Simulating Collisional Dark Matter

Javier Alejandro Acevedo Barroso

September 23, 2018

Contents

Chapter 1

Objetives

1.1 General Objetive

To simulate the phase space of a collisional dark matter fluid using a Lattice-Boltzmann Method

1.2 Specific Objetives

- To implement a Lattice-Boltzmann simulation using a 4-dimensional phase space and a varying collisional term.
- To implement a Lattice-Boltzmann simulation using a 6-dimensional phase space and a varying collisional term.
- To study the dynamical behavior of a dark matter fluid using different equilibrium distributions in the collisional term.
- To compare the phase space of a collisional dark matter fluid with its collisionless version.

Chapter 2

Introduction

In this work, we simulate the phase space of a collisional dark matter fluid, for that, it is essential to know some concepts and computational techniques. In this chapter, we present all the necessary knowledge for the understanding (and development) of this work.

2.1 Dark Matter, or the Missing Mass Problem

Modern cosmology describes the universe as being composed of two fundamental types of energy: dark energy and matter[1], with dark energy being associate with a cosmological constant and matter being divided into two categories: dark matter and standard model matter[2]. The energy density in the universe is 69% dark energy and 31% matter.

Standard model matter is all the particles whose interactions can be properly described by the standard model, that includes: Protons, Electrons, Atoms and naturally, any structure they form, like Humans or Stars. On the other hand, dark matter is all the matter we measure from astrophysical sources which cannot be explained by baryonic matter. We know of the existence of dark matter entirely from astrophysical evidence, during this section we are going to do an historical review of such evidence.

2.2 Types of Dark Matter

2.3 The Boltzmann Equation

2.4 Lattice Automata and Lattice Boltzmann

2.5 BGK Approximation

[1] In relativity, mass and energy are equivalent.

[2] Which is very often called "Baryonic matter" due to Baryions being the largest fraction of this mass.

Chapter 3

The Lattice Boltzmann Algorithm

3.1 General Description

As previously asserted, the heart of the Lattice-Boltzmann Algorithm lies on its discretization of the phase space[1] [2]. To discretize the phase space, one must first choose the region to simulate. In this work, we name the extremal values in the w axis of the phase space W_{min} and W_{max}. Then, one has to fix either the size of the grid or the size of the lattice. We named the size of the grid in the w axis N_w (i.e. N_x or N_{vz}). The size of the lattice in the w axis (dw) and the extremal values are related by:

$$dw = \frac{W_{max} - W_{min}}{N_w} \tag{3.1}$$

In this work we are going to use the density-velocity phase space. Now that we have properly defined the phase space grid, we can proceed to the initialization. For simplicity, we choose gaussian initial conditions given by:

$$f(\mathbf{r}, \mathbf{v}, 0) = A \exp\left\{ -\frac{\mathbf{r}^2}{\sigma_r^2} - \frac{\mathbf{v}^2}{\sigma_v^2} \right\} \tag{3.2}$$

Where $f(\mathbf{r}, \mathbf{v}, 0)$ is the initialization of the phase space, A is an indirect measure of the total mass in the system, $\mathbf{r}$ is the vector $= (x, y, z)$, $\mathbf{v}$ is the vector $= (vx, vy, vz)$, and σ_r and σ_v are a measure of the width of the gaussian profile in the given axis. After initialization, the system evolves by classical mechanics and the modelling of the collisional step, which can be seen more easily in figure 3.1

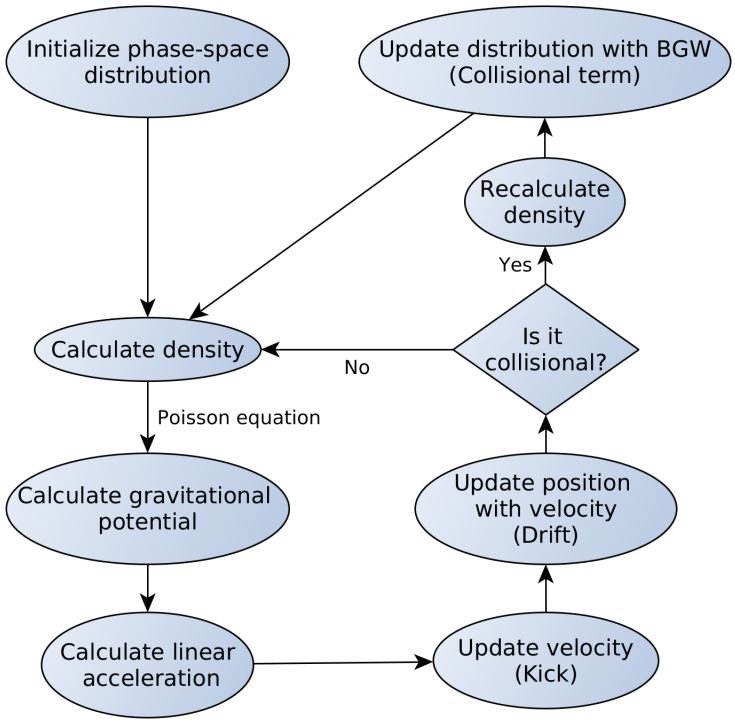

Figure 3.1: Flowchat of the algorithm.

Given the density-velocity phase space, the spatial density of matter is given by the integral:

$$\rho(\mathbf{r}, t) = \int_{-\infty}^{\infty} f(\mathbf{r}, \mathbf{v}, t)\, \mathrm{d}\mathbf{v}.$$

Which, when evaluating in the lattice becomes:

$$\rho(\mathbf{r}, t) = \sum_{\mathbf{V}_{min}}^{\mathbf{V}_{max}} f(\mathbf{r}, \mathbf{v}, t)\, \mathrm{d}\mathbf{v} \tag{3.3}$$

and during initialization:

$$\rho(\mathbf{r}, 0) = \sum_{\mathbf{V}_{min}}^{\mathbf{V}_{max}} f(\mathbf{r}, \mathbf{v}, 0)\, \mathrm{d}\mathbf{v} \tag{3.4}$$

Once we have calculated the density, we can use the Poisson equation to calculate the potential due the gravitational force[2]

$$\nabla^2 \Phi(\mathbf{r}, t) = 4\pi G \rho(\mathbf{r}, t) \tag{3.5}$$

To solve the Poisson equation we use the pseudo-spectral Fourier method, which allows for a very fast numerical solution by making use of the Fast Fourier Transform algorithm. The idea is

simply to apply a Fast Fourier Transform (FFT), then solve the equation in the Fourier space, and then apply an inverse transform (IFFT). In the phase space the Poisson equation is given by:

$$\lambda_{\mathbf{k}}^2 \hat{\Phi}(\mathbf{k}, t) = 4\pi G \hat{\rho}(\mathbf{k}, t) \tag{3.6}$$

Where $\hat{g}(\mathbf{k}, t)$ is the Fourier transform of $g(\mathbf{r}, t)$, and $\lambda_{\mathbf{k}}$ is a constant that depends on the size of the lattice and the wavevector $\mathbf{k}$. $\lambda_{\mathbf{k}}$ is calculated according to the approximation scheme used to solve the equation, here we use the pseudo-spectral approximation, in which $\lambda_{\mathbf{k}}$ becomes:

$$\lambda_{\mathbf{k}}^2 = \left(\frac{2\pi k_x}{X_{max} - X_{min}}\right)^2 + \left(\frac{2\pi k_y}{Y_{max} - Y_{min}}\right)^2 + \left(\frac{2\pi k_z}{Z_{max} - Z_{min}}\right)^2 \tag{3.7}$$

Therefore, solving the Poisson equation is reduced to calculating $\hat{\Phi}(\mathbf{k})$ for every $\mathbf{k}$ and then transforming out of Fourier space.

Once we have the potential, calculating the acceleration is straight-forward:

$$\mathbf{a}(\mathbf{r}, t) = -\boldsymbol{\nabla}\Phi(\mathbf{r}, t)| \tag{3.8}$$

Which, in the context of the lattice can be calculated simply with a central difference numerical derivative. Now, in order to update the phase space, we must first define the time interval to simulate: we name N_t the number of time intervals to simulate and $\mathrm{d}t$ the length of each of such intervals. After calculating the acceleration and defining $\mathrm{d}t$, we can update our phase space, the subtlety here is that we will only use integer arithmetic, which means that we do not exactly care for the change in velocity during a time $\mathrm{d}t$ but for how many cells in the phase space that change represents. This is modeled by:

$$\mathbf{v}_{n+1} = \mathbf{v}_n + \lfloor \mathbf{a}_n \mathrm{d}t \rceil \tag{3.9}$$

With $\lfloor x \rceil$ representing the operator "to nearest integer", so that $\mathbf{v}$ and $\lfloor \mathbf{a}\mathrm{d}t \rceil$ are vectors of integers and n represents the time instant. The update of the velocity is known as "kick". Analogously, the update of the position is known as "drift", and is given by:

$$\mathbf{r}_{n+1} = \mathbf{r}_n + \lfloor \mathbf{v}_n \mathrm{d}t \rceil \tag{3.10}$$

Using only integer arithmetics allows for the elimination of the rounding error but introduces lattice noise. Regardless, this method creates a one to one map with the continous solution.[1] [2]

The "kick" and "drift" together are known as the "Streaming" step, and it represents the classical movement of particles under a potential but without considering the collision of particles. If we want a collisionless simulation, we can just calculate again the density and continue following the same steps. If we want a collisional simulation, we must define a collisional step.

3.2 The Collisional Step

3.3 Units and Systems to Simulate

Chapter 4

Results

4.1 No Collisional

4.2 Collisional with reported $< \sigma v >$

4.3 Different Equlibrium Distributions

Chapter 5

Conclusions

5.1 A Numerically Stable Simulation

Bibliography

[1] Sebastián Franco Ulloa. Simulaciones de un fluido débilmente auto-interactuante con métodos de lattice-boltzmann, 5 2017.

[2] Philip Mocz and Sauro Succi. Integer lattice dynamics for Vlasov–Poisson. *Mon. Not. Roy. Astron. Soc.*, 465(3):3154–3162, 2017.