

NPAC Computing projects: Heavy-Ions collision with the Vlasov equation

1 Formalism

1.1 Vlasov equation

The aim is to simulate a low-energy collision of two nuclei (energy per nucleon less than the Fermi energy ~ 30 MeV) with the help of the Vlasov equation:

$$\frac{\partial f}{\partial t} + \frac{\vec{p}}{m} \cdot \vec{\nabla}_r f - \vec{\nabla} U \cdot \vec{\nabla}_p f = 0. \quad (1)$$

In this equation, $f(\vec{r}, \vec{p}, t)$ is a Wigner function, which can be interpreted as a distribution function in phase space, $U(\vec{r}, t)$ is the mean-field potential, and m is the mass of a nucleon. The starting point is an initial distribution $f(\vec{r}, \vec{p}, 0)$ which represents two nuclei approaching each other with an impact parameter b . Then the distribution evolves according to the Vlasov equation, which allows one to find the distribution after the collision.

1.2 Definitions and properties

One shall recall the definition of a Wigner-function

$$f(\vec{r}, \vec{p}) \equiv \frac{1}{\hbar\pi} \int \psi(\vec{r} + \vec{s})^* \psi(\vec{r} - \vec{s}) e^{\frac{2i\vec{p} \cdot \vec{s}}{\hbar}} d\vec{s}$$

This **real** valued distribution is establishing the link between the wave-functions solutions of a Schrodinger-type equation and a quasi-probability distribution in phase-space. (This can be related to The Wronskian in Classical Mechanics). It is also interesting to note that $f(\vec{r}, \vec{p})$ is Galilean-covariant but **not** Lorentz-covariant.

The Vlasov equation as written above describes only the time-evolution under the effect of the mean field U and does not account for collisions between individual nucleons. At higher energies, these collisions become more and more important and it is necessary to go beyond the Vlasov equation (Boltzmann-Langevin equation with collision term). Furthermore Coulomb interaction shall be neglected for a first approximation.

2 Numerical implementation

A common method for solving the Vlasov equation (a partial differential equation in seven dimensions), is the “test particle method”: The continuous distribution function $f(\vec{r}, \vec{p}, t)$ is replaced by a large but finite number of delta functions,

$$f(\vec{r}, \vec{p}, t) = \frac{1}{N} \sum_{i=1}^{NA} \delta(\vec{r} - \vec{r}_i(t)) \delta(\vec{p} - \vec{p}_i(t)), \quad (2)$$

where A is the total number of nucleons, N is the number of test particles per nucleon, and \vec{r}_i and \vec{p}_i are the coordinates and momenta of the test particles. Inserting this expression for f into the Vlasov equation (1), one finds that each test particle must follow a trajectory which is the solution of the classical equations of motion

$$\dot{\vec{r}}_i(t) = \frac{\vec{p}_i(t)}{m}, \quad \dot{\vec{p}}_i(t) = -\vec{\nabla}U(\vec{r}_i(t), t). \quad (3)$$

For the mean field U , a simple function of the total density will be used [1]:

$$U(\rho) = -356 \frac{\rho}{\rho_0} + 303 \left(\frac{\rho}{\rho_0} \right)^{7/6} \quad (\text{MeV}), \quad (4)$$

where $\rho_0 = 0.17 \text{ fm}^{-3}$ is the saturation density. In practice, it is impossible to calculate the density ρ corresponding to the distribution function (2),

$$\rho(\vec{r}, t) = \int d^3p f(\vec{r}, \vec{p}, t). \quad (5)$$

It is evident that, in order to obtain a reasonable result, one has to average the density over a certain volume. There exist different methods to do this:

The first method consists in dividing the space into small cubic cells of volume L^3 and to count the number of test particles $N_{n_1 n_2 n_3}$ in each cell. The simplest way would be to define the density at $\vec{r}_{n_1 n_2 n_3}$ as $N_{n_1 n_2 n_3} / (NL^3)$. However, it is better to average over several neighbouring cells with appropriate weights in order to obtain a smoother function $\rho(\vec{r})$:

$$\rho_{n_1 n_2 n_3} = \rho(\vec{r}_{n_1 n_2 n_3}) = \frac{1}{NL^3} \sum_{k_1 k_2 k_3 = -k_{max}}^{k_{max}} w_{k_1} w_{k_2} w_{k_3} N_{n_1+k_1, n_2+k_2, n_3+k_3}, \quad (6)$$

with, e.g., $k_{max} = 1$, $w_0 = 0.5$, $w_{\pm 1} = 0.25$. The force, $-\vec{\nabla}U$, can be obtained numerically. For example, the derivative $\partial U / \partial x$ can be approximated by

$$\left(\frac{\partial U}{\partial x} \right)_{\vec{r}_{k_1 k_2 k_3}} \approx \frac{U(\rho_{k_1+1, k_2, k_3}) - U(\rho_{k_1-1, k_2, k_3})}{2L}. \quad (7)$$

Another method, which is a bit more sophisticated, consists in replacing the functions $\delta(\vec{r} - \vec{r}_i(t))$ in Eq. (2) by Gaussians $g(\vec{r} - \vec{r}_i(t))$ with a certain width,

$$f(\vec{r}, \vec{p}, t) = \frac{1}{N} \sum_{i=1}^{NA} g(\vec{r} - \vec{r}_i(t)) \delta(\vec{p} - \vec{p}_i(t)). \quad (8)$$

with

$$g(\vec{r}) = \frac{e^{-r^2/(2d^2)}}{(\sqrt{2\pi} d)^3}, \quad (9)$$

In this case, the density is given by

$$\rho(\vec{r}) = \frac{1}{N} \sum_{i=1}^{NA} g(\vec{r} - \vec{r}_i). \quad (10)$$

It is clear that the ansatz (8) does not exactly solve the Vlasov equation (1): the width and even the form of the Gaussians would have to change as a function of time. But after integration of Eq. (1) over \vec{r} , one finds that Eq. (1) is at least satisfied in average if the acceleration of the centre of each Gaussian is given by the force averaged over the Gaussian, i.e. (cf. [2, 3] for more rigorous arguments):

$$\dot{\vec{p}}_i(t) = - \int d^3 r' g(\vec{r}_i(t) - \vec{r}') \vec{\nabla}' U(\vec{r}', t) = - \vec{\nabla} \bar{U}(\vec{r}_i(t), t), \quad (11)$$

with

$$\bar{U}(\vec{r}, t) = \int d^3 r' g(\vec{r} - \vec{r}') U(\rho(\vec{r}', t)) \quad (12)$$

and therefore

$$\vec{\nabla} \bar{U}(\vec{r}, t) = - \frac{1}{d^2} \int d^3 r' (\vec{r} - \vec{r}') g(\vec{r} - \vec{r}') U(\rho(\vec{r}', t)). \quad (13)$$

In practice, when using this method, one has to introduce a grid $r_{n_1 n_2 n_3}$, too, in order to store $\rho_{n_1 n_2 n_3} = \rho(\vec{r}_{n_1 n_2 n_3})$ and/or $U_{n_1 n_2 n_3} = U(\rho_{n_1 n_2 n_3})$. Of course, the spacing L between the mesh points must be smaller than the width of the Gaussians, such that it is possible to replace the integral in Eq. (13) by a sum:

$$\vec{\nabla} \bar{U}(\vec{r}, t) = - \frac{L^3}{d^2} \sum_{n_1 n_2 n_3} (\vec{r} - \vec{r}_{n_1 n_2 n_3}) g(\vec{r} - \vec{r}_{n_1 n_2 n_3}) U_{n_1 n_2 n_3}. \quad (14)$$

In order to reduce the calculation time, one should keep only terms with $|\vec{r} - \vec{r}_{n_1 n_2 n_3}| \lesssim 3d$ (and analogously for the calculation of $\rho_{n_1 n_2 n_3}$).

Having calculated the force, one has now to solve the differential equations (3). There are algorithms whose error is of the order $(\Delta t)^2$ (where $\Delta t = t_{n+1} - t_n$ is the time step) although the force is evaluated only once per time step. The trick is to calculate the force $\vec{F} = -\vec{\nabla} U$ in the middle of the time step. For example, we give here the ‘‘velocity Verlet’’ algorithm:

$$\vec{r}_i^{(n+1)} = \vec{r}_i^{(n)} + \frac{\vec{p}_i^{(n)}}{m} \Delta t + \frac{1}{2} \frac{\vec{F}_i^{(n)}}{m} (\Delta t)^2, \quad (15)$$

$$\vec{p}_i^{(n+\frac{1}{2})} = \vec{p}_i^{(n)} + \frac{1}{2} \vec{F}_i^{(n)} \Delta t, \quad (16)$$

$$\vec{F}_i^{(n+1)} = - \vec{\nabla} U(\vec{r}_i^{(n+1)}, t_{n+1}), \quad (17)$$

$$\vec{p}_i^{(n+1)} = \vec{p}_i^{(n+\frac{1}{2})} + \frac{1}{2} \vec{F}_i^{(n+1)} \Delta t. \quad (18)$$

3 Nucleus at rest

Before simulating the collision of two nuclei, one has to show that the numerical code can describe a nucleus at rest. It is easy to show that, if the distribution function is given by the Thomas-Fermi approximation

$$f_{TF}(\vec{r}, \vec{p}) = \frac{4}{(2\pi\hbar)^3} \theta\left(E_F - \frac{p^2}{2m} - U(\vec{r})\right), \quad (19)$$

the Vlasov equation reduces to $\dot{f}_{TF} = 0$. The factor 4 accounts for the spin-isospin degeneracy which allows one to put four nucleons per state, one state corresponding to a phase-space volume $h^3 = (2\pi\hbar)^3$.

Hence, such a distribution describes a nucleus at rest. The corresponding density is given by

$$\rho_{TF}(\vec{r}) = 4 \frac{[2m(E_F - U(\vec{r}))]^{3/2}}{6\pi^2\hbar^3}. \quad (20)$$

With $U(\vec{r}) = U(\rho(\vec{r}))$, the solution would simply be a homogeneous sphere with density ρ_0 and radius $R = R_0 A^{1/3}$, where $R_0 = (3/4\pi\rho_0)^{1/3} = 1.12$ fm. However, as later in the Vlasov calculation the potential U will be replaced by the smeared potential \bar{U} , this solution is not stationary and gives rise to radial oscillations of the nucleus. In order to avoid this effect, it is necessary to smear out a bit the surface of the nucleus. The best solution would be to look for a self-consistent solution of eq. (20) with U replaced by \bar{U} .

Once one has determined the density $\rho(\vec{r})$, one can initialise the coordinates \vec{r}_i and momenta \vec{p}_i of the test particles. First, one initialises the \vec{r}_i according to the density $\rho(\vec{r})$: One takes a random vector \vec{r} and calculates the density $\rho(\vec{r})$. Then, the vector \vec{r} is kept with probability $\rho(\vec{r})/\rho(0)$ (the density has its maximum at the centre of the nucleus), but with probability $1 - \rho(\vec{r})/\rho(0)$ it is rejected and one starts again with another random \vec{r} . Then, the momenta \vec{p}_i are determined as follows: For a test particle at a position with density ρ , the absolute value of \vec{p}_i lies between 0 and $p_F = \hbar(\frac{3}{2}\pi^2\rho)^{1/3}$.

After all the \vec{r}_i and \vec{p}_i have been initialised, one can run the Vlasov code to see whether the distribution of the test particles remains stationary (although the test particles move individually). In order to test this, one can look at the density profiles as a function of time. A more quantitative test is to look at the time dependence of certain expectation values, e.g.,

$$\langle r^2 \rangle = \frac{1}{NA} \sum_{i=1}^{NA} r_i^2, \quad \langle p^2 \rangle = \frac{1}{NA} \sum_{i=1}^{NA} p_i^2, \quad \text{etc.} \quad (21)$$

4 Collective oscillations

Now one can try to describe collective excitations of the nucleus. For example, in order to excite the monopole oscillation, one can change all momenta according to $\vec{p}_i \rightarrow \vec{p}_i + C\vec{r}_i$ with some (small) constant C . In the same way, in order to excite a quadrupole oscillation, one changes the momenta according to $(\vec{p}_i)_x \rightarrow (\vec{p}_i)_x + C(\vec{r}_i)_y$ and $(\vec{p}_i)_y \rightarrow (\vec{p}_i)_y + C(\vec{r}_i)_x$.

5 Collisions of two nuclei

Finally, in order to simulate the collision of two nuclei with masses A_1 and A_2 , one first initialises both nuclei independently of each other. In order to displace a nucleus as a whole, it is sufficient to change all coordinates according to $\vec{r}_i \rightarrow \vec{r}_i + \vec{d}$. Similarly, it is given a kick by replacing: $\vec{p}_i \rightarrow \vec{p}_i + m\vec{v}$. In this way, one can easily prepare two nuclei at an arbitrary distance with some relative velocity which lets them collide. A systematic study of the conserved quantities during the collisions will be appreciated.

References

- [1] J. Aichelin and G. Bertsch, Phys. Rev. C **31**, 1730 (1985).
- [2] B. Remaud, F. Sébille, C. Grégoire, and L. Vinet, Journal de Physique **48**, supplément au n° 6, C2-195 (1987).
- [3] T. Fennel, G.F. Bertsch, and K.-H. Meiwes-Broer, Eur. Phys. J. D **29**, 367 (2004).