$

where the Coulomb logarithm is roughly constant $\ln\lambda \simeq 17 - 20$.

Regarding the change in velocity direction, small deflections are the dominant result of a collision. If we imagine the space of the possible positions of the tip of the velocity

[2] In a mean-field model approach, the overall interaction due to the rest of particles is substituted by some mean expression that no longer requires to know what the other particles do!

[3] The method just described is the standard used these days to estimate the transport in tokamaks and stellarators. One might say that it is not strictly a Monte-Carlo method, in the sense that there is no randomness in the particle distribution, since it is known analytically. But randomness is still present in the random operator that models the mean-field particle interaction!

vector in the plane $X-Y$ it is going to be a circle of radius equal to the magnitude of the velocity. Position in the sphere is expressed in terms of the polar angle $\theta \in [0, 2\pi]$. To model the change in direction as a result of the collision is it enough to choose a small deflecting angle, $\Delta\theta$ and displace the tip of the vector from its original location in the circle θ_0 to $\theta_0 + \Delta\theta$. We can choose the displacements from a distribution $P(\Delta\theta)$ centered around $(0, 0)$ and with some typical width (for instance, a Gaussian). We will discuss later how to implement this process in the code.

The main consequence of the fact that electrons do not collide with themselves but only with fixed ions is that, in **local thermodynamical equilibrium** the mean velocity of the electrons must equal that of the ions, which is zero. Therefore, the local equilibrium distribution of the electrons must be given by a Maxwellian of some temperature T_e that is centered at $\mathbf{v}_e = 0$ and where $n_e = n_i/Z$ due to quasineutrality:

$$f^{\mathrm{eq}}(\vec{r}, \vec{v}_e) = \frac{n_i}{Z}\left(\frac{m_e}{2\pi T_e(\vec{r})}\right)^{3/2} \exp\left\{\frac{m(v^e)^2}{2T_e(\vec{r})}\right\}. \tag{7.35}$$

Method.-

The **diffusivity** is a transport coefficient that tells us that, for *a small departure of local thermodynamical equilibrium and at constant electron temperature*, the flux of electrons that appears to try to move the system back to equilibrium is given by:

$$\mathbf{\Gamma}^e_\perp = -D_\perp(T_e)\nabla_\perp n_e. \tag{7.36}$$

One easy way to estimate the diffusivity $D_\perp(T_e)$ is by using the well-known **Kubo formula**. This tells us that the diffusivity:

$$D = \frac{1}{2}\lim_{t\to\infty}\frac{d}{dt}\left[\left\langle|\Delta\mathbf{r}|^2\right\rangle\right], \tag{7.37}$$

being $\Delta\mathbf{r}(t)$ the displacement vector of the particle from its initial location $\mathbf{r_0}$ and with $\langle\cdot\rangle$ denoting the (ensemble) average over all possible initial positions (i.e., $\mathbf{r_0}$) and initial velocities ($\mathbf{v_0}$). Since we are considering a uniform ion distribution (n_i is constant) in a uniform magnetic field, the system is spatially invariant and thus, the average over $\mathbf{r}_0$ is redundant. We can simply start all particles from $\mathbf{r}_0 = 0$ and average over all possible initial velocities. However, note that particles have a particular v_0 value distributed according to a Maxwellian distribution. It is therefore sufficient with computing D for several (meaningful) values of v_0, and then average them using as weight the Maxwellian distribution.

Implementation.-

The way to go is to initialize our particle at $\mathbf{r}_0 = 0$ with some random velocity vector $\mathbf{v}_0 = v_x^0\mathbf{i} + v_y^0\mathbf{j}$. Note that in the temporal evolution of the particle the total velocity $v_0 = \sqrt{v_{0x}^2 + v_{y0}^2}$ remains constant. Since later, we will assume that the particle pdf will be a Maxwellian with temperature T_e within a range of values, we need to choose the initial velocities v_0 so that we can compute later a meaningful average for all those temperatures without having to run more particles. Recalling that for each T_e the mean thermal velocity of the particles is simply $v_e^T = (3K_BT_e/2m_e)^{1/2}$, it makes thus sense to normalize all velocities to the maximum thermal velocity of interest before actually doing any coding. In this way, we make sure that the information we are going to obtain will be meaningful for the whole range of temperatures of interest. It is then sufficient with running the code for $v_0 = v_e^T, v_e^T/2, v_e^T/4, v_e^T/8, \cdots, 2v_e^T, 3v_e^T, 4v_e^T, cdots$.

The way to choose the initial velocities is to choose one of the two components, say v_{x0} randomly between 0 and v_0. Then, $v_{y0} = \sqrt{v_0^2 - v_{x0}^2}$. One should repeat the runs several times for each value of v_0 with different v_{x0} and average the results to obtain the diffusivity for that particular value of v_0.

To calculate the squared displacement we need to integrate the trajectory of the particle in the XY-plane perpendicular to the magnetic field. The equations of motion are simply:

$$m\frac{d\mathbf{v}}{dt} = q\mathbf{v} \times \mathbf{B} \tag{7.38}$$

where m and q are the mass and charge of the particle respectively and $\mathbf{B}$ is the magnetic field. Details about how to do this integration were already given in the Project for Chapter 2. But the important thing to take into account is that the particles will describe a circular orbit in the XY with a radius equal to the Larmor radius $\rho_L = v_e/\Omega_e$, where $\Omega_e = |q|B_0/m$ is the cyclotron frequency until a collision with an ion changes its velocity vector.

To model collisions, integrate the motion of the particle for an amount of time $\Delta t = 1/\nu_{ei}$. Then pick up a deflection angle $\Delta\phi$ from a narrow Gaussian distribution centered at $\Delta\phi = 0$ with width Δ_0. To update the velocity, simple write it in complex notation,

$$V = V_x + iV_y. \tag{7.39}$$

Then rotating it by an angle $\Delta\theta$ is simply to multiply this complex number by $\exp(i\Delta\theta)$, which gives,

$$\begin{aligned} V' = (V_x + iV_y) \cdot (\cos\Delta\theta + i\sin\Delta\theta) = \\ = (V_x\cos\Delta\theta - V_y\sin\Delta\theta) + i(V_x\sin\Delta\theta + V_y\cos\Delta\theta) = V_x' + iV_y'. \end{aligned} \tag{7.40}$$

Using the new velocity vector, $\mathbf{v}' = V_x'\mathbf{i} + V_y'\mathbf{j}$, we simply continue the integration for another Δt time. Then repeat the collision process.

Of course, recall that the objective of this simulation is to find what $|\Delta\mathbf{r}|^2(t)$ is on average, so remember to write out its value at each iteration. In fact, to avoid the output file being too long, it may make sense to do it also every ν_{ei}^{-1} steps), since it is only at collisions that the position of the particle changes importantly.

Regarding boundary conditions, it makes sense to use a periodic domain in both x and y for this calculation to avoid dealing with the boundaries. But make sure to choose the periodicities L_x and L_y much larger than the characteristic length of the process, which is given by the Larmor radius ρ_L.

Generating the Gaussian distribution.-

Generating values distributed according to a Gaussian law,

$$p(x) = (2\pi)^{-1/2} \exp\left(-\frac{x^2}{2}\right) \tag{7.41}$$

is not easy. Note that this is a Gaussian of width 1 and centered at $x_0 = 0$. If we want a Gaussian centered at x_0 with width σ we simply need to do the transformation,

$$x' = x_0 + \sigma x. \tag{7.42}$$

But who do we generate x to begin with?

To start, let's recall that when transforming between variables, the conservation of the total probability requires, for two dimensions that:

$$p(x,y)dxdy = u(r,s)drds \quad \Rightarrow \quad p(x,y) = u(r,s)\left|\frac{\partial(r,s)}{\partial(x,y)}\right|, \tag{7.43}$$

where the determinant is the Jacobian of the transformation:

$$\left|\frac{\partial(r,s)}{\partial(x,y)}\right| = \frac{\partial r}{\partial x}\frac{\partial s}{\partial y} - \frac{\partial s}{\partial x}\frac{\partial r}{\partial y}. \tag{7.44}$$

To apply this to a Gaussian, we consider $2\pi r$ as angles obtained from a uniform random distribution r, and x and y as Cartesian coordinates that will have a Gaussian distribution. The two are related by,

$$x = \sqrt{-2\ln r}\cos(2\pi s), \qquad y = \sqrt{-2\ln r}\sin(2\pi s) \tag{7.45}$$

The inversion of this mapping gives a Gaussian distribution,

$$r = \exp(-(x^2+y^2)/2); \qquad s = (2\pi)^{-1}\tan^{-1}(y/x), \tag{7.46}$$

with Jacobian,

$$J = -(2\pi)^{-1} \exp(-(x^2 + y^2)/2). \tag{7.47}$$

Therefore, it is very easy to generate our x and y distributed as Gaussian distributions (one of them can be used to generate the deflection angle $\Delta\theta$ that we need!). Start with two random numbers r and s obtained from any standard uniform distribution in $[0, 1]$. Then use Eq. 7.45 to obtain x and y, which will be distributed around $(x_0, y_0) = (0, 0)$ with widths $(\sigma_x, \sigma_y) = (1, 1)$. And if we need distributions centered at $(x_0, y_0) \neq (0, 0)$ and $(\sigma_x, \sigma_y) \neq (1, 1)$, simply make:

$$x' = x_0 + \sigma_x x \qquad y' = y_0 + \sigma_y y. \tag{7.48}$$

Objectives.-

Run the code for a sufficiently large number of particles (say, 10 particles) with different velocities v_0 (say 15 values), sampled so that both sub-thermal and suprathermal velocities are considered within the range of temperatures T_e of interest (we may choose, for instance, $K_B T_e$ in the range of 10-100 eV). Then, pick up a value of T_e and average the results obtained for each v_0 with a Maxwellian distribution with that T_e. What is the value of the diffusivity obtained using Eq. 7.37? How does it scale with temperature? Is the value consistent with the expression,

$$D_\perp \sim \left\langle \rho_L^2 \nu_{ei} \right\rangle, \tag{7.49}$$

that assumes that the characteristic length of the process is the Larmor radius and the characteristic time the inverse collision frequency?