

# Fully Self Consistent Simulations of the 1D Fuzzy Dark Matter and Particle System

Boris Zupancic

July 17, 2022

# Contents

<b>1</b>	<b>The System</b>	<b>3</b>
1.1	Fuzzy Dark Matter (FDM) . . . . .	3
1.2	Particles . . . . .	4
1.3	Together . . . . .	5
<b>2</b>	<b>1D Numerical Schemes</b>	<b>6</b>
2.1	Initial Setup . . . . .	6
2.1.1	Initial Conditions . . . . .	6
2.2	Calculating Density . . . . .	7
2.2.1	Poisson's Equation for Particles . . . . .	8
2.3	Calculating Potential . . . . .	9
2.4	Phase Space Distributions . . . . .	9
2.4.1	Choosing the resolution $\eta$ . . . . .	10
2.5	Time Evolution . . . . .	10
2.5.1	FDM . . . . .	10
2.5.2	Particles . . . . .	10
<b>3</b>	<b>Simulations</b>	<b>12</b>
3.1	Inputs . . . . .	13
3.1.1	Optional Inputs . . . . .	14
3.2	Outputs . . . . .	14
<b>4</b>	<b>Analysis + Diagnostics</b>	<b>15</b>
4.1	Running purely N-Body . . . . .	15
4.1.1	Energies . . . . .	15
4.1.2	Checking the Virial Theorem . . . . .	16
4.1.3	Centroid Drifting . . . . .	17
4.1.4	Virialized Density Profile . . . . .	18
4.1.5	Virialized Velocity Profile . . . . .	19
4.1.6	Convergence of $z_{\text{rms}}$ and $v_{\text{rms}}$ . . . . .	19
4.2	Running purely FDM . . . . .	19
4.3	Running Together . . . . .	19

<i>CONTENTS</i>	2
<b>A Gravity in 1D</b>	<b>21</b>
A.1 Spitzer Distribution . . . . .	21
A.2 1D Virial Theorem . . . . .	21
<b>B Kick-Drift-Kick is Symplectic</b>	<b>22</b>

# Chapter 1

## The System

The system is comprised of a Fuzzy Dark Matter (FDM) component, as well as a set of particles, both confined to a 1D box. The FDM and particles share the same gravitational potential, defined by their combined density  $\rho$  and Poisson's equation:

$$\nabla^2 \Phi = 4\pi G \rho \quad (1.1)$$

The equations of motion (EOM) for FDM and the particles both rely on  $\Phi$ , but are different, as seen below. Our goal is to numerically integrate these equations of motion in a fully self-consistent manner.

### 1.1 Fuzzy Dark Matter (FDM)

The EOM for FDM in a periodic box is the Schrodinger-Poisson (SP) system :

$$\begin{cases} i\hbar \frac{\partial \psi}{\partial t} = -\frac{\hbar^2}{2m_{\text{FDM}}} \frac{\partial^2 \psi}{\partial x^2} + m_{\text{FDM}} \Phi \psi \\ \frac{\partial^2 \Phi}{\partial x^2} = 4\pi G \rho_{\text{FDM}} = 4\pi G m_{\text{FDM}} (|\psi|^2 - \langle |\psi|^2 \rangle) \end{cases} \quad (1.2)$$

where  $\hbar$  is the reduced Planck's constant,  $m_{\text{FDM}}$  is the FDM-particle mass,  $\Phi$  is the specific gravitational potential, and  $\psi$  is a Schrodinger wave-function. The wave-function  $\psi$  specifies the density:

$$\rho_{\text{FDM}} = m_{\text{FDM}} |\psi|^2$$

Thus  $|\psi|^2$  is interpreted as a number density. Further notice that in Eq(1.2) the average number density is subtracted off in Poisson's equation. This is due to Gauss' Law requiring that the boundary conditions are null for a periodic box (this "Jean's Swindle" also allows us to solve Eq(1.1) using an FFT method, outlined further down).

To non-dimensionalize our system Eq(1.2), we would like length, velocity,

time and mass scales,  $L_s, v_s, T_s, M_s$ , such that

$$\begin{cases} \Phi = v_s^2 \varphi \\ \psi = L_s^{-3/2} \chi \\ x = L_s z \\ t = T_s \tau \\ m_{\text{FDM}} = M_s \mu \end{cases} \quad (1.3)$$

To illustrate, let us fix length and velocity scales  $L_s$  and  $v_s$ , respectively. For example, we may take:

$$\begin{aligned} L_s &= 1 \text{ kpc} \\ v_s &= 100 \text{ km s}^{-1} \end{aligned}$$

and then define a time scale

$$T_s = \frac{L_s}{v_s} \quad (1.4)$$

We may define a mass scale by first setting Newton's constant to unity  $G = 1$ , then:

$$[G] = M_s^{-1} L_s^3 T_s^{-2} \Rightarrow M_s \equiv \frac{L_s^3}{T_s^2} = v_s^2 L_s \quad (1.5)$$

We may further define a "fuzziness" parameter

$$r \equiv \frac{\hbar}{2m_{\text{FDM}} v_s L_s} \quad (1.6)$$

and finally substituting (1.3) along with  $r$  into Eq(1.2), our dimensionless SP system is:

$$\begin{cases} i \frac{\partial \chi}{\partial \tau} = -r \frac{\partial^2 \chi}{\partial z^2} + \frac{1}{2r} \varphi \chi \\ \frac{\partial^2 \varphi}{\partial z^2} = 4\pi\mu(\chi\chi^* - \langle\chi\chi^*\rangle) \end{cases} \quad (1.7)$$

## 1.2 Particles

In the particle scenario, the equation of motion is the familiar form of Newton's second law, where for a single particle with position  $x$  and mass  $m$  in a gravitational potential  $U = m\Phi$ :

$$\frac{d^2 x}{dt^2} = -\nabla \Phi = -\frac{\partial}{\partial x} \Phi(x, t) \quad (1.8)$$

Substituting the same dimensionless parameters (1.3) as in the wave scenario into Eq(1.8):

$$\frac{d^2 z}{d\tau^2} = -\frac{\partial \varphi}{\partial z} \quad (1.9)$$

The gravitational potential is again given by Poisson's equation, and the particle density is

$$\rho_{\text{part}} = n \cdot m_{\text{part}} \quad (1.10)$$

where  $n$  is the volumetric/surface/linear number density and  $m_{\text{part}}$  is the particle mass. We specify how to non-dimensionlize the particle density further down below in §2.2.1, when we address the numerical schemes.

### 1.3 Together

Combining the above information, the dimensionless FDM + Particle system looks like:

$$\begin{cases} i \frac{\partial \chi}{\partial \tau} = -r \frac{\partial^2 \chi}{\partial z^2} + \frac{1}{2r} \varphi \chi \\ \frac{d^2 z}{d\tau^2} = -\frac{\partial \varphi}{\partial z} \\ \frac{\partial^2 \varphi}{\partial z^2} = 4\pi [\varrho - \langle \varrho \rangle] \end{cases} \quad (1.11)$$

where we take  $\varrho$  as specifying the total dimensionless density, given by the addition of FDM density and Particle density:

$$\varrho \equiv \varrho_{\text{FDM}} + \varrho_{\text{part}} \quad (1.12)$$

where  $\varrho_{\text{FDM}} = m_{\text{FDM}} |\chi|^2$ , and  $\varrho_{\text{part}}$  is not defined until later, in §2.2.1.

## Chapter 2

# 1D Numerical Schemes

### 2.1 Initial Setup

The system is setup on a grid of  $N$  linearly spaced points:

$$z = [z_0, z_1, z_2, \dots, z_{N-1}] \quad (2.1)$$

(indexing starts with  $i = 0$  in Python3). These values will serve as edges for histogramming positions of particles, and positions at which the density, potential and acceleration fields are defined. The width of the bins is the constant

$$dz = z_{i+1} - z_i$$

The choice of  $N$  will determine the accuracy of the FFT Poisson Solver described below; a higher value  $N$  will both increase the accuracy, and increase the computational cost.

In any given simulation, we further define the box width  $L$ , and apply the boundaries:

$$z_0 = -L/2 \quad (2.2)$$

$$z_{N-1} = L/2 \quad (2.3)$$

#### 2.1.1 Initial Conditions

Our goal is to now setup an initial distribution of FDM and/or particles.

To create these initial conditions, a collection of particles is generated from a random gaussian distribution centered at

$$z_{\text{center}} = \frac{z_0 - z_{N-1}}{2}$$

The standard deviation of this distribution is up to choice (treated as an input parameter in the simulation outlined in Chapter 3). Each particle is given an

initial velocity of zero:  $v = 0$ . Densities of each time step are later calculated, as outlined below.

In the case of FDM, the initial distribution is generated in a similar way, but as a true, smooth gaussian:

$$\rho(z, t = 0) \equiv \frac{M}{\sqrt{2\pi\varsigma_{\text{FDM}}^2}} \exp\left(-\frac{z^2}{2\varsigma_{\text{FDM}}^2}\right) \quad (2.4)$$

where we have taken  $M$  to be the total dimensional mass of the system, and  $\varsigma_{\text{FDM}}$  to be the dimensionless standard deviation of the distribution. This ensures that the density profile is normalized such that:

$$\int_{-L/2}^{L/2} \rho dz = M$$

From here we can define the initial dimensional wavefunction as the square root of the number density:

$$\psi = \sqrt{\frac{M/m_{\text{FDM}}}{\sqrt{2\pi\varsigma_{\text{FDM}}^2}} \exp\left(-\frac{z^2}{2\varsigma_{\text{FDM}}^2}\right)} \quad (2.5)$$

this conserves the equation  $\rho_{\text{FDM}} = m_{\text{FDM}}|\psi|^2$ , and following 1.3 gives the dimensionless initial wavefunction:

$$\chi(z, \tau = 0) = L^{3/2} \sqrt{\frac{M/m_{\text{FDM}}}{\sqrt{2\pi\varsigma_{\text{FDM}}^2}} \exp\left(-\frac{z^2}{2\varsigma_{\text{FDM}}^2}\right)} \quad (2.6)$$

## 2.2 Calculating Density

At any given time step of a simulation, it is necessary to calculate the acceleration field defined on the grid. To do this, density must be calculated first. It is straightforward to calculate the FDM density:  $\rho_{\text{FDM}} = m_{\text{FDM}}|\chi|^2$ . Calculating particle density amounts to histogramming the positions of particles.

Following Equation (1.10), in our numerical scheme, the particle density in a given bin of length  $dL_{\text{box}}$  will be:

$$\rho_k = \frac{n_k}{dL_{\text{box}}} \cdot \Sigma \quad (2.7)$$

where  $n_k$  is the number of particles counted in the  $k^{\text{th}}$  bin, and  $\Sigma$  is the mass per unit area defined for each particle. Note here that  $\Sigma$  must have the specified dimensions in the 1D scenario, as each particle is really treated as an infinite 2D sheet perpendicular to our grid  $z$  (see appendix A.1).

To retrieve a non-dimensional measure for density, we must consider again our scales. The values above may be written as:

$$\begin{cases} dL_{\text{box}} = L_s dz \\ \Sigma = \frac{M_s}{L_s^2} \sigma \end{cases}$$



where  $\sigma$  is a newly defined non-dimensional analogue for the mass (per unit area) of each particle. Note that we choose the grid  $z$  in Eq(2.1) as our bin edges, thus the non-dim bin size is  $dz$ . Substitution of these values into (2.7) yields a newly defined (non-dim.) density:

$$\varrho_k \equiv \frac{L_s^3}{M_s} \rho_k = \frac{n_k}{dz} \sigma \quad (2.8)$$

This is  $\varrho_k$  is the  $\varrho_{\text{part}}$  talked about in §1.3, just defined in a single bin on the grid Eq(2.1).

In practice it is necessary to interpolate the histogrammed values  $N_k$  from *between* the grid points  $z_0, z_1, \dots, z_{N-1}$  *onto* the grid points. We do this simply by taking an arbitrary array of length  $N$ :

$$\tilde{n} = [\tilde{n}_0, \tilde{n}_1, \dots, \tilde{n}_{N-1}]$$

and then for  $i \in \{1, 2, \dots, N-2\}$ , we let:

$$\tilde{n}_i = \frac{1}{2}(n_{i-1} + n_{i+1})$$

and for  $i = 0, N-1$ :

$$\begin{aligned} \tilde{n}_0 &= \frac{1}{2}(n_{N-1} + n_1) \\ \tilde{n}_{N-1} &= \frac{1}{2}(n_{N-2} + n_0) \end{aligned}$$

preserving the periodicity of the 1D box. We then re-assign  $\forall k \in \{0, \dots, N-1\}$ :

$$n_k \leftarrow \tilde{n}_k$$

### 2.2.1 Poisson's Equation for Particles

Now that we have covered how dimensionless particle density is calculated, we can non-dimensionalize the corresponding Poisson's Equation. In the dimensional case, Poisson's equation in a periodic box always looks like:

$$\nabla_x^2 \Phi = 4\pi G(\rho - \langle \rho \rangle)$$

When we make the substitutions  $\Phi = v_s^2 \varphi$  and  $\rho_k = \frac{N_k}{dL_{box}} \cdot \Sigma$ , we get (in our 1D case):

$$\frac{v_s^2}{L_s^2} \frac{\partial^2 \varphi}{\partial z^2} = 4\pi \frac{M_s}{L_s^3} \left( \frac{n_k}{\Delta z} \cdot \sigma - \left\langle \frac{n_k}{\Delta z} \cdot \sigma \right\rangle \right)$$

Which reduces to

$$\frac{\partial^2 \varphi}{\partial z^2} = 4\pi \sigma \left( \frac{n_k}{\Delta z} - \left\langle \frac{n_k}{\Delta z} \right\rangle \right) \quad (2.9)$$

You could plug this back into Eq(1.11) to get the clearer picture of the full dimensionless FDM+Particle system.

## 2.3 Calculating Potential

The second main ingredient in evolving the system is knowing the gravitational potential  $\Phi$ , or in the dimensionless case,  $\varphi$ . We must solve for  $\varphi$  from Poisson's Equation corresponding to FDM+Particles, seen in Eq(5), and we do so using a Fast-Fourier-Transform (FFT) method. The basic idea is that applying the Discrete-Discrete Fourier Transform (since we're in a periodic box with a discrete signal/sequence) on Poisson's equation yields:

$$\frac{\partial^2}{\partial z^2} \left( \sum_{n=0}^{N-1} \hat{\varphi}(k_n) e^{ik_n z} \right) = 4\pi \left( \sum_{n=0}^{N-1} \hat{\varrho}(k_n) e^{ik_n z} - \langle \varrho \rangle \right)$$

where  $k_n \equiv 2\pi n/L$ ,  $L$  being the length of the box. Notice that  $\langle \varrho \rangle = \hat{\varrho}(k_0)$ , so that from the above equation we get:

$$\begin{cases} -k_n^2 \hat{\varphi}(k_n) = 4\pi \hat{\varrho}(k_n), \forall n \in \mathbb{N} \\ 0 \cdot \hat{\varphi}(0) = 0 \end{cases}$$

yielding Poisson's equation fully solvable on our 1D mesh by simply applying an FFT to the density and applying an IFFT to the Fourier Coefficients of the potential, defined by:

$$\begin{cases} \hat{\varphi}(k_n) = -\frac{4\pi}{k_n^2} \cdot \hat{\varrho}(k_n), \forall n \in \mathbb{N} \\ \hat{\varphi}(0) = 0 \end{cases}$$

## 2.4 Phase Space Distributions

### [Check This]

One of the visual tools we'd like to generate in our simulations is the phase space distribution of not just the particles, but the phase space representation of the FDM wavefunction  $\chi$  as well.

In our codes, each particle is an instance of a Python class called `star`, with attributes of mass, position, and velocity. Thus creating a phase space plot simply means creating a scatter plot of each particle's phase space position  $(z_{\text{star}}, v_{\text{star}})$  in  $\mathbb{R}^2$ .

For creating a phase space representation of the FDM component, we turn to a windowed-FFT (or "Short-time Fourier Transform") similar to the Husimi representation. We get the phase space density by convolving, inside an FFT, the wavefunction with a Gaussian window of width  $\eta$  centered at a position  $z$

$$f(z, k, \tau) \equiv \sum_{m=0}^{N-1} \left[ e^{imk/N} \exp \left( -\frac{(z - \tilde{z})^2}{2\eta^2} \right) \chi(z, \tau) \right] \quad (2.10)$$

where  $k = 2\pi n$  is the wavenumber in the FFT, ranging in  $n \in \{0, \dots, N-1\}$ . In NumPy, which we use in our code, this sign convention requires us to use

`numpy.ifft`. By convention, `numpy.ifft` also divides by  $N$ , but we don't mind since we re-normalize  $f$  right after so that:

$$\int \int f(z, k) dz dk = 1 \quad (2.11)$$

### 2.4.1 Choosing the resolution $\eta$

Investigate

## 2.5 Time Evolution

The final ingredient in running a fully self-consistent simulation is evolving both the FDM and Particles with the same time-differencing scheme. We use a kick-drift-kick algorithm. Each step in the kick-drift-kick is performed at the same time for FDM and Particles (Kick FDM+Particles, then Drift FDM+Particles, etc). This time-integration scheme is also symplectic (see appendix B).

### 2.5.1 FDM

For the FDM, we employ a kick-drift-kick algorithm following [2], [3], to solve Eq(1.7):

$$\chi(\tau + \Delta\tau, z) = \left[ \mathcal{T} \circ e^{-i \int_{\tau}^{\tau+\Delta\tau} \left( -r\partial_z^2 + \frac{1}{2r}\varphi(\tau, z) \right)} \right] \cdot \chi(\tau, z) \quad (2.12)$$

$$\approx \left[ e^{-i \frac{\Delta\tau}{4r} \varphi(\tau+\Delta\tau, z)} \circ e^{ir\Delta\tau\partial_z^2} \circ e^{-i \frac{\Delta\tau}{4r} \varphi(\tau, z)} \right] \cdot \chi(\tau, z) \quad (2.13)$$

$$= \left[ e^{-i \frac{\Delta\tau}{4r} \varphi(\tau+\Delta\tau, z)} \circ \text{FFT}^{-1} \circ e^{ir\Delta\tau k^2} \circ \text{FFT} \circ e^{-i \frac{\Delta\tau}{4r} \varphi(\tau, z)} \right] \cdot \chi(\tau, z) \quad (2.14)$$

The "kick-drift-kick" steps are identified as:

$$\text{KICK 1: } \left[ e^{-i \frac{\Delta\tau}{4r} \varphi(\tau, z)} \right] \cdot \chi(\tau, z) \rightarrow \chi'(\tau, z)$$

$$\text{DRIFT: } \left[ \text{FFT}^{-1} \circ e^{ir\Delta\tau k^2} \circ \text{FFT} \right] \chi'(\tau, z) \rightarrow \chi''(\tau, z)$$

Update Potential

$$\text{KICK 2: } \left[ e^{-i \frac{\Delta\tau}{4r} \varphi(\tau+\Delta\tau, z)} \right] \cdot \chi''(\tau, z) \rightarrow \chi(\tau + \Delta\tau, z)$$

The "Update Potential" here (and below) means to recalculate the densities  $\varrho_{\text{FDM}}$  and  $\varrho_{\text{part}}$ , and then solve Poisson's equation before continuing onto KICK 2.

### 2.5.2 Particles

To evolve the particle system, we again employ a "kick-drift-kick" algorithm based off of that used for the FDM. We solve the particle EOM from Eqs(1.11),

for an individual particle:

$$\text{KICK 1: } v(\tau + \Delta\tau/2) = v(\tau) + \frac{\Delta\tau}{2}a(\tau)$$

$$\text{DRIFT: } z(\tau + \Delta\tau) = z(\tau) + \Delta\tau v(\tau + \Delta\tau/2)$$

$$\text{Update Potential (to get } a(\tau + \Delta\tau/2) = -\partial_z\varphi(\tau + \Delta\tau/2))$$

$$\text{KICK 2: } v(\tau + \Delta\tau) = v(\tau + \Delta\tau/2) + \frac{\Delta\tau}{2}a(\tau + \Delta\tau/2)$$

## Chapter 3

# Simulations

So far, the numerical schemes outlined above have been coded in Python3, and a repo is available on GitHub <sup>1</sup>.

In the program, the dimensional scales, as well as  $G, \hbar$  and the (non-dim.) total Mass  $M/M_s$ , are all set to unity in arbitrary units (of correct dimension):

$$\begin{cases} L_s = 1 \\ v_s = 1 \end{cases} \Rightarrow \begin{cases} T_s = 1 \\ M_s = 1 \end{cases}$$

$$\begin{cases} G = 1 \\ \hbar = 1 \\ M/M_s = 1 \end{cases}$$

Though not previously defined, the total mass of the FDM + Particle system is

$$M \equiv M_{\text{FDM}} + M_{\text{part}} \quad (3.1)$$

where

$$M_{\text{FDM}} \equiv \int_{-L/2}^{L/2} \rho_{\text{FDM}} dx \quad \text{and} \quad M_{\text{part}} \equiv \int_{-L/2}^{L/2} \rho_{\text{part}} dx \quad (3.2)$$

We further define a new expression for the fuzziness parameter  $r$  by rearranging:

$$r \equiv \frac{\hbar}{2m_{\text{FDM}}v_sL_s} \quad (3.3)$$

$$= \frac{1}{4\pi} \left( \frac{\hbar}{m_{\text{FDM}}\sigma_{\text{FDM}}R_{\text{syst}}} \right) \left( \frac{\sigma_{\text{FDM}}}{v_s} \right) \left( \frac{R_{\text{syst}}}{L_s} \right) \quad (3.4)$$

$$= \frac{1}{4\pi} \left( \frac{\sigma_{\text{FDM}}}{v_s} \right) \left( \frac{R_{\text{syst}}}{L_s} \right) \left( \frac{\lambda_{\text{dB}}}{R_{\text{syst}}} \right) \quad (3.5)$$

---

<sup>1</sup>BorisZupancic/FDM.n.Bodies on GitHub.

where  $\sigma_{\text{FDM}}$  is the velocity dispersion of the system,  $R_{\text{syst}}$  is the characteristic size of the system, and  $\lambda_{\text{deB}}$  is the deBroglie wavelength of the FDM "particle". These can be treated as inputs in the simulations.

### 3.1 Inputs

Simulations take in as input:

1. Box Length:  $L$
2. Percentage of total mass occupied by FDM: % FDM.
3. The ratio of velocity dispersion to velocity scale:  $\frac{\sigma_{\text{FDM}}}{v_s}$
4. The ratio of characteristic size to length scale:  $\frac{R_{\text{syst}}}{L_s}$
5. The ratio of deBroglie wavelength to characteristic size:  $\frac{\lambda_{\text{deB}}}{R_{\text{syst}}}$
6. The number of particles (not FDM): Num\_Particles
7. The standard deviation of the initial distributions for FDM and Particles

From these inputs, all other quantities characteristic to the FDM and particle system(s) can be determined, and we can run the simulation. From input 2, we know the percentage of mass occupied by particles:

$$\% \text{ Particles} = \%100 - \% \text{ FDM}$$

These two percentages give us the mass occupied by each of FDM and particles.

The inputs 3-5 tell us the fuzziness  $r$  of the FDM, which gives us the FDM boson mass:  $m_{\text{FDM}}$ . Combining that with the total FDM mass, we can determine the normalization of  $\psi$  (or otherwise  $\chi$ ):

$$\int_{-L/2}^{L/2} \psi(x) dx = \frac{M_{\text{FDM}}}{m_{\text{FDM}}} \equiv \text{Num\_Bosons}$$

which can be interpreted as the number of FDM bosons present.

In a similar but opposite way, we can determine the particle mass from the total number of particles (input 5):

$$m_{\text{part}} = \frac{M_{\text{part}}}{\text{Num\_Particles}}$$

Other parameters like the grid resolution  $N$ , are set at fixed values in the code.

### 3.1.1 Optional Inputs

There are other possible/optional inputs in the code, such as:

- Fixing the potential to a quadratic.
- Whether to output plots at every time-step or at snapshots.
- Diagnostic/Analysis functionalities:
  - Tracking the centroid of the particle distribution.
  - Tracking the Energies of a few individual particles
  - Tracking the total Kinetic and Potential Energies of the system.

## 3.2 Outputs

The standard output of the code is the final wavefunction  $\chi$  (collected in a csv) and the collection of particles (or `star` class instances), saved in .csv files, as well as plots of the FDM distribution, Particle Distribution, their respective phase-space distributions, and a plot of force contributions from FDM and Particles.

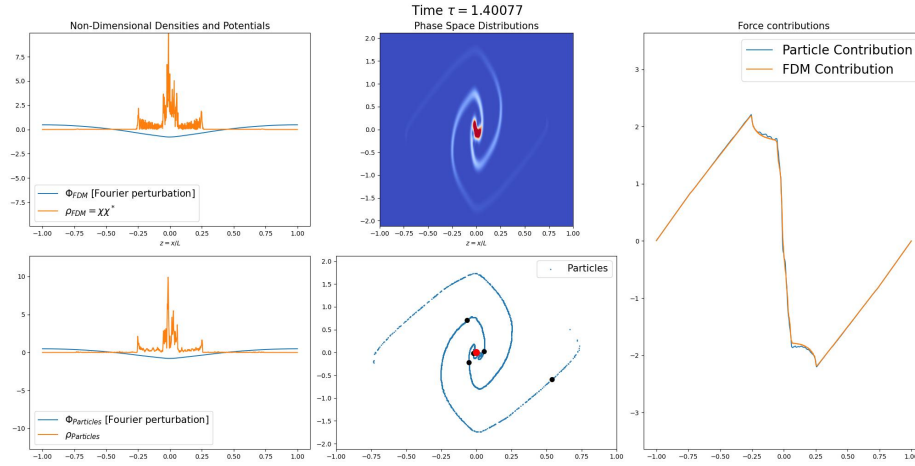


Figure 3.1: An example plot from a simulation of a 50% FDM, 50% particle system. (Can't remember how many particles, or the specific  $r$ , or any other input parameters).

The optional diagnostics above are also outputted as arrays and saved in .csv files.

## Chapter 4

# Analysis + Diagnostics

### 4.1 Running purely N-Body

Diagnostics/tests were performed for an array of particle numbers, between 5 and 100000 particles, all generated from a Gaussian distribution of standard deviation 0.1, in a system of total mass  $M = 1$  (just as described in Chapter 2.)

To illustrate the results of the analysis and diagnostics, we use the 100000-Body run as the primary example below. We also include an analysis of the convergent behaviour of the system upon increasing the particle number.

#### 4.1.1 Energies

To check that conservation of Energy holds, and to analyse possible oddities in the energy of individual particles over time, we perform several diagnostics tracking Kinetic and Potential Energies.

Kinetic energy for a single particle is calculated simply as:

$$K = \frac{1}{2}m_{\text{part}}v^2 \quad (4.1)$$

Potential energy for a single particle is calculated by locating the particle's position on the grid by index, then interpolating between grid points to determine what the potential  $\Phi$  is at the exact location of the particle, then calculate:

$$W = m_{\text{part}}\Phi(x) \quad (4.2)$$

Now, since the FFT solving method for Poisson's equation, seen in §2.3, actually returns

$$\Phi \leftarrow \Phi - \langle \Phi \rangle$$

instead of  $\Phi$ , we attempt to properly zero the potential by subtracting the maximum value in the box:

$$\Phi \leftarrow \Phi - \sup_{[-L/2, L/2]} (\Phi)$$



and then use this value in the potential energy Eq(4.2).

We also compute the total potential energy in a given particle system as

$$W_{\text{tot}} = \frac{1}{2} \sum_{i=1}^{\text{Num\_Particles}} m_{\text{part}} \Phi(x_i) \quad (4.3)$$

(this has been giving me a little trouble, on whether or not to include the factor of 1/2.) We of course complete these calculations in our analogous non-dimensional values.

Using these definitions of  $K$  and  $W$ , we track the energies of 5 individual particles, at every time-step. We also track the energies of *all* particles, but only at snapshots in time, defined at indices:

$$i = 99 * [0, 1, 2, 4, 8, 16, 32, 64]$$

In Figure 4.1 are shown the relevant plots for the 5 individual particles, showing Kinetic, Potential and Total Energies, and in Figure 4.2 are the totals of these energies, along with “Virial Ratio”:

$$\left| \frac{K}{W} \right| \quad (4.4)$$

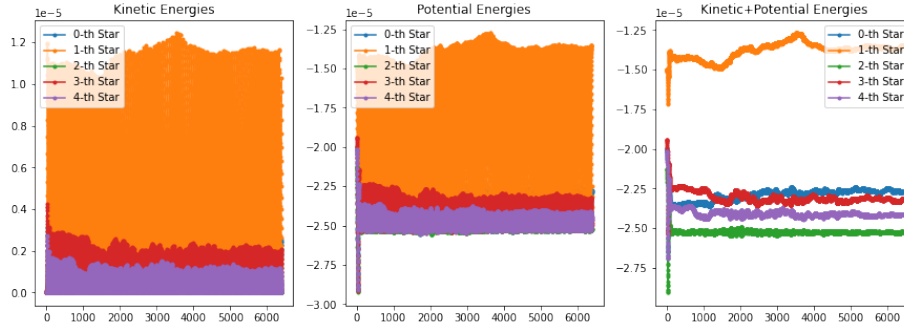


Figure 4.1: Caption

Now in Figure 4.3 we see the plot of the same values seen above in 4.2, but for every single particle, taken at the snapshot times defined above.

#### 4.1.2 Checking the Virial Theorem

As the simulation goes on, we expect the system to eventually virialize, especially with a large number of particles such as 100000. Visually, this may seem obvious simply by checking that the phase space distribution no longer has any

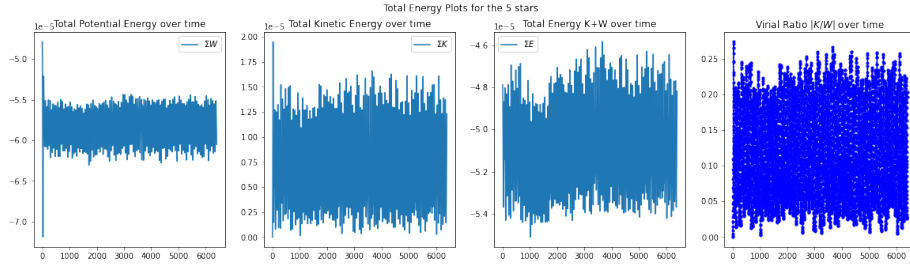


Figure 4.2: Caption

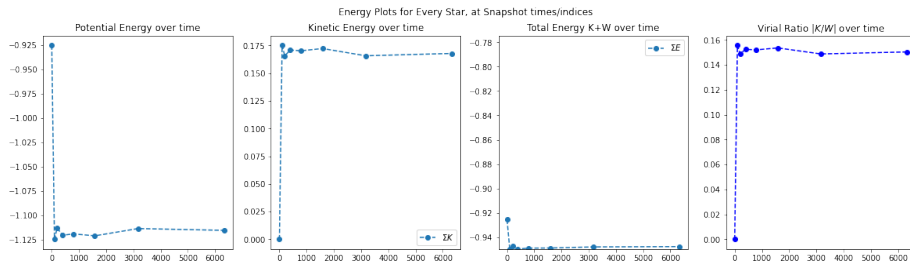


Figure 4.3: Caption

spirals/phase-wrapping, but the Virial theorem specifically states [1]:

$$\text{Tensor:} \quad \frac{1}{2} \frac{d^2 I_{jk}}{dt^2} = 2K_{jk} + W_{jk} \quad (4.5)$$

$$\text{Scalar:} \quad 2K + W = 0 \quad (4.6)$$

As it turns out, in 1D this changes to (see appendix A.1, appendix A.2):

$$K + W = 0 \quad (4.7)$$

and this is what we would like to check.

Note that, while we do wish to check Eq(4.7), the condition becomes again slightly modified since we have zeroed our gravitational potential energy in an odd way (Eq(4.2)). **Going to have to check this; since we are in a periodic box, the boundary conditions may not be such that  $\Phi \rightarrow 0$  as  $x \rightarrow \infty$ . This means that Eq(4.7) can actually take the form  $K + W = C$ , where  $C$  is some unknown constant, not to be confused with the constant total Energy.**

### 4.1.3 Centroid Drifting

An important quantity that should be conserved is the centre of mass of the distribution, so we track this. **Currently, it appears to drift over time, and I**

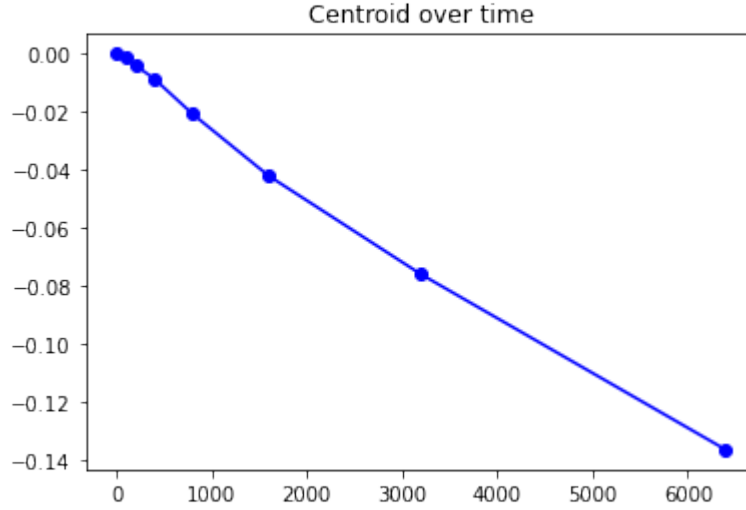


Figure 4.4: Caption

**don't know why.** There is also seemingly no pattern in drift direction, or shape, between different instances of the same simulations, and between simulations with different numbers of particles.

#### 4.1.4 Virialized Density Profile

For the amount of time we run the simulation, we expect the system to virialize by the end. We want to see the general form of the density profile in the virialized system.

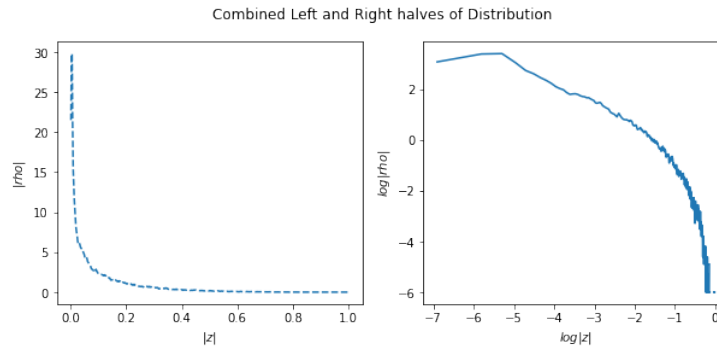


Figure 4.5: Caption

We manage to curve fit the density profile, with 3 different functions, seen

in Figure 4.6. Most important is the fitting function:

$$f(z; a_0, a_1, a_2) = \frac{a_0}{|z|^{a_2}(|z| + a_1)^2} \quad (4.8)$$

where  $a_0, a_1, a_2$  are fitting parameters, which resembles a modified NFW profile [4].

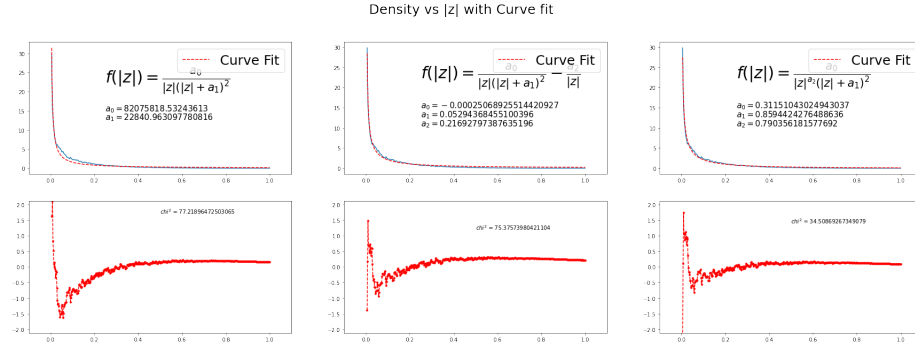


Figure 4.6: Caption

#### 4.1.5 Virialized Velocity Profile

Just the same as the density profile approaches a particular distribution, we expect the velocity profile to also approach a particular distribution. We plot this on the right of Figure 4.7, displaying the root-mean-square velocity  $v_{\text{rms}}$  as a function of radial position  $|z|$ . The values are computed by sorting/histogramming the particles by position, the computing the  $v_{\text{rms}}$  in these bins. A horizontal line through the root-mean-square velocity of *all* particles would represent an isothermal distribution as seen in the Spitzer distribution (see problem 4.21 in Binney & Tremaine [1] and appendix A.1).

#### 4.1.6 Convergence of $z_{\text{rms}}$ and $v_{\text{rms}}$

We calculate the root-mean-square position  $z_{\text{rms}}$  and velocity  $v_{\text{rms}}$  at the end of each trial simulation, when the system has virialized. They should converge as the number of particles increases and the system becomes more “smooth”, since it approaches a state further resembling that described by the virial equation. This can be seen in Figure 4.8.

### 4.2 Running purely FDM

### 4.3 Running Together

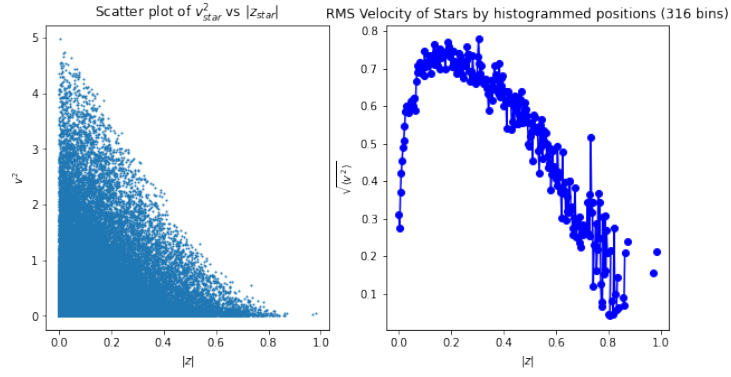


Figure 4.7: Caption

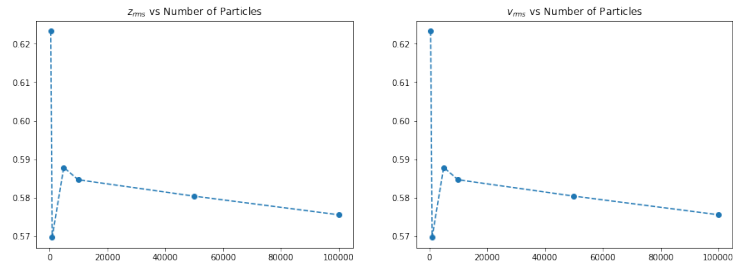


Figure 4.8: Caption

## Appendix A

# Gravity in 1D

### A.1 Spitzer Distribution

### A.2 1D Virial Theorem

## Appendix B

# Kick-Drift-Kick is Symplectic

# Bibliography

- [1] James Binney and Scott Tremaine. *Galactic Dynamics: Second Edition*. 2008.
- [2] Faber Edwards, Emily Kendall, Shaun Hotchkiss, and Richard Easther. PyUltraLight: a pseudo-spectral solver for ultralight dark matter dynamics. *Journal of Cosmology and Astroparticle Physics*, 2018(10):027–027, oct 2018.
- [3] Simon May and Volker Springel. Structure formation in large-volume cosmological simulations of fuzzy dark matter: impact of the non-linear dynamics. *Monthly Notices of the Royal Astronomical Society*, 506(2):2603–2618, jun 2021.
- [4] Julio F. Navarro, Carlos S. Frenk, and Simon D. M. White. The Structure of Cold Dark Matter Halos. 462:563, May 1996.