Fast particle-mesh code for Milgromian dynamics

Visser, P.M.; Eijt, S.W.H.; de Nijs, J.V.

DOI
10.1051/0004-6361/202347830

Publication date
2024

Document Version
Final published version

Published in
Astronomy & Astrophysics

A&A, 681, A90 (2024)
https://doi.org/10.1051/0004-6361/202347830

Astronomy & Astrophysics

Fast particle-mesh code for Milgromian dynamics

P. M. Visser[1], S. W. H. Eijt[2], and J. V. de Nijs[1,2]

[1] Delft Institute of Applied Mathematics, Delft University of Technology, Mekelweg 4, 2628 CD Delft, The Netherlands
e-mail: p.m.visser@tudelft.nl
[2] Department of Radiation Science and Technology, Applied Sciences, Delft University of Technology, Mekelweg 15, 2629 JB Delft, The Netherlands

Received 30 August 2023 / Accepted 3 October 2023

ABSTRACT

Context. Modified Newtonian dynamics (MOND) is a promising alternative to dark matter. To further test the theory, there is a need for fluid- and particle-dynamics simulations. The force in MOND is not a direct particle-particle interaction, but derives from a potential for which a nonlinear partial differential equation (PDE) needs to be solved. Normally, this makes the problem of simulating dynamical evolution computationally expensive.
Aims. We intend to develop a fast particle-mesh (PM) code for MOND (the AQUAL formalism).
Methods. We transformed the nonlinear equation for MOND into a system of linear PDEs plus one algebraic equation. An iterative scheme with the fast Fourier transform (FFT) produces successively better numerical approximations.
Results. The algorithm was tested for dynamical systems in MOND where analytical solutions are known: the two-body problem, a body with a circular ring, and a spherical distribution of particles in thermal equilibrium in the self-consistent potential.
Conclusions. The PM code can accurately calculate the forces at subpixel scale and reproduces the analytical solutions. Four iterations are required for the potential, but when the spatial steps are small compared to the kernel width, one iteration is suffices. The use of a smoothing kernel for the accelerations is inevitable in order to eliminate the self-gravity of the point particles. Our PDE solver is 15 to 42 times as slow as a standard Poisson solver. However, the smoothing and particle propagation takes up most of the time above one particle per 10^3 pixels. The FFTs, the smoothing, and the propagation part in the code can all be parallelized.

Key words. gravitation – methods: numerical – planets and satellites: dynamical evolution and stability – galaxies: kinematics and dynamics – galaxies: clusters: general – dark matter

1. Introduction

If stars and galaxies move under Newtonian gravity, dark matter is required to bind the stars to galaxies and galaxies in clusters. There is ample evidence. The mass inferred from the stellar velocities in clusters (with the virial equation; Zwicky 1933, 1937) and in galaxies (with Huygens' equation; Rubin & Ford 1970; Rubin et al. 1980) is much higher than the mass of the luminous matter. ΛCDM cosmology requires six times as much matter as can be directly detected, hence, a large nonradiative matter component must dominate the Universe.

Despite a search of over half a century, no dark matter particle has been directly detected, while the list of astronomical observations that are difficult to explain with dark matter models has been growing: (i) The absence of cusps at galactic cores (McGaugh et al. 2003; de Blok 2010); (ii) bar structures at galactic cores (Chiba & Schönrich 2021; Roshan et al. 2021a,b); (iii) the low number of galaxy satellites (Bullock 2013); (iv) the absence of dynamical friction on galaxy satellites by its dark matter halo (Angus et al. 2011; Kroupa 2015; Oehm et al. 2017); (v) the shape and central brightness of the Fornax cluster (Asencio et al. 2022); (vi) the symmetry breaking of tidal tails of open star clusters (Kroupa et al. 2022); (vii) galaxies without dark matter (Moreno et al. 2022; Comerón et al. 2023); (viii) the intracluster gas does not cool (Fabian 1994); (ix) strong local lensing in galaxy clusters (Meneghetti et al. 2020); (x) the early formation of very large galaxies and clusters (Asencio et al. 2020, 2023; Labbé et al. 2023); (xi) the Hubble H_0 tension and the S_8 tension occur in standard ΛCDM cosmology (Planck Collaboration VI 2020; Haslbauer et al. 2020; Asgari et al. 2021; Riess et al. 2021); (xii) the violation of the strong equivalence principle (Blanchet & Novak 2011; Chae et al. 2020); (xiii) and the velocities in wide binary star systems (Banik & Zhao 2018; Pittordis & Sutherland 2019; Chae 2023; Hernandez 2023).

An alternative to dark matter is modified Newtonian dynamics (MOND), which was invented by Milgrom and Bekenstein (Milgrom 1983; Bekenstein & Milgrom 1984; Milgrom & Sanders 2008). Whereas Einstein's general relativity modifies Newton when the gravity potential ϕ is deep, Milgrom's MOND instead modifies Newton at low accelerations. The critical acceleration scale is close to the scale set by the cosmological constant Λ. It is $a_0 \approx .127 \cdot c^2\Lambda^{1/2}$. The strengths of the mutual gravity between stars, the centripetal acceleration of stars in the galaxy outskirts, and the accelerations in galaxy clusters all have values of about a_0. Because MOND is nonrelativistic, it applies in the regime in which $|\phi| \ll c^2$ and $|\nabla\phi| \lesssim c^2\Lambda^{1/2}$. The theory explains some of the listed problems and it gives a natural explanation for the observed flat rotation curves of galaxies as well as for the Tully-Fisher relation (McGaugh et al. 2000).

Although the MOND model makes definite predictions without the need for fit parameters, the model is complicated by the fact that it is not relativistic. It needs a preferred inertial reference frame (i.e., it is an aether theory) and the gravity from distant

sources cause an external field effect (violating the strong equivalence principle). These effects need to be considered when the possibility of MOND is to be ruled out or confirmed.

General relativistic models that reduce to MOND at low energies introduce additional degrees of freedom. The new TeVeS theories of Skordis & Złośnik (2019, 2021) have resolved the long-standing problem that gravitational waves traveled faster than photons. This problem plagued older MOND-extensions (Bekenstein 2004).

Milgrom's MOND modifies the equation for the gravitational potential using a nonlinear PDE. The numerical codes that have been developed to simulate the dynamical evolution in MOND are found in Brada & Milgrom (1999); Nipoti et al. (2007); Llinares et al. (2008), N-MODY by Londrillo & Nipoti (2009), and RayMOND by Candlish et al. (2014), based on multigrid discretization. The work of Angus & Diaferio (2011); Peng et al. (2016) solved for QuMOND, the quasilinear MOND model (Milgrom 2010). The code called Phantom of Ramses (Lüghausen et al. 2015; Nagesh et al. 2022) also solves QuMOND. The codes RayMOND and Phantom of Ramses are both based on RAMSES (by Teyssier 2002).

We describe an algorithm for N-body simulations in MOND. The particles can be Solar System planets, stars in a galaxy, or the galaxies in a galaxy cluster. The following main fields in astronomy are affected by Milgromian gravity:

(i) Celestial mechanics. In the Solar System, the perturbation of the planetary orbits due to a possible residual MOND effect may have to be calculated. Because outer Solar System dynamics is in the crossover regime, this correction critically depends on the interpolation function, which describes the transition between MOND and normal gravity. In particular, the perihelion precession of the outer planets, where acceleration is lowest, could be affected (Pitjeva & Pitjev 2013; Paučo & Klačka 2016; Paučo 2017; Yoon & Darriba 2020). Here, the effect of MOND may lie just below the current detection limit. Binary star systems with a wide separation are in the transition of the interpolation (Hernandez et al. 2019). For both these systems a two-body model is appropriate when the external field from the Milky Way is taken into account (Iorio 2009, 2017).

(ii) Dense stellar systems. These include open clusters and globular clusters, where close encounters between stars are relevant. The evolution requires inclusion of short-range gravitational interactions. Hence, individual particles must be tracked.

(iii) Galactic dynamics. Examples are transient phenomena such as tidal effects between neighboring galaxies, the collisions between galaxies, the formation of tails, and the collapse or merging of smaller stellar systems. Galaxy-formation simulations in QuMOND were made by Wittenburg et al. (2020); Eappen et al. (2022).

(iv) Galaxy clusters. Here, the particles are the individual galaxies. An N-body code for the dynamics with small particle numbers, between 10^2 and 10^3, possibly supplemented with hydrodynamics for the gas component, would be the appropriate model.

(v) Cosmology. Structure formation in the early Universe would be affected by MOND because the density fluctuations will initially be small, accelerations could begin with low values, and structure formation may start in the deep-MOND regime. Cosmological simulations in QuMOND were made by Katz et al. (2013); Haslbauer et al. (2020); Wittenburg et al. (2023).

2. Theory and main result: The idea of the method

As the particles move in space under the influence of the gravitational potential $\phi(\boldsymbol{r}, t)$, they experience an acceleration given by Newton's second law,

$$m_i \ddot{\boldsymbol{r}}_i(t) = -m_i \boldsymbol{\nabla}\phi(\boldsymbol{r}_i(t), t). \tag{1}$$

The baryonic mass density for the set of N point masses is

$$\rho_{\rm B}(\boldsymbol{r}, t) = \sum\nolimits_{i=1}^{N} m_i \delta(\boldsymbol{r} - \boldsymbol{r}_i(t)). \tag{2}$$

Here, δ is the three-dimensional Dirac-delta function. Whereas the potential is a solution of the Poisson equation in the Newton theory, in MOND (the AQUAL formalism), the potential is a solution to the following nonlinear partial differential equation (PDE):

$$\boldsymbol{\nabla} \bullet \mu\Big(\frac{|\boldsymbol{\nabla}\phi|}{a_0}\Big)\boldsymbol{\nabla}\phi = 4\pi G \rho_{\rm B}(\boldsymbol{r}, t), \tag{3}$$

with the boundary condition $\boldsymbol{\nabla}\phi(\boldsymbol{r}, t) \longrightarrow \boldsymbol{0}$ as $|\boldsymbol{r}| \longrightarrow \infty$. The scalar function $\mu(x)$ is called the interpolation function because it interpolates between the regime of high and low accelerations. At high acceleration, $\mu(x) \longrightarrow 1$ so that the potential will approach the Newton potential. At low acceleration, the deep MOND regime begins, and $\mu(x) \longrightarrow x$. An isolated and localized mass m has the asymptotic spherically symmetric potential $\phi(\boldsymbol{r}) = \sqrt{Gma_0}\,\log|\boldsymbol{r}|$. See Table A.1 for the symbols and notation. A popular choice for the interpolation function that is consistent with all observations so far (Brouwer et al. 2021) is

$$\mu(x) = \frac{x}{\sqrt{1 + x^2}}. \tag{4}$$

2.1. An equivalent system for AQUAL

We now demonstrate how to rewrite the Eq. (3) for the potential in MOND by transforming this equation into a system of coupled linear PDEs for four vector fields $\boldsymbol{g}_{\rm N}$, $\boldsymbol{g}_{\rm M}$, $\boldsymbol{H}$, and $\boldsymbol{F}$, combined with one algebraic equation. This system of equations is

$$\begin{cases} \boldsymbol{\nabla} \bullet \boldsymbol{g}_{\rm N} = -4\pi G\rho_{\rm B}, & \boldsymbol{\nabla} \times \boldsymbol{g}_{\rm N} = \boldsymbol{0}, & (5)\\ \boldsymbol{F} = \boldsymbol{g}_{\rm N} + \boldsymbol{H}, & & (6)\\ & \boldsymbol{\nabla} \bullet \boldsymbol{H} = 0, & (7)\\ \boldsymbol{g}_{\rm M} = \nu\Big(\frac{F}{a_0}\Big)\boldsymbol{F}, & & (8)\\ & \boldsymbol{\nabla} \times \boldsymbol{g}_{\rm M} = \boldsymbol{0}, & (9)\\ \boldsymbol{F} = \mu\Big(\frac{g_{\rm M}}{a_0}\Big)\boldsymbol{g}_{\rm M}. & & (10) \end{cases}$$

The vector fields $\boldsymbol{g}_{\rm N}$ and $\boldsymbol{g}_{\rm M}$ are the Newton and MOND acceleration fields. With the identification $\boldsymbol{g}_{\rm N} = -\boldsymbol{\nabla}\phi_{\rm N}$, the two equations in Eq. (5) are equivalent to the Poisson equation. Our new system effectively expresses the MOND gravity field $\boldsymbol{g}_{\rm M}$ in the Newtonian gravity field $\boldsymbol{g}_{\rm N}$. Equations (8) and (10) are not independent equations, but are the inverse of each other. Here, F and $g_{\rm M}$ are the lengths of the vectors $\boldsymbol{F}$ and $\boldsymbol{g}_{\rm M}$. The pair $\mu(x)$ and $\nu(y)$ relate the strength of the Newtonian gravity to the strength of the Milgromian gravity by

$$\mu(x)x = y = \frac{F}{a_0}, \quad \text{and} \quad \nu(y)y = x = \frac{g_{\rm M}}{a_0}.$$

The choice Eq. (4) for the interpolation function gives

$$\nu(y) = \sqrt{\frac{1}{2} + \frac{1}{2}\sqrt{1 + \frac{4}{y^2}}}\ .$$

Here follows the proof that the system of Eqs. (5)–(10) is equivalent to Eq. (3). Instead of an equation for the potential ϕ, we have an equation for the acceleration field $\boldsymbol{g}_{\rm M}$. We thus need to identify $\boldsymbol{g}_{\rm M} = -\boldsymbol{\nabla}\phi$. When we now combine Eq. (6) with Eq. (10), solve for $\boldsymbol{g}_{\rm N}$, then substitute this solution in the first of Eq. (5), and use Eq. (7), we obtain

$$-4\pi G\rho_{\rm B} \overset{(5)}{=} \boldsymbol{\nabla}\bullet\boldsymbol{g}_{\rm N} \overset{(6)}{=} \boldsymbol{\nabla}\bullet\boldsymbol{F} - \boldsymbol{\nabla}\bullet\boldsymbol{H} \overset{(7)}{=} \boldsymbol{\nabla}\bullet\boldsymbol{F} \overset{(10)}{=} \boldsymbol{\nabla}\bullet\mu\Big(\frac{g_{\rm M}}{a_0}\Big)\boldsymbol{g}_{\rm M}.$$

Equation (9) states that the field is curl-free. Hence, $\boldsymbol{g}_{\rm M}$ is a conservative field with a potential. When we substitute $\boldsymbol{g}_{\rm M} = -\boldsymbol{\nabla}\phi$, we find Eq. (3). This proves that the system Eqs. (5)–(10) implies Eq. (3). For the converse, we note that when we have a solution of Eq. (3), we can calculate $\boldsymbol{g}_{\rm M}$. This automatically satisfies Eq. (9). We can then define $\boldsymbol{F}$ with Eq. (10). We then let Eq. (6) be its Helmholtz decomposition. This defines the functions $\boldsymbol{g}_{\rm N}$ and $\boldsymbol{H}$, which must satisfy the second of Eqs. (5) and (7), stating that the first component is curl-free and the second component is divergence-free. This completes the proof.

In any numerical code that uses this system, the linear PDEs can be solved using standard numerical methods for linear equations. The nonlinear equation requires a special solver. We have devised the following numerical iteration scheme that approaches the solution of the system (5)–(10). We first solve Eq. (5) for the Newtonian gravity field $\boldsymbol{g}_{\rm N}$, which we store in memory. Then we take $\boldsymbol{H} = \boldsymbol{0}$ for the initial value. The iteration scheme consists of the following steps:

1. Calculate $\boldsymbol{F}$ with Eq. (6).
2. Calculate $\boldsymbol{g}_{\rm M}$ with Eq. (8).
3. Project $\boldsymbol{g}_{\rm M} \longrightarrow \boldsymbol{g}_{\rm M\|}$ onto the curl-free part, as per Eq. (9).
4. Calculate $\boldsymbol{F}$ with Eq. (10).
5. Calculate $\boldsymbol{H}$ with Eq. (6).
6. Project $\boldsymbol{H} \longrightarrow \boldsymbol{H}_\perp$ onto the divergence-free part, as per Eq. (7).
7. Repeat: Go to step 1.

The calculation in steps 3 and 6 can quickly be made in the Fourier domain (see Appendix C for the equations). The use of the fast Fourier transform (FFT) allows us to perform these steps fast.

The iterative procedure converges to a numerically stable solution. Because we start with a zero $\boldsymbol{H}$ field, its change after the iterations allows us to estimate the error in the numerical solution. It was found that the error initially drops with a factor of 10 per iteration, but after about 15 iterations, 5 iterations are required for the error to drop with a factor 10, until machine precision is reached (see Platschorre 2019, where it was used for the first time). One clue to the explanation for the fast convergence is the following. For a spherical field, we have $\boldsymbol{H} = \boldsymbol{0}$ (see Bekenstein & Milgrom 1984). Because the masses are spherical, close to a mass, the solution is approximately spherically symmetric. Far away from a collection of masses, the solution becomes spherical as well. Thus, near the masses and away from the masses, the initial value is correct. In the iterative steps, the field in the intermediate regions is changed. Here, $\boldsymbol{H}$ is not zero.

The apparent matter (baryonic plus dark matter) distribution can be calculated by substituting the correct potential (the potential that is consistent with the observed particle motions and lensing) in the incorrect Newton-Poisson equation. This is equivalent to

$$\rho_{\rm B}(\boldsymbol{r}) = \frac{-1}{4\pi G}\boldsymbol{\nabla}\bullet\boldsymbol{g}_{\rm N}, \quad \rho_{\rm A}(\boldsymbol{r}) = \frac{-1}{4\pi G}\boldsymbol{\nabla}\bullet\boldsymbol{g}_{\rm M}. \tag{11}$$

The apparent density $\rho_{\rm A}$ is also known as the effective density (see Banik & Zhao 2022). This leads to an apparent dark matter density of $\rho_{\rm A} - \rho_{\rm B}$, also called phantom dark matter density. The total baryonic mass and apparent mass in a volume W can be found from the flux through its boundary,

$$M_{\rm B}(W) = -\oiint_{\partial W}\boldsymbol{g}_{\rm N}\bullet{\rm d}\boldsymbol{S}, \quad M_{\rm A}(W) = -\oiint_{\partial W}\boldsymbol{g}_{\rm M}\bullet{\rm d}\boldsymbol{S}. \tag{12}$$

2.2. An equivalent system for QuMOND

An alternative to the AQUAL formalism of MOND is quasi-linear MOND or QuMOND (Milgrom 2010; Candlish et al. 2014; Lüghausen et al. 2015; Nagesh et al. 2022). This model can be cast as a similar system

$$\begin{cases} \boldsymbol{\nabla}\bullet\boldsymbol{g}_{\rm N} = -4\pi G\rho_{\rm B}, & \boldsymbol{\nabla}\times\boldsymbol{g}_{\rm N} = \boldsymbol{0}, \\ \boldsymbol{g}_{\rm M} + \boldsymbol{H} = \nu\Big(\frac{g_{\rm N}}{a_0}\Big)\boldsymbol{g}_{\rm N}, & \boldsymbol{\nabla}\times\boldsymbol{g}_{\rm M} = \boldsymbol{0}, \quad \boldsymbol{\nabla}\bullet\boldsymbol{H} = 0. \end{cases}$$

We do not consider QuMOND any further. However, the algorithm can easily be adapted using the same steps.

3. The algorithm: Dealing with the self-gravity

We now describe the implementation of the method. The code needs to numerically approximate the solution of the MOND PDEs in Eqs. (5)–(10) and propagate particle motion and a time-dependent mass density. The particle equations of motion Eq. (1) are implemented with Leapfrog,

$$\boldsymbol{v}_i \longrightarrow \boldsymbol{v}_i + \boldsymbol{g}_i(t)\Delta t, \quad \boldsymbol{r}_i \longrightarrow \boldsymbol{r}_i + \boldsymbol{v}_i\Delta t, \quad i = 1\ldots N. \tag{13}$$

Although the formulas do not explicitly show this, the velocities and positions represent the values at a half time step difference. In principle, an actual half time step needs to be made initially (and also finally) to obtain the true Leapfrog integration scheme.

The code uses a discrete Fourier transform, so that space is represented as a periodic cubic lattice of n^3 cells we call pixels. Hence, the scheme does not have nested grids; the grid cells are all equal and do not change. However, due to the periodic nature of the simulated space, matter inside one cubic volume is effectively influenced by the gravity of the neighboring copies. This effect is nonphysical, and a volume of empty space needs to surround the matter when this effect is to be small. Ideally, the center of mass is placed in the origin, and it is ensured that particles stay away from the boundary of the cube.

3.1. Interpolation for the particle density function

The main drawback of the particle-mesh (PM) method is the lack of accuracy at the short-distance scale of the pixels. Although each particle is located in a single pixel, we desire accuracy that is below the grid size. This influences both the field created by and the acceleration on the particle. First, the field created by particle i close to the mass should resemble a spherical field about the point $\boldsymbol{r}_i$, which is not the center of a pixel. Second, we also need to approximate the acceleration on particle i by the field at $\boldsymbol{r}_i$, for which the value at the center of the pixel is not accurate.

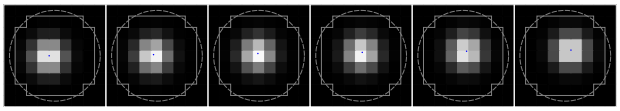

Fig. 1. Anti-aliasing needed to suppress the self-gravity. A cross section of the Gaussian density Eq. (14) with one-pixel variance about (blue) points in the middle grid cell, representing a 12th order weight function. We only account for the pixels inside a sphere with a radius of $4s$ (dashed circles). This results in a domain of 251 cells (inside the contours). The linear motion of a point particle across the middle cell (left to top right) shows that the simulated density changes smoothly between panels.

We used a Gaussian kernel to create a smoothed density in order to suppresses anti-aliasing. The density of Eq. (2) was replaced by

$$\rho_{\mathrm{B}}(\boldsymbol{r}) = \frac{1}{s^3(2\pi)^{3/2}} \sum_{i=1}^{N} m_i \exp\left(\frac{-|\boldsymbol{r}-\boldsymbol{r}_i|^2}{2s^2}\right). \tag{14}$$

Hence, each of the N point particles is represented by the Gaussian in Eq. (14) centered about the particle position. We chose a standard deviation with a width of one pixel s. The Gaussian is then effectively a cloud-in-cell (CIC) method of order 12 (see Hockney & Eastwood 1988). This is explained in Appendix B. Because Gaussians drop off fast, we implemented the calculation of the density by only addressing the pixels that are within a distance of $4s$ from the central pixel containing the particle. To do this, we stored a list with the 251 coordinates of the ball-shaped domain with a radius of $4s$. This is illustrated in Fig. 1. Moreover, we created a lookup table for the exponential function.

3.2. Interpolation for the forces on the particles

Next, we describe how we approximated the acceleration of particle i. Again, we wished to know the acceleration at a point $\boldsymbol{r}_i$ inside some pixel with subpixel accuracy. However, the acceleration $\boldsymbol{g}_{\mathrm{B}}$ is only calculated at the pixel positions. We would therefore need to interpolate between the values at different pixels. For the force on particle i, located at $\boldsymbol{r}_i$, we used the following smoothed average:

$$\boldsymbol{g}_i = \frac{-1}{s^3(2\pi)^{3/2}} \iiint \boldsymbol{\nabla}\phi(\boldsymbol{r}) \exp\left(\frac{-|\boldsymbol{r}-\boldsymbol{r}_i|^2}{2s^2}\right) \mathrm{d}x\, \mathrm{d}y\, \mathrm{d}z. \tag{15}$$

On the mesh, the gradient of the potential $\boldsymbol{\nabla}\phi$ in this expression is calculated with the finite-difference gradient, and the integral in this expression becomes a pixel sum. We only summed over pixels within a distance $4s$ from the pixel containing $\boldsymbol{r}_i$.

The smoothing methods discussed in this section will generally improve the accuracy in any PM method. However, in the case of MOND, this smoothing is essential for dealing with the self-gravity. The software is available on GitHub[1] under the MIT license and is archived in the ASCL (de Nijs et al. 2023).

4. Results: Validation using analytic solutions

The problem of self-gravity dominating the actual force on a point particle is inherent to the PM method for MOND. The effect of an imprecise subpixel resolution for the position of a particle and the consequently imprecise evaluation of the acceleration on that particle is always present, even for a single particle. We therefore also need to test the accuracy of the method for linear motion. We have found that in our simulations, a point particle of mass m experiences nonphysical accelerations with the average value of $1.08 \cdot 10^{-3}\sqrt{Gma_0}\,/s$ for a 256^3 grid (see de Nijs 2023). This small error is independent of the number of pixels.

In this section, we present results of simulations with the code for three systems: the two-body problem, a ringed system, and an isotropic three-dimensional system. Because the main interest is in MOND, we considered the deep MOND case. This means that the interpolation functions we used in our simulation are

$$\mu(x) = x, \quad \nu(y) = \frac{1}{\sqrt{y}}.$$

For this regime, Milgrom (1994, 1997, 2014) found, using the method of the virial, that the kinetic energy for a bound system with the center-of-mass motion at rest is

$$E_{\mathrm{kin}} = \frac{M\overline{\boldsymbol{v}\bullet\boldsymbol{v}}}{2} = \frac{M\overline{v^2}}{2} = \frac{\sqrt{Ga_0}}{3}\left(M^{3/2} - \sum_i {m_i}^{3/2}\right), \tag{16}$$

with m_i the individual particle masses, and M the total mass. If there is no external field, this single equation allows us to calculate the MOND forces between particles for highly symmetric cases, thereby bypassing the nonlinear PDE. We used these analytic solutions to test our PM code.

4.1. Example I: Wide binary system

We considered the two-body problem in MOND. This is a model for a binary stellar system with two comparable masses with a wide separation. The interparticle force is found from Eq. (16) to be

$$F_{12} = \frac{2\sqrt{Ga_0}}{3r}\left(M^{3/2} - m_1^{3/2} - m_2^{3/2}\right), \tag{17}$$

with $M = m_1 + m_2$, and where $r = |\boldsymbol{r}_2 - \boldsymbol{r}_1|$ is the separation. We can rewrite the mass dependence using

$$\begin{aligned} M^{3/2} - m_1^{3/2} - m_2^{3/2} &= m_1\left(\sqrt{M} - \sqrt{m_1}\right) + m_2\left(\sqrt{M} - \sqrt{m_2}\right) \\ &= m_1 m_2\left(\frac{1}{\sqrt{M} + \sqrt{m_1}} + \frac{1}{\sqrt{M} + \sqrt{m_2}}\right). \end{aligned} \tag{18}$$

The equations of motion of the two particle system are

$$\begin{cases} \dot{\boldsymbol{r}}_1 = \boldsymbol{v}_1, \\ \dot{\boldsymbol{r}}_2 = \boldsymbol{v}_2, \\ \dot{\boldsymbol{v}}_1 = \dfrac{2\sqrt{Ga_0}}{3}\dfrac{M^{3/2} - m_1^{3/2} - m_2^{3/2}}{m_1}\dfrac{\boldsymbol{r}_2 - \boldsymbol{r}_1}{|\boldsymbol{r}_1 - \boldsymbol{r}_2|^2}, \\ \dot{\boldsymbol{v}}_2 = \dfrac{2\sqrt{Ga_0}}{3}\dfrac{M^{3/2} - m_1^{3/2} - m_2^{3/2}}{m_2}\dfrac{\boldsymbol{r}_1 - \boldsymbol{r}_2}{|\boldsymbol{r}_1 - \boldsymbol{r}_2|^2}. \end{cases} \tag{19}$$

A numerical solution of this system of ordinary differential equations (ODE) is shown in Fig. 2. Here, it is compared with the results of our PM code, showing that the two methods agree

[1] `MONDPMesh` codebase: https://github.com/Joost987/MONDPMesh

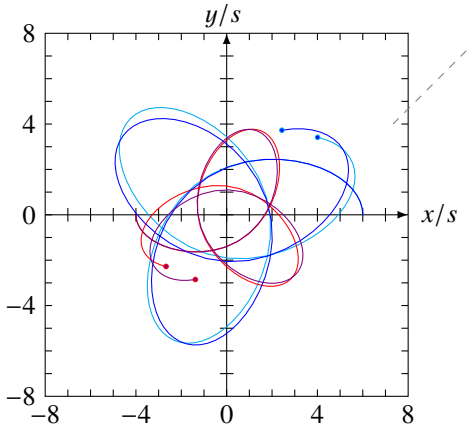

Fig. 2. Simulated orbits of the two-body system projected onto the xy plane for the mass ratio $m_1/m_2 = 3/2$. The mesh has a size of 128^3, and there are 100 time steps, each with four iterations for the MOND force. Red (m_1) and cyan (m_2) are the orbits obtained from numerical integration of Eqs. (19). Purple and blue are the orbits calculated with the PM code. The difference between the PM code and the ODEs is a slow dephasing of the orbits.

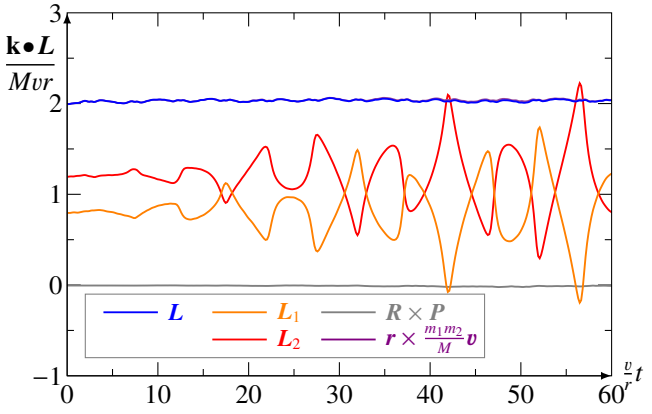

Fig. 3. Angular momentum vs. time t for the simulation shown in Fig. 2. These are constants of motion of the ODE system Eq. (19) and numerically constant (with machine precision, not shown) when integrated with Leapfrog. In the PM code, the total angular momentum (blue) is well conserved, but the single-particle angular momenta (orange and red) have large errors when the distance to the origin is smaller than one pixel.

reasonably well, but diverge ultimately. The system has the following constants of motion:

$$\boldsymbol{P} = m_1\boldsymbol{v}_1 + m_2\boldsymbol{v}_2,$$

$$\boldsymbol{R}(t) - \frac{\boldsymbol{P}}{M}t = \frac{m_1\boldsymbol{r}_1 + m_2\boldsymbol{r}_2}{M} - \frac{\boldsymbol{P}}{M}t,$$

$$\boldsymbol{R}(t) \times \boldsymbol{P},$$

$$\boldsymbol{L} = m_1\boldsymbol{r}_1 \times \boldsymbol{v}_1 + m_2\boldsymbol{r}_2 \times \boldsymbol{v}_2,$$

$$E = \frac{m_1v_1^2 + m_2v_2^2}{2} + 2\sqrt{Ga_0}\,\frac{M^{3/2} - m_1^{3/2} - m_2^{3/2}}{3}\log|\boldsymbol{r}_1 - \boldsymbol{r}_2|.$$

Because the PM is cubic, which breaks the local spherical symmetry, we expect that the total angular momentum does not remain constant in the simulations. How well angular momentum is conserved by our PM code is shown in Fig. 3. If the center of mass is located at the origin, $\boldsymbol{R} = \boldsymbol{0}$, the single-particle angular momenta $\boldsymbol{L}_1$ and $\boldsymbol{L}_2$ are also constant. However, the numerical error in $\boldsymbol{L}_1$ and $\boldsymbol{L}_2$ becomes large when the particle is within a distance of one pixel from the origin. The reason is that the center of mass may have drifted away from the origin by one pixel. When we subtract $\boldsymbol{R}$ from $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$, the errors disappear. For grid sizes 128^3, 256^3, and 512^3, the error in total angular momentum is 4%, 2%, and 1%, respectively.

The solution with the two particles in circular orbit about the center of mass is

$$\boldsymbol{r}_1 = \frac{m_2 r}{M}(\mathbf{i}\cos\omega t + \mathbf{j}\sin\omega t), \qquad \boldsymbol{r}_2 = \frac{-m_1 r}{M}(\mathbf{i}\cos\omega t + \mathbf{j}\sin\omega t),$$

$$\boldsymbol{v}_1 = \frac{-m_2 v}{M}(\mathbf{i}\sin\omega t - \mathbf{j}\cos\omega t), \qquad \boldsymbol{v}_2 = \frac{m_1 v}{M}(\mathbf{i}\sin\omega t - \mathbf{j}\cos\omega t),$$

where $r = |\boldsymbol{r}_2 - \boldsymbol{r}_1| = r_1 + r_2$ is the separation, $v = |\boldsymbol{v}_2 - \boldsymbol{v}_1|$ is the relative velocity, and ω is the orbital frequency. With Huygens' law of the centripetal force applied to the motion of the reduced coordinate r, we obtain for the force

$$\frac{m_1v_1^2}{r_1} = \frac{m_2v_2^2}{r_2} = \frac{m_1m_2v^2}{Mr} = F_{12},$$

and for the kinetic energy

$$\frac{m_1v_1^2 + m_2v_2^2}{2} = \frac{m_1m_2v^2}{2M} = \frac{M\overline{v^2}}{2} = E_{\rm kin}.$$

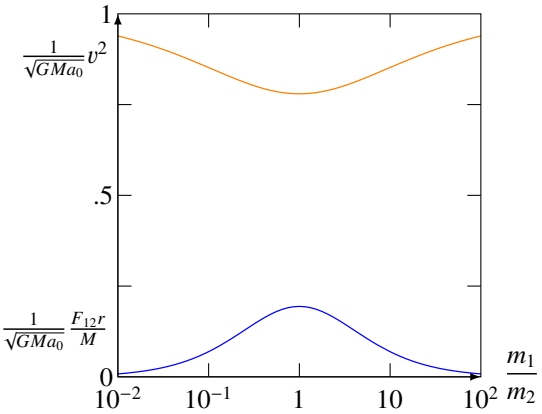

Fig. 4. Force times radius (blue) and squared velocity (orange) vs. mass ratio for the two-body system. The horizontal axis is logarithmic, making the curves symmetric about $m_1/m_2 = 1$.

Using Eqs. (16)–(18), we find that the velocity v of the reduced motion and the orbital frequency ω are given by the expression

$$v^2 = (\omega r)^2 = \frac{M^2\overline{v^2}}{m_1m_2} = \frac{2\sqrt{GMa_0}}{3}\left(\frac{1}{1+\sqrt{m_1/M}} + \frac{1}{1+\sqrt{m_2/M}}\right). \tag{20}$$

The velocities are independent of the separation. The theoretical dependence of the velocity on the mass ratio is plotted in Fig. 4. Figure 5 shows the numerical simulation of our PM code for two particles with the initial relative velocity from Eq. (20). It convincingly shows that motion is circular, which verifies Milgrom's two-particle force law. It also illustrates that we can obtain reasonable subpixel resolution.

4.2. Example II: Ring galaxy

Again, we tested the code with a system that was solved analytically by Milgrom (1994): a central mass m_0, surrounded by a ring with a radius r_2 with N particles with a mass m (i.e., the system has $N + 1$ particles). A beautiful realization of such a ring galaxy is Hoag's Object (Schweizer et al. 1987). The central mass is at

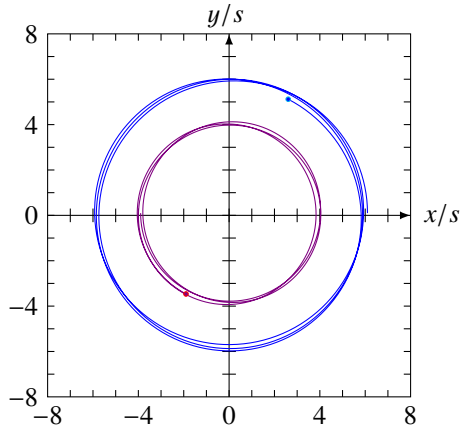

Fig. 5. Simulated orbits of two bodies with a mass ratio $m_1/m_2 = 3/2$ as in Fig. 2, but with 400 time steps. The initial velocities are given by Eq. (20) and should result in circular orbits. These are plotted in Fig 4. This simulation thus confirms the force formula Eq. (17) for the deep MOND case and validates the algorithm. It also illustrates how well the algorithm performs on the subpixel scale.

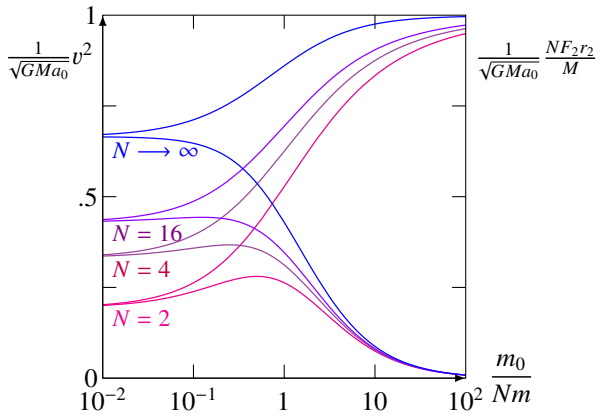

Fig. 6. Force times radius (lower curves) and squared velocity (higher curves) vs. the mass ratio for particles on a regular N-gon orbiting a central body in circular motion. From Eqs. (21)–(22), where m_0 is the central mass and Nm is the ring mass.

rest at the origin. We created a rotating ring of N particles on a regular N-gon ($N \geq 2$),

$$\boldsymbol{r}_i = r_2\mathbf{i}\cos(\omega t + 2\pi i/N) + r_2\mathbf{j}\sin(\omega t + 2\pi i/N),$$
$$\boldsymbol{v}_i = -v\mathbf{i}\sin(\omega t + 2\pi i/N) + v\mathbf{j}\cos(\omega t + 2\pi i/N).$$

The force on the particles in the ring is given by

$$F_2 = \frac{2\sqrt{Ga_0}}{3Nr_2}\left(M^{3/2} - m_0^{3/2} - Nm^{3/2}\right). \tag{21}$$

Here, $M = m_0 + Nm$. We can now write

$$M^{3/2} - m_0^{3/2} = m_0\left(\sqrt{M} - \sqrt{m_0}\right) + Nm\sqrt{M}$$
$$= MNm\left(\frac{1}{\sqrt{M} + \sqrt{m_0}} + \frac{\sqrt{m_0}}{M}\right).$$

The velocity of the ring particles and the orbital frequency are found by equating the force Eq. (21) to $F_2 = mv^2/r_2 = m\omega^2 r_2$,

$$v^2 = \frac{M\overline{v^2}}{Nm} = \frac{2\sqrt{GMa_0}}{3}\left(\frac{1}{1+\sqrt{m_0/M}} + \sqrt{m_0/M} - \sqrt{m/M}\right). \tag{22}$$

The force times radius and squared velocity is plotted in Fig. 6. The result of the simulation for $N = 100$ particles is plotted in Fig. 7. If the mass of the ring is not very low compared to the central mass, the system is unstable, and the ring breaks apart after a few orbits.

4.3. Eample III: The isothermal sphere in MOND

Our third application is the simulation of a spherically symmetric system in thermodynamic equilibrium. This could be a globular cluster with up to 10^6 stars or a galaxy cluster with up to 10^3 galaxies. These system are in the deep MOND regime or in the crossover regime (accelerations around a_0).

A system of N point particles with equal mass m in hydrostatic equilibrium with a temperature T has a velocity dispersion given by

$$\frac{3k_BT}{2m} = \frac{\overline{\boldsymbol{v}\bullet\boldsymbol{v}}}{2} = \frac{\overline{v^2}}{2} = \frac{\sqrt{GMa_0}}{3}. \tag{23}$$

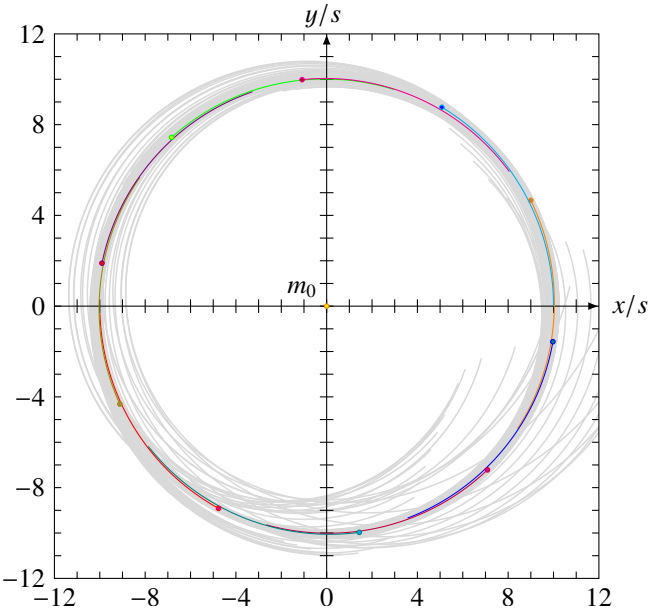

Fig. 7. Simulated orbits for a ring of $N = 100$ particles with a mass m around a central mass m_0 for a mass ratio of $m_0/Nm = 3/2$ as in Fig. 2. The orbits of the first 80 time steps of 10 particles are shown in color. In gray we plot 576 steps for all particles. The initial velocities are given by Eq. (22), plotted in Fig. 6, which in theory results in circular orbits. Although this confirms the force formula Eq. (21), numerical round-off errors on the subpixel scale propagate and cause the orbits to diverge.

The last equality is the thermodynamic limit of Eq. (16). For a spherical system, the particle number density in terms of the potential is given by

$$\frac{n(r)}{n(0)} = \exp\left(\frac{-m\phi(r)}{k_BT}\right) = \exp\left(\frac{-3\phi(r)}{\overline{v^2}}\right) = \exp\left(\frac{-9\phi(r)}{2\sqrt{GMa_0}}\right). \tag{24}$$

Here, we used hydrostatic equilibrium and the ideal gas law. Substitution of the mass density $\rho_B(r) = mn(r)$ and Eq. (24) in the MOND Eq. (3) gives the following equation for the potential:

$$\frac{1}{a_0r^2}\frac{\mathrm{d}}{\mathrm{d}r}\left(r\frac{\mathrm{d}\phi}{\mathrm{d}r}\right)^2 = 4\pi G\rho_B(0)\exp\left(\frac{-9\phi(r)}{2\sqrt{GMa_0}}\right). \tag{25}$$

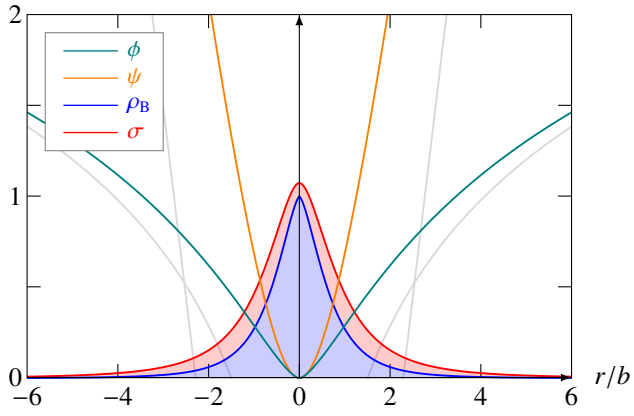

Fig. 8. Potentials and mass distributions for the isothermal state. In teal and orange we show the potential $\phi(r)$ as given by Eq. (26) and the deflection potential $\psi = \int \phi \mathrm{d}z$ for the line of sight passing the center at distance r, determining weak gravitational lensing. The asymptotes are shown (in gray) and the units are $\sqrt{GMa_0}\, b^j$, $j = 0, 1$. In blue and red we show the density $\rho_\mathrm{B}(r)$ as given by Eq. (29) and the surface mass density $\sigma = \int \rho_\mathrm{B} \mathrm{d}z$ in units of $3M/4\pi b^j$, $j = 3, 2$.

The solution of this equation is found to be

$$\phi(r) = \frac{2\sqrt{GMa_0}}{3} \log\left(1 + \frac{r^{3/2}}{b^{3/2}}\right), \tag{26}$$

$$\boldsymbol{g}_\mathrm{M}(\boldsymbol{r}) = -\boldsymbol{r}\sqrt{\frac{GMa_0}{b^3 r}} \left(1 + \frac{r^{3/2}}{b^{3/2}}\right)^{-1}, \tag{27}$$

$$M(r) = \frac{Mr^3}{b^3}\left(1 + \frac{r^{3/2}}{b^{3/2}}\right)^{-2}, \tag{28}$$

$$\rho_\mathrm{B}(r) = \frac{3M}{4\pi b^3}\left(1 + \frac{r^{3/2}}{b^{3/2}}\right)^{-3}. \tag{29}$$

Here, b is a parameter with the dimension of a length. When the function in Eq. (26) is substituted in Eq. (25), we can solve for this parameter b in terms of the mass M and the central density $\rho_\mathrm{B}(0)$. Hence, we have found a two-parameter family of solutions. The function $M(r)$ in Eq. (28) represents the mass inside a sphere of radius r, and $M = \lim_{r\to\infty} M(r)$ is the total mass. The physical meaning of the parameter b becomes clear when we calculate the mass of a homogeneous sphere with radius b, with the constant density equal to the value $\rho(0)$ at the center of the isothermal distribution, given in Eq. (29). This mass is precisely equal to the total mass

$$M = 4\pi \int_0^\infty \rho_\mathrm{B}(r) r^2 \mathrm{d}r = 4\pi \int_0^b \rho_\mathrm{B}(0) r^2 \mathrm{d}r = \frac{4\pi b^3}{3}\rho_\mathrm{B}(0).$$

The density is finite at the origin and also drops off sufficiently fast to zero to allow normalization: No cutoff is needed to keep the total mass finite. These physically convenient properties are absent from the isothermal solution in Newtonian gravity. The potential and density and the resulting deflection potential and surface-mass density are plotted in Fig. 8. Kinetic energy and potential energy are, with Eq. (23),

$$E_\mathrm{kin} = \frac{M\overline{v^2}}{2}, \quad E_\mathrm{pot} = 4\pi \int_0^\infty \rho_\mathrm{B}(r)\phi(r) r^2 \mathrm{d}r = \frac{3M\overline{v^2}}{2}.$$

The solution given in Eqs. (26)–(29) is applicable for $g_\mathrm{M} \ll a_0$. Because the maximum acceleration is found at $r = b/2^{2/3}$, the requirement becomes $GM/b^2 \ll 9a_0/2^{4/3}$. The Virgo cluster, for example, satisfies this. Equation (29) could be a realistic description for spherical gas clouds, elliptical galaxies, and clusters.

We derived the formula for the apparent mass density, which would give the apparent dark matter distribution function. Equations (11)–(12) give

$$M_\mathrm{A}(r) = \sqrt{\frac{Ma_0}{G}}\, r \left(1 + \frac{b^{3/2}}{r^{3/2}}\right)^{-1},$$

$$\rho_\mathrm{A}(r) = \frac{1}{4\pi}\sqrt{\frac{Ma_0}{Gb^3 r}} \left(\frac{5}{2} + \frac{r^{3/2}}{b^{3/2}}\right)\left(1 + \frac{r^{3/2}}{b^{3/2}}\right)^{-2}.$$

The total apparent mass diverges as $r \longrightarrow \infty$. Whereas the actual density ρ_B is smooth at the center, the apparent mass density ρ_A has a cusp at the core. The isothermal solution shows typical deep MOND behavior: Only the total baryonic mass M determines the velocity distribution, with a central density that can be varied independently. The lower the density at the center, the larger the system system size and the higher the apparent dark matter mass.

We now study the dynamics of the isothermal solution with our PM code. We need the cumulative distribution function for the radial probability density. According to Eq. (28), this is

$$\frac{M(r)}{M} = \left(1 + \frac{b^{3/2}}{r^{3/2}}\right)^{-2}.$$

In order to obtain random realizations, our point masses were found from seven random numbers $\xi_1 \in (-1, 1)$, $\eta_1, \eta_2, \eta_3 \in (0, 1)$, $\zeta_1, \zeta_2, \zeta_3 \in (0, 2\pi)$, sampled homogeneously from these intervals and the formulae

$$\boldsymbol{r}_i = b\left(\sqrt{1-\xi_1^2}\left(\mathbf{i}\cos\zeta_1 + \mathbf{j}\sin\zeta_1\right) + \mathbf{k}\xi_1\right)\left(\frac{1}{\sqrt{\eta_1}} - 1\right)^{-2/3}, \tag{30}$$

$$\boldsymbol{v}_i = \sqrt{-\tfrac{2}{3}\overline{v^2}\log\eta_2}\left(\mathbf{i}\cos\zeta_2 + \mathbf{j}\sin\zeta_2\right) + \sqrt{-\tfrac{2}{3}\overline{v^2}\log\eta_3}\,\mathbf{k}\cos\zeta_3.$$

The Box-Muller transform was used to generate the Gaussian values of the Maxwell-Boltzmann velocity distribution. Figures 9 and 10 show how close the realizations are to the cumulative distributions for the radial coordinate and for the velocity. Figure 11 shows the realizations projected onto the celestial plane and the resulting apparent dark matter surface density. For $N = 10^3$ or larger, the potential is nearly spherical, and it is well described with our analytic solution.

We solved the evolution of N particles in the external potential Eq. (26) as a system of ODEs, and we simulated the N particles in MONDian gravity using the PM code, both with the same initial distribution. The two models are comparable because both must behave identical in the thermodynamic limit $N \longrightarrow \infty$. We also tested whether the distributions are stationary, as they should be for large N. In the external potential, the energies and angular momenta of the individual particles are strictly conserved, and so is the total energy and angular momentum. The single-particle kinetic energies and momenta oscillate, and so does the total kinetic energy and the total momentum. We compare this with the exact system of Eqs. (1)–(3). Here, the total energy, total momentum, and total angular momentum are conserved. The PM code simulates this system. The fluctuations in the energies are shown in Fig. 12. Because the mesh in our PM approach breaks spherical symmetry, angular momentum also fluctuates slightly.

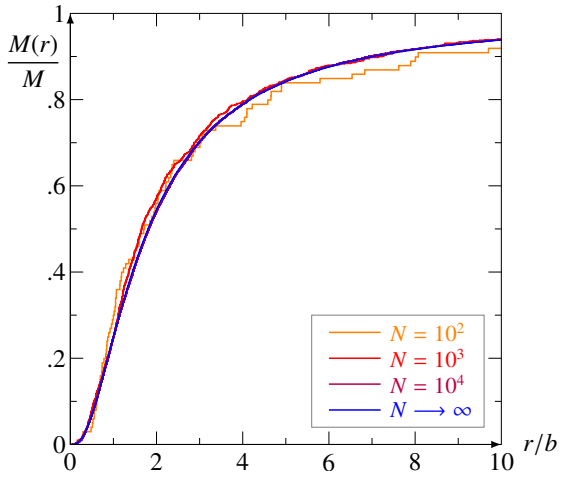

Fig. 9. Cumulative mass $M(r)$ of the spherical isothermal galaxy vs. r. The (free) parameter b determines the galaxy size. The (blue) curve is the theoretical result given by Eq. (28). The realizations (orange, red, and purple) are calculated from Eq. (30).

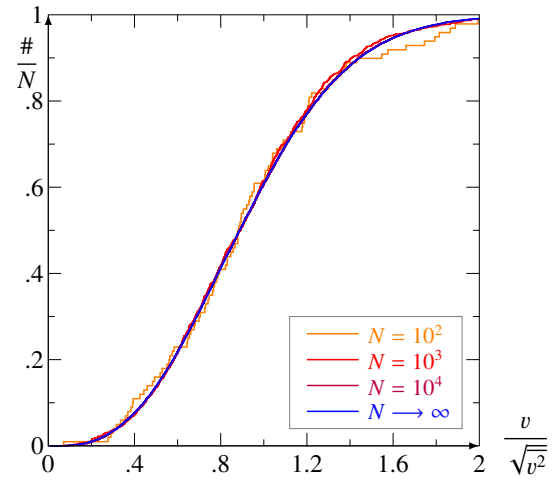

Fig. 10. Cumulative velocity distribution for the spherical isothermal galaxy. This is the fraction $\#(\leq v)/N$ of particles with velocities below v vs. the value v of this upper bound. The realizations (orange, red, and purple) are calculated from Eq. (30).

5. Discussion

5.1. Accuracy of the method

The inaccuracies in the algorithm arise from the fact that we used a fixed grid. The discrete grid has n pixels in each dimension with a pixel width s and has a finite volume $n^3 s^3$.

The use of FFTs in the calculation of the potentials and acceleration fields generally gives rise to anti-aliasing: The functions obtain nonphysical short-wavelength oscillations. This effect is particularly strong when the point particles are represented with isolated nonzero pixel values in the mass density, but also when the CIC method is used. Smoothing with a minimum variance of one pixel width s in a spherical domain of radius $4s$ needs to be used to overcome this problem. Although the discretization creates a minimum size, we reach a subpixel spatial resolution by using interpolation. We found that the retrieval of the forces on the particles with a Gaussian smoothing gives the most accurate results. To further limit the aliasing and increase accuracy, we did not use the numerical acceleration field, but calculated the potential, and we derived the forces instead using the finite-difference gradient of this scalar function.

The runtime of our code has a time complexity of $O(Nn^3 \log n)$, hence it scales linearly in the number of particles and almost linearly in the number of pixels because we made use of the FFT. The runtime as a function of particle count is plotted in Fig. (13). For a 128^3 grid, the evaluation of the accelerations or/and densities starts to dominate the runtime at $N = 10^3$. Therefore, in the general case, we expect that the crossover occurs at a critical density of $10^3/128^3 \approx 1/10^3$ particles per pixel.

5.2. Improvements of the code

The code could be made faster, allowing simulations with higher resolution/larger volumes, with smaller time steps/longer simulated times, or with more particles. We list below a few ideas for possible improvements that we did not implement.

(i) The code is currently written in Python, using the FFTW library, which is a C library for the FFTs. The code could be entirely rewritten in a compiler programming language such as C, and compiled into fast-running machine code.

(ii) A primitive data type can be chosen for the floats with half-precision, used by GPUs (with the cuFFTW library). The machine precision would be 10^{-3}, which is close to the unavoidable error due to self-gravity. Its implementation should reduce memory use and speed up the code.

(iii) Both the potential solver and the particle propagator in the code can be parallelized. The calculation of the potential requires 42 FFTs (3 scalar plus 13 vector FFTs). These three-dimensional FFTs need to be performed sequentially. For each direction, we have n^2 normal FFTs of a vector with n elements. These can be performed in parallel. The increments of positions and velocities of the N particles are independent, hence their calculations can also be made in parallel. Parallelizing the updating of the density is less straightforward. If the distance between two particles is smaller than 8 pixels, the density needs to be updated one particle at a time.

(iv) If the density function changes only slightly at each time step, the potential solver needs just one iteration when it is seeded with the previous solution. This strategy can be employed when the spatial steps are smaller than the width of the weight functions (i.e. for small time steps), or when nearby particles have mostly overlapping weight functions (i.e. for high number densities). Densities are also smooth when the matter is modeled as a fluid. We found that the PDE solver indeed sped up by a factor three.

(v) If changes in the fields at the time steps are small, we can calculate the increments $\Delta\rho_{\rm B}$, $\Delta \boldsymbol{g}_{\rm N}$, $\Delta \boldsymbol{g}_{\rm M}$, $\Delta \boldsymbol{F}$, and $\Delta \boldsymbol{H}$, instead of the full fields. This will improve the accuracy and reduce the required number of significant digits so that half-precision can be used. For the nonlinear Eqs. (8) and (10), we can use the linearizations

$$\Delta \boldsymbol{g}_{\rm M}(\boldsymbol{r}) = \nu\left(\frac{F}{a_0}\right)\Delta \boldsymbol{F} + \nu'\left(\frac{F}{a_0}\right)\frac{\boldsymbol{F}\bullet\Delta\boldsymbol{F}}{a_0 F}\boldsymbol{F},$$

$$\Delta \boldsymbol{F}(\boldsymbol{r}) = \mu\left(\frac{g_{\rm M}}{a_0}\right)\Delta \boldsymbol{g}_{\rm M} + \mu'\left(\frac{g_{\rm M}}{a_0}\right)\frac{\boldsymbol{g}_{\rm M}\bullet\Delta\boldsymbol{g}_{\rm M}}{a_0 g_{\rm M}}\boldsymbol{g}_{\rm M}.$$

5.3. Applications targeted to special systems

We validated the PM code by simulating the two-body system and ring systems. For applications of systems with a high symmetry like this, however, the code could be adapted to make it

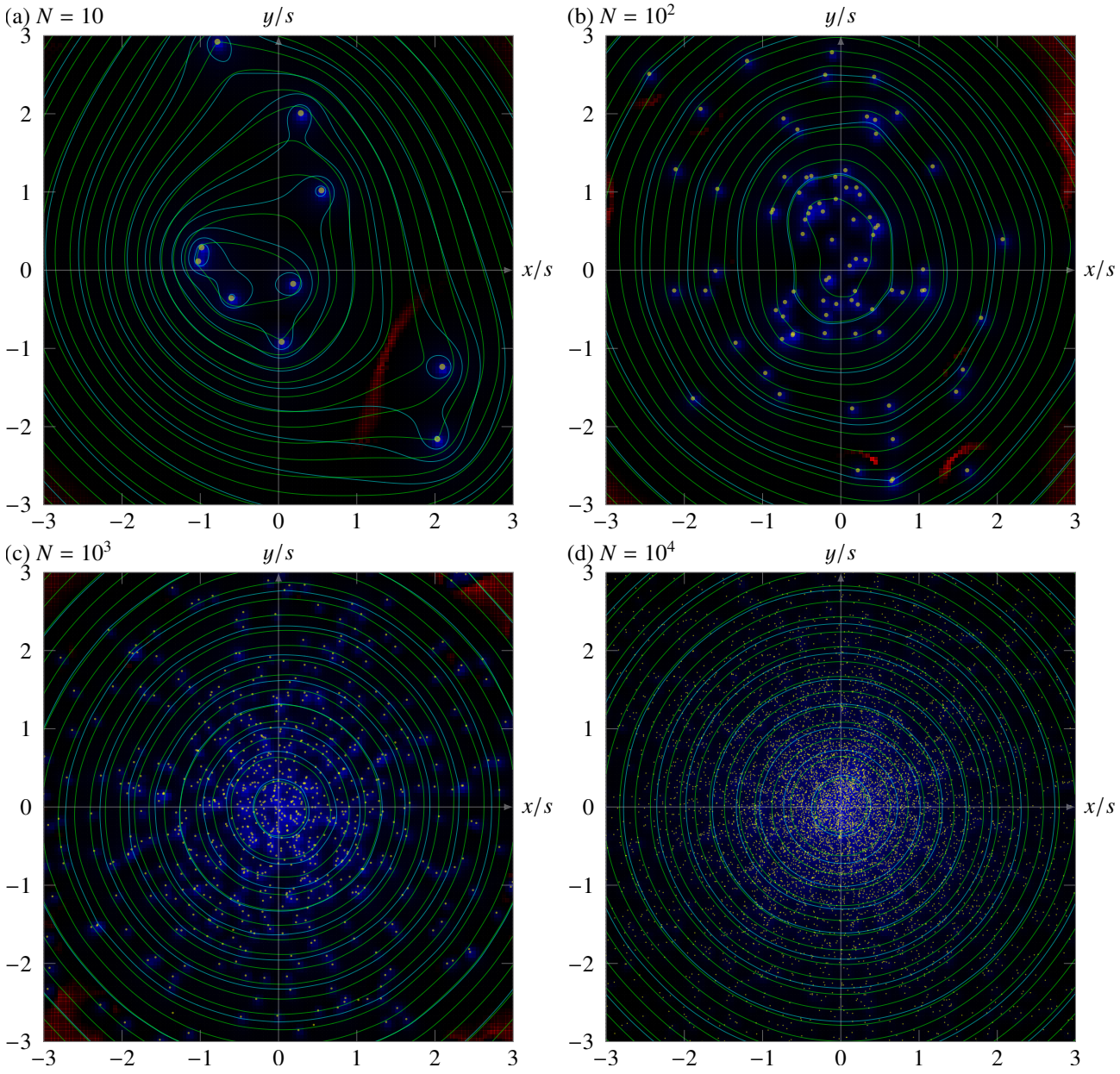

Fig. 11. Surface-mass densities $\sigma(x, y)$ and deflection potentials $\psi(x, y)$ for a globular cluster from the central 128×128 part of a simulated cube of 256^3 pixels. The yellow dots are the stars that define the actual density. The resulting apparent dark matter surface density is plotted in blue according to Eq. (11). Negative value are shown in red (Milgrom 1986). They are found for $N \leq 10^2$. The equipotential curves of the deflection potential for Newtonian and Milgromian gravity are plotted in green and cyan. Although the densities are far from circularly symmetric, the deflection potential is nearly circular for $N \geq 10^3$. This indicates that the retrieval of a clumpy mass distribution from weak lensing may be problematic for large N.

more accurate and much faster. For example, a binary system can be simulated using ODEs and does not require a PDE solver, as shown in Sect. 4.1 for the deep MOND system Eq. (19). For the general case, the interpolation function μ is different, but the system is still governed by a two-body force that is a function of distance and the masses. One could now use the PDE solver of our code to calculate the force. This numerical force function can then be used in the ODE system for the actual simulations of the binary.

In absence of an external field, the two-body system has a perfectly cylindrical symmetry: The potential is a function of the distance to and height on the axis of symmetry. By writing the PDE in cylindrical coordinates and using the Hankel transform, we limit the number of transforms in the PDE solver. The evaluation of the force function could thus be calculated fast, allowing one to test various choices of interpolation functions. For other systems with cylindrical symmetry, such as the motion of a test particle in a disk galaxy, this method can also be used to enhance the accuracy and speed (see Platschorre 2019, for a study of rotation curves using the Hankel transform in this way).

The external field can be included in our PM code by fixing a plane at maximum distance from the particles in the cube where a permanent dipolar mass distribution is placed, that is, a layer of pixels with mass M and an adjacent pixel layer of mass $-M$.

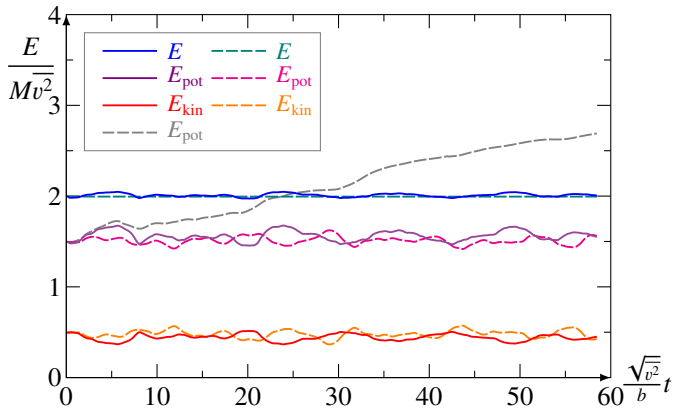

Fig. 12. Energies vs. time for $N = 100$ particles on a 256^3 grid. The kinetic, potential, and total energy for the self-consistent interacting system of the particles from the PM code are shown as solid curves (red, purple, and blue). The kinetic, potential, and total energy from the ODE system of the particles in the external potential are shown as dashed curves (orange, violet, and teal). The fluctuations in the total energies are purely numerical, and the other energies have thermal fluctuations. The dashed gray curve is $E_{\rm pot} = \int \rho_{\rm B}\phi \mathrm{d}x\mathrm{d}y\mathrm{d}z$ with the PM matter density and Eq. (26), hence substituting a finite N solution in the potential obtained in the thermodynamic limit. We subtracted the self-energies.

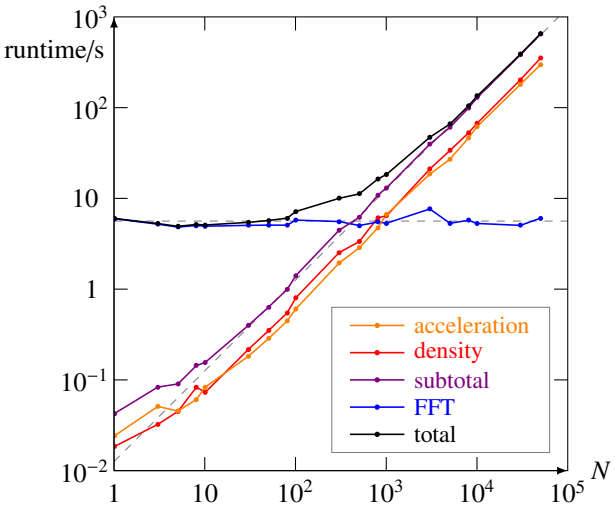

Fig. 13. Runtime vs. particle count N for a 128^3 grid with the Python code on an Intel Core i7-9750H. The time spend on the FFTs (blue) is roughly constant, and the time for the evaluation of the particle accelerations and/or densities grows linear with N (orange, red, and purple). The total runtime shows a crossover around 10^3 particles (black). The dashed lines are guides with slopes of zero and one.

When the external field is included, one can again evaluate the force function for the two-body problem numerically. Now, the force also depends on the polar angle (the angle between the direction of the external field and the vector $\boldsymbol{r}_2 - \boldsymbol{r}_1$), and not just on the interparticle distance.

5.4. Extensions and future research directions

Extensions to the code could make the code useful for more realistic modeling and other fields. Some examples are listed below.

(i) Close encounters. If the ratio Gm/s^2 is much higher than a_0, the field on the pixel scale is Newtonian. It may be possible to obtain subpixel resolution for this case by calculating the near field with a particle-particle component.

(ii) Hydrodynamics. In galaxy clusters, the intergalactic gas component dominates the total mass contribution from the galaxies. One could implement Euler's equations for inviscid flow inside the MOND acceleration field. The evolution of the density $\rho_{\rm B}$, pressure p, and flow field $\boldsymbol{u}$ can be calculated from the system

$$\begin{cases} \dot{\rho}_{\rm B} = \boldsymbol{u} \bullet \mathscr{F}^{-1}(\mathrm{i}\boldsymbol{k}\widehat{\rho_{\rm B}}) + \rho_{\rm B}\mathscr{F}^{-1}(\mathrm{i}\boldsymbol{k} \bullet \widehat{\boldsymbol{u}}), \\ \dot{p} = \boldsymbol{u} \bullet \mathscr{F}^{-1}(\mathrm{i}\boldsymbol{k}\widehat{p}) + \gamma p\mathscr{F}^{-1}(\mathrm{i}\boldsymbol{k} \bullet \widehat{\boldsymbol{u}}), \\ \dot{\boldsymbol{u}} = \mathscr{F}^{-1}(\mathrm{i}\boldsymbol{k}\widehat{\mathcal{E}}) - \boldsymbol{u} \times \mathscr{F}^{-1}(\mathrm{i}\boldsymbol{k} \times \widehat{\boldsymbol{u}}) + \mathscr{F}^{-1}(\mathrm{i}\boldsymbol{k}\widehat{p})/\rho_{\rm B} + \boldsymbol{g}_{\rm M}. \end{cases}$$

Here, $\widehat{\mathcal{E}} = \mathscr{F}(\boldsymbol{u} \bullet \boldsymbol{u})/2$ is the transformed specific kinetic energy, and γ is the polytropic constant. These are the Euler equations in Lamb form, where the derivatives are calculated using Fourier transforms. By running these on a GPU with cuFFTW in half-precision floats, one gains a speed-up of at least one order of magnitude.

(iii) Cosmology. In simulations of structure formation in the early Universe, one can introduce comoving coordinates to model the expansion of the Universe. This amounts to the equations for particle motion (Peebles 1993)

$$\dot{\boldsymbol{r}}_i = \boldsymbol{v}_i/a, \quad \dot{\boldsymbol{v}}_i = \boldsymbol{g}_{\rm M}(\boldsymbol{r}_i) - \boldsymbol{v}_i\dot{a}/a,$$

and the density in Eq. (5) can be replaced with $a\rho_{\rm B}(\boldsymbol{r}) - a\overline{\rho_{\rm B}}$, where the scale factor of the expanding Universe is given by $a(t)$.

6. Conclusions

We described, coded, and tested an algorithm for the simulation of the dynamical evolution of N bodies for AQUAL-MOND. MOND is an alternative model for the gravity between stars, galaxies, and galaxy clusters, where standard theory requires CDM. We now list our final conclusions:

(i) The MOND Eq. (3) is shown to be equivalent to the system Eqs. (5)–(10) of linear PDEs plus an algebraic equation. Therefore, linear PDE solvers can be used;

(ii) Because we have a linear system, we can use Fourier transforms to solve it. Our PM code relies on 42 FFTs, where a standard Poisson solver needs only two. The use of FFTs makes it faster than the finite-element codes N-MODY and RayMOND (Londrillo & Nipoti 2009; Candlish et al. 2014), especially for large grids;

(iii) Four iterations are required to obtain an accuracy of 1%;

(iv) Self-gravity is the dominant cause of the numerical error. We needed a Gaussian smoothing kernel of one pixel width over a spherical domain with a radius of four pixels. The standard CIC method used in Poisson solvers cannot be used;

(v) When the number of particles per pixel exceeds $1/10^3$, particle propagation takes more time than the PDE solver;

(vi) We derived the analytic isothermal state Eqs. (26)–(29) for the spherical deep MOND case. It is a two-parameter family, with a mass and a length scale. Because the deflection potential deviates significantly from that for a point mass, the solution could be relevant for elliptic galaxies and clusters;

(vii) We tested our PM code for systems where deep MOND analytic solutions (by Milgrom 1994) are known and found good agreement. We also tested the isothermal state by creating a Maxwell-Boltzmann distribution with density Eq. (29) and verified that the system is stationary.

Promising applications of the *N*-body MOND code include the numerical comparison with actual astrophysical observations on the Solar System, wide-binaries, and galaxy clusters.

References

Angus, G. W., & Diaferio, A. 2011, MNRAS, 417, 941
Angus, G. W., Diaferio, A., & Kroupa, P. 2011, MNRAS, 416, 1401
Asencio, E., Banik, I., & Kroupa, P. 2020, MNRAS, 500
Asencio, E., Banik, I., Mieske, S., et al. 2022, MNRAS, 515, 2981
Asencio, E., Banik, I., & Kroupa, P. 2023, ApJ, 954, 162
Asgari, M., Chieh-An, L., Joachimi, B., et al. 2021, A&A, 645, A104
Banik, I., & Zhao, H. 2018, MNRAS, 480, 2660
Banik, I., & Zhao, H. 2022, Symmetry, 14, 1331
Bekenstein, J. D. 2004, Phys. Rev. D, 70, 083509
Bekenstein, J., & Milgrom, M. 1984, ApJ, 286, 7
Blanchet, L., & Novak, J. 2011, MNRAS, 412, 2530
Brada, R., & Milgrom, M. 1999, ApJ, 519, 590
Brouwer, M. M., Oman, K. A., Valentijn, E. A., et al. 2021, A&A, 650, A113
Bullock, J. 2013, Notes on the Missing Satellites Problem, ed. D. Martínez-Delgado, Canary Islands Winter School of Astrophysics (Cambridge University Press), 95
Candlish, G. N., Smith, R., & Fellhauer, M. 2014, MNRAS, 446, 1060
Chae, K.-H. 2023, ApJ, 952, 128
Chae, K.-H., Lelli, F., Desmond, H., et al. 2020, ApJ, 904, 51
Chiba, R., & Schönrich, R. 2021, MNRAS, 505, 2412
Comerón, S., Trujillo, I., Cappellari, M., et al. 2023, A&A, 675, A143
de Blok, W. J. G. 2010, Adv. Astron., 2010, 789293
de Nijs, J. V. 2023, Bachelor Thesis, TU Delft, Netherlands
de Nijs, J. V., Visser, P. M., & Eijt, S. W. 2023, Astrophysics Source Code Library [record ascl:2311.006]
Eappen, R., Kroupa, P., Wittenburg, N., Haslbauer, M., & Famaey, B. 2022, MNRAS, 516, 1081
Fabian, A. C. 1994, Cluster Cooling Flows, ed. W. C. Seitter (Dordrecht: Springer Netherlands), 163
Haslbauer, M., Banik, I., & Kroupa, P. 2020, MNRAS, 499, 2845
Hernandez, X. 2023, MNRAS, 525, 1401
Hernandez, X., Cortés, R. A. M., Allen, C., & Scarpa, R. 2019, Int. J. Mod. Phys. D, 28, 1950101
Hockney, R. W., & Eastwood, J. W. 1988, Computer Simulation using Particles (Bristol: Hilger)
Iorio, L. 2009, Astrophys. Space Sci., 323, 215
Iorio, L. 2017, Eur. Phys. J. C, 77, 1
Katz, H., McGaugh, S., Teuben, P., & Angus, G. W. 2013, ApJ, 772, 10
Kroupa, P. 2015, Can. J. Phys., 93, 169
Kroupa, P., Jerabkova, T., Thies, I., et al. 2022, MNRAS, 517, 3613
Labbé, I., van Dokkum, P., Nelson, E., et al. 2023, Nature, 616, 266
Llinares, C., Knebe, A., & Zhao, H. 2008, MNRAS, 391, 1778
Londrillo, P., & Nipoti, C. 2009, Mem. S.A.It. Suppl., 13, 89
Lüghausen, F., Famaey, B., & Kroupa, P. 2015, Can. J. Phys., 93, 232
McGaugh, S. S., Schombert, J. M., Bothun, G. D., & De Blok, W. 2000, ApJ, 533, L99
McGaugh, S. S., Barker, M. K., & de Blok, W. 2003, ApJ, 584, 566
Meneghetti, M., Davoli, G., Bergamini, P., et al. 2020, Science, 369, 1347
Milgrom, M. 1983, ApJ, 270, 365
Milgrom, M. 1986, ApJ, 306, 9
Milgrom, M. 1994, ApJ, 429, 540
Milgrom, M. 1997, Phys. Rev. E, 56, 1148
Milgrom, M. 2010, MNRAS, 403, 886
Milgrom, M. 2014, Phys. Rev. D, 89, 024016
Milgrom, M., & Sanders, R. H. 2008, ApJ, 678, 131
Moreno, J., Danieli, S., Bullock, J. S., et al. 2022, Nat. Astron., 6, 496
Nagesh, S. T., Kroupa, P., Banik, I., et al. 2022, MNRAS, 519, 5128
Nipoti, C., Londrillo, P., & Ciotti, L. 2007, ApJ, 660, 256
Oehm, W., Thies, I., & Kroupa, P. 2017, MNRAS, 467, 273
Paučo, R. 2017, A&A, 603, A11
Paučo, R., & Klačka, J. 2016, A&A, 589, A63
Peebles, P. J. E. 1993, Principles of Physical Cosmology (Princeton University Press)
Peng, B., Wang, T., Jin, X., & Wang, C. 2016, Int. J. Reconfig. Comput., 2016, 1
Pitjeva, E., & Pitjev, N. 2013, MNRAS, 432, 3431
Pittordis, C., & Sutherland, W. J. 2019, MNRAS, 488, 4740
Planck Collaboration VI. 2020, A&A, 641, A6
Platschorre, A. 2019, Bachelor Thesis, TU Delft, The Netherlands
Riess, A. G., Casertano, S., Yuan, W., et al. 2021, ApJ, 908, L6
Roshan, M., Banik, I., Ghafourian, N., et al. 2021a, MNRAS, 503, 2833
Roshan, M., Ghafourian, N., Kashfi, T., et al. 2021b, MNRAS, 508, 926
Rubin, V. C., & Ford, W. Kent, J. 1970, ApJ, 159, 379
Rubin, V. C., Ford Jr, W. K., & Thonnard, N. 1980, ApJ, 238, 471
Schweizer, F., Ford, W. Kent, J., Jedrzejewski, R., & Giovanelli, R. 1987, ApJ, 320, 454
Skordis, C., & Złośnik, T. 2019, Phys. Rev. D, 100, 104013
Skordis, C., & Złośnik, T. 2021, Phys. Rev. Lett., 127, 161302
Teyssier, R. 2002, A&A, 385, 337
Wittenburg, N., Kroupa, P., & Famaey, B. 2020, ApJ, 890, 173
Wittenburg, N., Kroupa, P., Banik, I., Candlish, G., & Samaras, N. 2023, MNRAS, 523, 453
Yoon, Y., & Darriba, L. A. 2020, Classical Quant. Grav., 37, 135007
Zwicky, F. 1933, Helv. Phys. Acta, 6, 110
Zwicky, F. 1937, ApJ, 86, 217

Appendix A: Table of symbols

Table A.1. List of symbols and notation.

symbol	quantity
n	pixel count for each axis
s	pixel width
$L = ns$	system size
N	particle count
$i = 1 \dots N$	particle index
t	time
ω	orbital angular frequency
$\boldsymbol{r} = x\mathbf{i} + y\mathbf{j} + z\mathbf{k}$	coordinate vector
r	distance
b	width of isothermal
$\boldsymbol{k}$	wave vector
$k = \lvert\boldsymbol{k}\rvert$	length of the vector
m_i	mass of particle i
$\boldsymbol{r}_i(t)$	position of particle i
$\boldsymbol{v}_i(t) = \dot{\boldsymbol{r}}_i$	velocity of particle i
$\boldsymbol{g}_i(t) = \dot{\boldsymbol{v}}_i$	acceleration of particle i
$\boldsymbol{R}(t)$	center of mass
$\boldsymbol{P}$	total momentum
$\boldsymbol{L}$	total angular momentum
$E_{\rm pot}$, $E_{\rm kin}$	potential, kinetic energy
$E = E_{\rm pot} + E_{\rm kin}$	total energy
M	total mass
$\phi(\boldsymbol{r}, t)$	MOND potential
$\psi(x, y, t)$	deflection potential
$\boldsymbol{g}_{\rm M}(\boldsymbol{r}, t) = -\boldsymbol{\nabla}\phi$	Milgromian gravity
$\boldsymbol{g}_{\rm N}(\boldsymbol{r}, t)$	Newtonian gravity
$\boldsymbol{F}(\boldsymbol{r}, t) = \boldsymbol{g}_{\rm N} + \boldsymbol{H}$	vector field
$\boldsymbol{H}(\boldsymbol{r}, t)$	magnetic component of $\boldsymbol{F}$
c	speed of light
$a_0 = 1.2\,10^{-10}\mathrm{ms}^{-2}$	Milgrom's constant
G	Newton's constant
Λ	cosmological constant
$a(t)$	cosmological scale factor
$x = g_{\rm M}/a_0$	strength of MOND force
$y = F/a_0$	strength of surface density
$\mu(x) = y/x$	interpolation function
$\nu(y) = x/y$	reciprocal interpolation
$\rho_{\rm B}(\boldsymbol{r}, t)$	baryonic matter density
$\rho_{\rm D}(\boldsymbol{r}, t)$	dark matter density
$\rho_{\rm A}(\boldsymbol{r}, t) = \rho_{\rm B} + \rho_{\rm D}$	apparent matter density
$\sigma(x, y, t)$	surface mass density
Δt	time step
$\mathscr{F}$	3d FT operator
$\widehat{\boldsymbol{f}}(\boldsymbol{k}) = \mathscr{F}(\boldsymbol{f})$	3d FT of $\boldsymbol{f}(\boldsymbol{r})$
$\mathbf{i}$, $\mathbf{j}$, $\mathbf{k}$	standard basis vectors
i, j, k	pixel coordinate index
ξ, η, ζ	random floating-points

Notes. Symbol and significance of the physical quantities used. Vectors and vector fields are in boldface. The length of a vector or the strength of a vector field is in normal font. Most MOND research papers simply use $\boldsymbol{g}$ for the Milgromian gravity field $\boldsymbol{g}_{\rm M}$.

Appendix B: Fourier transforms

In the text, we did not distinguish much between continuous functions on $\mathbb{R}^3$ and the discrete representation on $(\mathbb{Z}/n\mathbb{Z})^3$ for the $n \times n \times n$ grid. We simply considered the discrete functions as an approximation of the continuous functions on a finite interval. Let the physical dimension of the cubic pixels be s, and let the cubic volume have length $L = ns$. We assume that the functions smoothly drop to zero (or a constant) for $|x| > \frac{L}{2}$, $|y| > \frac{L}{2}$, $|z| > \frac{L}{2}$. The position coordinates that correspond to the grid points take on the values

$$\boldsymbol{r} = \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} i \\ j \\ k \end{pmatrix} s, \quad i, j, k \in \left\{-\tfrac{n}{2}, \dots, \tfrac{n}{2} - 1\right\}.$$

Fourier space has the elements $\boldsymbol{k}$ as reciprocal vectors. It has the grid values

$$\boldsymbol{k} = \begin{pmatrix} k_1 \\ k_2 \\ k_3 \end{pmatrix} = \begin{pmatrix} i \\ j \\ k \end{pmatrix} \frac{2\pi}{L}, \quad i, j, k \in \left\{-\tfrac{n}{2}, \dots, \tfrac{n}{2} - 1\right\}.$$

The scalar function $\phi(\boldsymbol{r})$ is expressed in its Fourier transform $\mathscr{F}(\phi) = \widehat{\phi}(\boldsymbol{k})$ via the expansion in plane waves, that is, the inverse Fourier transform

$$\phi(\boldsymbol{r}) = \frac{1}{(2\pi)^3} \iiint \widehat{\phi}(\boldsymbol{k}) \mathrm{e}^{\mathrm{i}\boldsymbol{k}\bullet\boldsymbol{r}} \mathrm{d}k_1 \mathrm{d}k_2 \mathrm{d}k_3 \approx \frac{1}{L^3} \sum_{\boldsymbol{k}} \widehat{\phi}(\boldsymbol{k}) \mathrm{e}^{\mathrm{i}\boldsymbol{k}\bullet\boldsymbol{r}},$$

and the Fourier coefficients, that is, the direct Fourier transform is

$$\widehat{\phi}(\boldsymbol{k}) = \iiint \phi(\boldsymbol{r}) \mathrm{e}^{-\mathrm{i}\boldsymbol{k}\bullet\boldsymbol{r}} \mathrm{d}x\, \mathrm{d}y\, \mathrm{d}z \approx s^3 \sum_{\boldsymbol{r}} \phi(\boldsymbol{r}) \mathrm{e}^{-\mathrm{i}\boldsymbol{k}\bullet\boldsymbol{r}}.$$

In our PM algorithm, we modeled point particles with a smooth density centered around the points $\boldsymbol{r}_i$. The standard CIC method uses the n-fold convolution of the box function $\mathrm{box}(x/s)\mathrm{box}(y/s)\mathrm{box}(z/s)$ for the nth-order method (Hockney & Eastwood 1988). Here, the box function is defined by

$$\mathrm{box}(y) = \begin{cases} 1 & \text{if} \quad |y| < \frac{1}{2}, \\ 0 & \text{else} \end{cases}.$$

For large n, the convolution in one dimension and its Fourier transform may be approximated by

$$\frac{1}{s^n} \underbrace{\mathrm{box}\left(\frac{x}{s}\right) * \cdots * \mathrm{box}\left(\frac{x}{s}\right)}_{n \text{ times}} \approx \frac{1}{s\sqrt{\pi n/6}} \exp\left(\frac{-6x^2}{ns^2}\right), \quad \text{for} \quad n \gg 1,$$

$$\left(\mathrm{sinc}\frac{sk_1}{2}\right)^n \approx \exp\left(\frac{-ns^2k_1^2}{24}\right), \quad \text{for} \quad n \gg 1.$$

We therefore used the following density, with order $n = 12$:

$$\rho(\boldsymbol{r}) = \frac{m_i}{s^3(2\pi)^{3/2}} \exp\left(\frac{-|\boldsymbol{r} - \boldsymbol{r}_i|^2}{2s^2}\right).$$

The Fourier transform of this single-particle density is

$$\widehat{\rho}(\boldsymbol{k}) = m_i \exp\left(\frac{-s^2|\boldsymbol{k}|^2}{2}\right) \mathrm{e}^{-\mathrm{i}\boldsymbol{k}\bullet\boldsymbol{r}_i}.$$

A standard deviation of a single pixel in position space implies a drop off of below $\mathrm{e}^{-\pi^2/2} = .007$ at the boundary of the reciprocal space. Furthermore, the density is very nearly spherically symmetric. The difference between the 12th-order CIC function and the Gaussian with the same variance in three dimensions is 10^{-7} at most.

Appendix C: Flow scheme of the PM code

This is the initialization procedure.

1. Create a lookup table for the exponential function.
2. Create a table of the 251 coordinates in a sphere of radius 4.
3. Create a n^3 matrix with the vector function $\boldsymbol{k}$.
4. Create a n^3 matrix for the inverse Laplace operator,

$$\mathrm{A} = \begin{cases} \dfrac{1}{\boldsymbol{k} \bullet \boldsymbol{k}} & \text{if} \quad \boldsymbol{k} \neq \boldsymbol{0}, \\ 0 & \text{if} \quad \boldsymbol{k} = \boldsymbol{0}. \end{cases}$$

5. Clear a n^3 matrix for the density: $\rho_{\mathrm{B}}(\boldsymbol{r}) = 0$.
6. Loop through the particles $i = 1 \ldots N$.
   - Set the mass m_i, and initialize the position $\boldsymbol{r}_i$ and velocity $\boldsymbol{v}_i$.
   - Next i. End particle loop.

This is the main loop for time steps in the simulation.

1. Loop through all particles $i = 1 \ldots N$ (in case $n^3 > 251N$).
   - Remove particle i from the mass density by filling the pixels with zeros according to the list of 251 pixels.
   - Next i. End particle loop.
2. Loop through all particles $i = 1 \ldots N$.
   - Implement the position step of Leapfrog: $\boldsymbol{r}_i \longrightarrow \boldsymbol{r}_i + \boldsymbol{v}_i \Delta t$.
   - Add particle i to the mass density, with the list of 251 pixels and the lookup table: $\rho_{\mathrm{B}}(\boldsymbol{r}) \longrightarrow \rho_{\mathrm{B}}(\boldsymbol{r}) + \rho_i(\boldsymbol{r})$, using Eq. (14).
   - Next i. End particle loop.
3. FFT the density: $\widehat{\rho}_{\mathrm{B}}(\boldsymbol{k}) = \mathscr{F}(\rho_{\mathrm{B}})$.
4. Calculate $\widehat{\phi}_{\mathrm{N}}(\boldsymbol{k}) = 4\pi G \mathrm{A} \widehat{\rho}_{\mathrm{B}}$.
5. Inverse FFT the potential: $\phi_{\mathrm{N}}(\boldsymbol{r}) = \mathscr{F}^{-1}(\widehat{\phi}_{\mathrm{N}})$.
6. Find Newton gravity $\boldsymbol{g}_{\mathrm{N}}(\boldsymbol{r})$ from ϕ_{N} with finite differences.
7. Initialize the mass flux field by $\boldsymbol{F}(\boldsymbol{r}) = \boldsymbol{g}_{\mathrm{N}}$.
8. Start the iteration loop.
   - Calculate the gravity field: $\boldsymbol{g}_{\mathrm{M}}(\boldsymbol{r}) = \nu(\sqrt{\boldsymbol{F} \bullet \boldsymbol{F}}\,/a_0)\boldsymbol{F}$.
   - FFT the gravity field: $\widehat{\boldsymbol{g}}_{\mathrm{M}}(\boldsymbol{k}) = \mathscr{F}(\boldsymbol{g}_{\mathrm{M}})$.
   - Calculate the potential in Fourier space: $\widehat{\phi}(\boldsymbol{k}) = \mathrm{i}\mathrm{A}\boldsymbol{k} \bullet \widehat{\boldsymbol{g}}_{\mathrm{M}}$.
   - Make the field conservative: $\widehat{\boldsymbol{g}}_{\mathrm{M}}(\boldsymbol{k}) = -\mathrm{i}\boldsymbol{k}\widehat{\phi}$.
   - Exit the iteration loop at the fourth iteration.
   - Inverse FFT the gravity field: $\boldsymbol{g}_{\mathrm{M}}(\boldsymbol{r}) = \mathscr{F}^{-1}(\widehat{\boldsymbol{g}}_{\mathrm{M}})$.
   - Calculate the field: $\boldsymbol{F}(\boldsymbol{r}) = \mu(\sqrt{\boldsymbol{g}_{\mathrm{M}} \bullet \boldsymbol{g}_{\mathrm{M}}}\,/a_0)\boldsymbol{g}_{\mathrm{M}}$.
   - Calculate the magnetic component, $\boldsymbol{H}(\boldsymbol{r}) = \boldsymbol{F} - \boldsymbol{g}_{\mathrm{N}}$.
   - FFT the magnetic field: $\widehat{\boldsymbol{H}}(\boldsymbol{k}) = \mathscr{F}(\boldsymbol{H})$.
   - Make the field divergence free: $\widehat{\boldsymbol{H}}(\boldsymbol{k}) \longrightarrow \widehat{\boldsymbol{H}} - \boldsymbol{k}\mathrm{A}\boldsymbol{k} \bullet \widehat{\boldsymbol{H}}$.
   - Inverse FFT the magnetic field: $\boldsymbol{H}(\boldsymbol{r}) = \mathscr{F}^{-1}(\widehat{\boldsymbol{H}})$.
   - Calculate the field with $\boldsymbol{F}(\boldsymbol{r}) = \boldsymbol{g}_{\mathrm{N}} + \boldsymbol{H}$.
   - Next iteration.
9. Inverse FFT the potential: $\phi(\boldsymbol{r}) = \mathscr{F}^{-1}(\widehat{\phi})$.
10. Loop through all particles $i = 1 \ldots N$.
    - Evaluate $\boldsymbol{g}_i$, with the list of 251 pixels and the lookup table using Eq. (15).
    - Implement the velocity step of Leapfrog: $\boldsymbol{v}_i \longrightarrow \boldsymbol{v}_i + \boldsymbol{g}_i \Delta t$.
    - Next i. End particle loop.
11. Update time $t \longrightarrow t + \Delta t$, and
12. Go to the next time step.