ADVANCES IN NUCLEAR SCIENCE AND APPLICATIONS

Volume 1, No. 4 (2025), P. 285-296

Development of computational methods for constructing collisionless gravitating systems in equilibrium

A. P. Gluchshenko*, D. V. Yurin, Ch. T. Omarov

Fesenkov Astrophysical Institute, Almaty, Kazakhstan

e-mail: gluchshenko@fai.kz

DOI: 10.63907/ansa.v1i4.58
Received: 10 November 2025

Abstract

We present a modification of the original GalIC code, which is designed to generate equilibrium initial conditions for numerical N-body models of collisionless, gravitationally bound systems. The proposed approach focuses on improving the procedure for assigning initial particle velocities, which is crucial for ensuring the stability and physical consistency of equilibrium models. Instead of drawing trial velocities from a uniform distribution bounded by the local escape velocity, we introduce a Gaussian-like distribution with compact support. Its dispersion is adapted to the local properties of the system and is derived from the Jeans equations. This choice provides a more physically motivated sampling of velocity space while preserving the bound nature of the system. A detailed comparison with the original GalIC method is performed using isotropic Hernquist models. We demonstrate that the modified algorithm converges significantly faster and produces well-resolved time-averaged mass profiles and velocity distributions. The resulting models show improved convergence toward the target distribution function while maintaining a comparable computational cost per iteration. Overall, the proposed modification provides a more efficient and physically realistic framework for generating equilibrium N-body initial conditions in collisionless stellar dynamical systems.

1 Introduction

The construction of equilibrium models of stellar systems is a fundamental problem in theoretical astrophysics and in the numerical modeling of gravitationally bound systems. Accurate equilibrium initial conditions are essential for studying the internal dynamics of self-gravitating systems. They are also crucial for reliably predicting their long-term evolution in numerical simulations. Even small deviations from equilibrium may induce artificial relaxation effects, thereby biasing the interpretation of simulation results, particularly in studies of galactic structure and dark matter distributions. In many applications, one seeks to construct an equilibrium system corresponding to a prescribed density profile $\rho(\mathbf{x})$ and a set of specified kinematic properties. A model is said to be self-consistent if its gravitational potential is generated by its own mass distribution through Poisson's equation. In equilibrium, the particle distribution function satisfies a stationary solution of the collisionless Boltzmann equation. From a practical standpoint, it is crucial that such a system evolves only negligibly, or not at all, on timescales comparable to the characteristic crossing time of the system. A variety of methods have been developed to construct equilibrium or near-equilibrium models of differing complexity. Early approaches employed the local Maxwellian approximation for the velocity distribution [1, 2], enabling the construction of axisymmetric halo and bulge models. However, a known limitation of this approach is that a Maxwellian distribution does not always provide an accurate representation of the true distribution function. Alternative techniques based on numerical inversion of the Eddington formula have also been proposed [3, 4], although these are restricted to spherical symmetry. Another widely used class of methods is based on Schwarzschild's orbital superposition technique [5] and its subsequent extensions [6, 7, 8, 9]. These methods are highly flexible and can accommodate a broad range of observational constraints, making them particularly powerful for constructing detailed stellar dynamical models of galaxies. A related approach, better suited for disk systems, has also been developed [10]. Despite their versatility, Schwarzschild-based methods are computationally expensive: they require the construction of large orbit libraries, often containing millions of orbits, followed by the solution of a large-scale optimization or linear programming problem to determine the appropriate orbital weights [16]. An alternative class of methods combines N-body simulations with iterative corrections designed to drive the system toward a desired target state. This category includes the made-to-measure (M2M) technique [11, 12, 13]. Closely related are iterative methods such as those proposed by Rodionov et al. [14] and the GalIC algorithm [15]. In these approaches, the full dynamical evolution of the system is not explicitly followed; instead, particle velocities are iteratively adjusted so as to reproduce prescribed constraints. The procedure is repeated until convergence is achieved according to a chosen criterion. Compared to Schwarzschild's method, these techniques are significantly less computationally demanding and scale efficiently with particle number. This advantage, however, comes with certain limitations. In particular, GalIC is generally restricted to axisymmetric configurations and is not well suited for systems dominated by large central masses [17, 18]. Nevertheless, it remains a popular tool for constructing equilibrium models of axisymmetric systems, including disk galaxies [19] and globular clusters [20].

In this work, we propose a modification of the GalIC method aimed at improving the assignment of initial particle velocities. In its original formulation, the algorithm discretizes space using a computational grid and iteratively assigns velocities to randomly selected particles, guided by a predefined objective function. Trial velocity components are drawn from a uniform distribution within the interval $v_i \in [-v_{\rm esc}, v_{\rm esc}]$, where v_i denotes a velocity component and $v_{\rm esc}$ is the local escape velocity. This procedure is repeated until the objective function reaches a minimum. We introduce a modification in which trial velocities are drawn from a Gaussian-like distribution, with a dispersion determined from the Jeans equations. This choice provides a more physically motivated sampling of velocity space, yielding initial conditions that are closer to the true equilibrium distribution while substantially improving the convergence rate of the algorithm. A detailed description of the method is presented in Section 2. In Section 3, we compare the performance of the modified algorithm with the original implementation and demonstrate that the proposed approach achieves faster convergence and improved accuracy of the resulting equilibrium models.

2 Algorithm description

In this work, we implement an algorithm that generally follows the approach described in [15]. The primary goal of the algorithm is to minimize an objective function defined on a discrete spatial grid. This function measures the deviation between the target mass distribution and the time-averaged mass distribution generated by particle orbits.

The objective function is defined as

$$S(\boldsymbol{v}_1, \dots, \boldsymbol{v}_N) = \sum_j \left| \overline{M}_j(\boldsymbol{v}_1, \dots, \boldsymbol{v}_N) - M_j^0 \right|, \tag{1}$$

where M_j^0 and $\overline{M}_j(\boldsymbol{v}_1, \dots, \boldsymbol{v}_N)$ denote the target and time-averaged masses contained in the j-th grid cell, respectively. Thus, the objective function represents the total deviation between the prescribed and realized mass distributions over the entire grid.

The time-averaged mass in the j-th cell is defined as

$$\overline{M}_j(\boldsymbol{v}_1, \dots, \boldsymbol{v}_N) = \sum_{i=1}^{N} m_p \, \tau_j^{\rm orbit}(\boldsymbol{r}_i, \boldsymbol{v}_i), \tag{2}$$

where m_p is the particle mass, and $\tau_j^{\rm orbit}(\boldsymbol{r}_i, \boldsymbol{v}_i)$ is the fraction of time that a particle with initial position $\boldsymbol{r}_i$ and velocity $\boldsymbol{v}_i$ spends in the j-th cell during its orbit integration.

For a given mass density distribution $\rho(\boldsymbol{x})$, the configuration space is divided into an adaptive grid. The resolution of the grid is determined by imposing an approximately constant mass in each cell,

$$M_j^0 = \mu M, \tag{3}$$

where M is the total mass of the system and $\mu < 1$ specifies the fraction of the total mass assigned to each cell. As a result, all grid cells contain approximately the same amount of mass. If all particles have equal masses, such adaptive partitioning minimizes shot noise arising from unequal particle numbers in different cells, which is essential for a stable evaluation of the objective function.

The algorithm allows for flexibility in the choice of grid geometry. Depending on the application, one may employ logarithmic grids, as in [15], or hierarchical octree partitioning similar to that used in the Barnes–Hut algorithm [21]. The iterative procedure then proceeds as follows:

1. A particle is randomly selected from the system.

2. A trial velocity is assigned to this particle, with each component drawn from the interval $v_i \in [-v_{\mathrm{esc}}, v_{\mathrm{esc}}]$, where v_{esc} is the local escape velocity.

3. The particle's orbit is integrated using the trial velocity, its time-averaged contribution to each grid cell is computed, and the objective function (1) is evaluated.

4. If the value of the objective function is reduced, the trial velocity is accepted; otherwise, the previous velocity is retained. The algorithm then returns to step 1.

This iterative procedure is repeated until the objective function reaches a minimum, that is, until further iterations of velocity selection no longer lead to its reduction. To enable efficient parallelization, the algorithm can be applied not to a single particle at a time, but to a randomly selected group of particles, each processed independently. This strategy serves two purposes: it allows for parallel computation and reduces the risk of overfitting individual particle velocities.

Since particles within a group are updated independently, the newly assigned velocities and the value of the objective function are not immediately synchronized. As a result, some velocity updates that locally improve the objective function may not contribute to a global decrease once all updates are applied. This behavior is analogous to that of simulated annealing algorithms [22] and is also commonly encountered in the training of deep neural networks [23].

Although this approach is effective, there is room for further improvement. In the original formulation, trial velocities are drawn from a uniform distribution over the allowed range of velocity components. While this choice guarantees uniform exploration of the accessible phase space for each particle, it also strongly influences both the convergence rate and the final equilibrium state. Modifying the sampling distribution therefore provides a means to guide the algorithm toward faster convergence.

Instead of drawing trial velocities from a uniform distribution, one may employ a distribution with a predefined dispersion. A natural choice is a normal (Gaussian) distribution, as used in earlier studies [1, 2], where the velocity dispersions are obtained from the Jeans equations. However, maintaining the boundedness of the system poses a difficulty in this case: the Gaussian distribution has infinitely extended tails, implying a nonzero probability of generating velocities outside the physically allowed interval $v_i \in [-v_{\mathrm{esc}}, v_{\mathrm{esc}}]$. Although the distribution can be artificially truncated at these boundaries, such truncation introduces discontinuities and compromises smoothness.

To overcome this limitation, we propose using a velocity distribution function with compact support, explicitly defined over the allowed domain. For example, in the isotropic case with $v \in [0, 1]$, the velocity distribution function is chosen as

$$f(v) = 4\pi C\, v^2 \exp\left(-\frac{v^2}{2\delta^2}\right)(1 - v^2), \tag{4}$$

where v is the velocity magnitude, δ is a free parameter, and C is a normalization constant determined from the condition

$$\int_0^1 f(v)\,\mathrm{d}v = 1. \tag{5}$$

The parameter δ is determined locally from the dispersion-matching condition and therefore varies with radius, rather than being a fixed global parameter. For a purely Gaussian distribution, the parameter δ corresponds directly to the velocity dispersion. In the present case, however, the presence of the compact-support factor $(1 - v^2)$ implies that δ no longer coincides with the actual dispersion of the resulting distribution. Moreover, the discrepancy between δ and the true dispersion increases with increasing dispersion.

The guiding principle of our algorithm is to generate trial velocities with a prescribed, physically motivated dispersion. To achieve this, we precompute the dispersion of the distribution (4) as a function of δ, denoted by $\sigma = \sigma(\delta)$. Given a desired dispersion σ, the corresponding value of δ is obtained via interpolation, ensuring that the sampled velocity distribution has the required second moment.

Finally, we emphasize that the proposed modification introduces a negligible computational overhead. Since the function $\sigma(\delta)$ is precomputed and stored in memory, the additional cost is limited to a simple interpolation step, which does not significantly affect the runtime per iteration.

3 Results

We performed a series of numerical tests to assess the performance of the proposed approach. As a benchmark model, we adopted the isotropic Hernquist profile [24], whose density distribution is given by

$$\rho(r) = \frac{M}{2\pi}\,\frac{a}{r}\,\frac{1}{(r + a)^3}, \tag{6}$$

where M is the total mass of the system and a is the scale radius.

There is no particular or exclusive reason for choosing the Hernquist profile for this study. In fact, a wide range of alternative models would be equally suitable, including other members of the Dehnen family [25], the Plummer model [26], and King models [27]. The only assumptions required for the present analysis are spherical symmetry and velocity isotropy.

The Hernquist model is especially convenient for testing purposes because, in the isotropic case, the density profile, gravitational potential, and distribution function all admit closed-form analytical expressions. This considerably simplifies orbit integration and allows for direct comparison between numerical and analytical results. Moreover, the spherical symmetry of the model permits the use of a one-dimensional spatial grid constructed from concentric spherical shells. The grid is defined using the cumulative mass profile, such that each radial shell contains approximately the same amount of mass.

To assess the effectiveness of the proposed approach, we performed a comparative analysis of the results obtained using the modified trial velocity generation scheme and the original method. The comparison was carried out for different particle numbers ($N = 128\,000$ and $512\,000$) and different grid resolutions (1024 and 2048 cells).

As evaluation criteria, we considered both the value of the objective function and the deviation of the resulting mass and velocity distributions from the corresponding analytical solutions derived for the Hernquist model [24].

For convenience, we adopted a system of normalized units in which the gravitational constant, the total mass of the system, and the scale radius are set to unity,

$$G = M = a = 1. \tag{7}$$

In all figures, the models are labeled using the following convention: number of particles – number of grid cells – velocity sampling method. For example, the label "128K–1024–gausslike" denotes a model constructed using $128\,000$ particles, a grid consisting of 1024 cells, and the proposed Gaussian-like velocity sampling scheme.

Exploiting the spherical symmetry of the Hernquist model, we constructed the spatial grid from concentric spherical shells, each containing approximately the same mass,

$$M_j^0 = \mu M, \tag{8}$$

where $\mu = 1/1024$ and $\mu = 1/2048$ for grids with 1024 and 2048 cells, respectively. The radial boundaries of the shells were determined from the cumulative mass profile of the Hernquist model [24].

All simulations were performed in parallel mode using blocks of 1024 independently processed particles. The calculations were carried out on a system equipped with two Intel Xeon Platinum 8362 processors, providing a total of 64 physical cores and 128 hardware threads.

The oscillations observed in Figure 1 arise once the algorithm reaches a local minimum of the objective function, beyond which further adjustments of particle velocities no longer lead to a systematic decrease. Because the method employs a parallel update scheme, the velocities of many particles are modified simultaneously and independently (see Section 2). Consequently, velocity updates that locally improve the objective function for individual particles do not necessarily result in a global reduction after all updates are applied and the objective function is recomputed. This collective effect leads to small fluctuations of the objective function around the local minimum, which manifest as the oscillatory behavior seen in Figure 1.

The results clearly demonstrate that the objective function converges significantly faster when the proposed velocity sampling scheme is employed. This trend is consistently observed across all tested models. Defining one iteration as the step in which, on average, each particle is processed once by the algorithm (i.e., the total number of particles divided by the block size), we find that each particle undergoes approximately 10 velocity updates when using the proposed method, compared to about 100 updates in the original implementation. This reduction directly translates into a substantial decrease in the total computational time.

Figure 2 illustrates the dependence of the CPU time on the number of iterations for all tested models. Although the computational time increases approximately linearly with the number of iterations in logarithmic scale, the proposed velocity

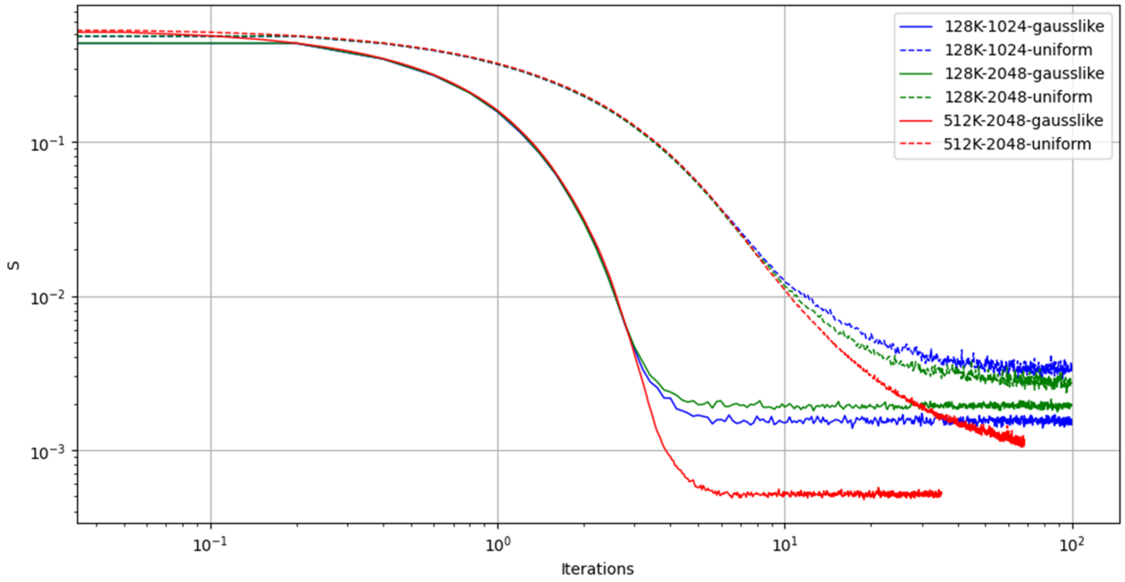

Figure 1: Evolution of the objective function during the iterative procedure for different particle numbers and grid resolutions. Solid lines correspond to velocity sampling using the proposed Gaussian-like distribution, while dashed lines indicate sampling from a uniform distribution.

sampling scheme does not introduce any noticeable computational overhead, and the per-iteration cost remains comparable to that of the original method.

Because the Gaussian-like sampling reaches convergence after approximately 10 iterations, whereas the uniform velocity sampling typically requires around 100 iterations, the proposed method achieves an almost order-of-magnitude reduction in the total computational time. This behavior is consistently observed for all models, independent of the particle number and grid resolution.

In addition to the global convergence of the objective function, we analyze the local deviations between the target and time-averaged mass distributions. Figure 3 shows the relative difference between the expected mass M_j^0 and the time-averaged mass $\overline{M}_j$ in each grid cell as a function of radius. This representation provides a more detailed view of how accurately the target mass profile is reproduced across the system.

Figure 3 shows that the original velocity sampling method exhibits significantly larger deviations toward the inner and outer edges of the grid, which manifest as pronounced ”tails” in the distribution of mass residuals. In contrast, the use of the proposed Gaussian-like velocity distribution largely suppresses these extreme deviations, although weak oscillatory patterns remain visible. Overall, the modified velocity selection scheme provides a more uniform accuracy across all grid cells and yields a better reproduction of both the central and outer regions compared to the original uniform sampling approach.

A comparison between the numerically obtained velocity dispersion profiles and the analytical solution for the isotropic Hernquist model shows good agreement in the vicinity of the scale radius, $r \simeq a$ (Figure 4). Deviations from the analytical profile become noticeable only in the innermost region, $r \ll a$, where numerical effects are expected to be most pronounced.

We note that the behavior of the algorithm in the innermost region is also relevant

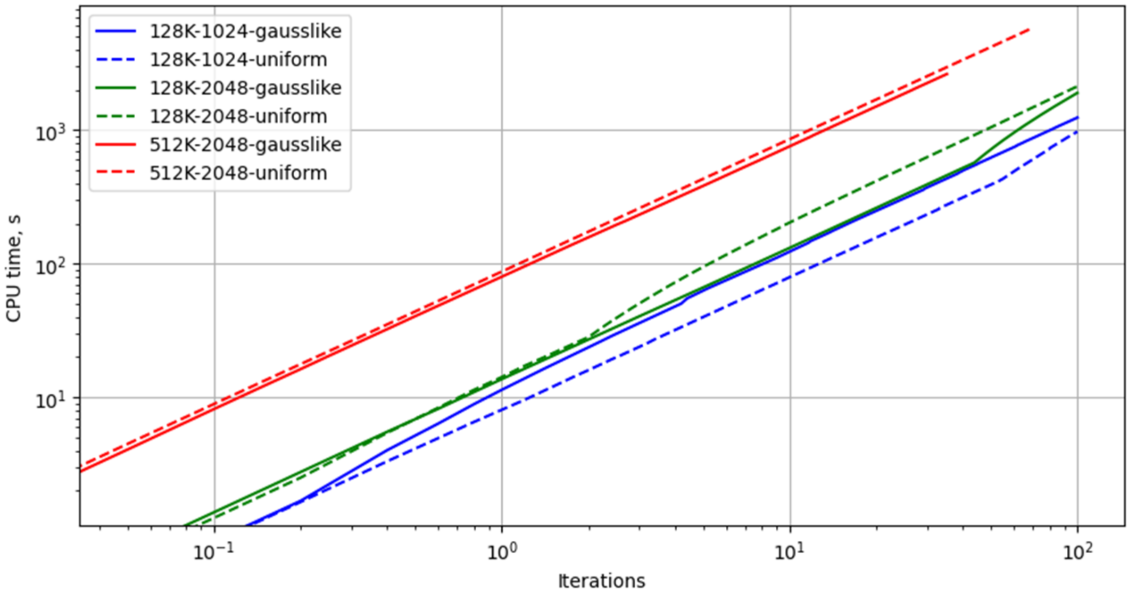

Figure 2: CPU time versus iteration number. Solid and dashed lines indicate Gaussian-like and uniform velocity sampling, respectively. Faster convergence leads to a significant overall speed-up.

in the context of the cusp problem. In systems with steep central density cusps, accurately reproducing the inner mass and velocity structure poses a well-known challenge for equilibrium modeling techniques. Although the present study focuses on isotropic Hernquist models, which possess a fixed central cusp, the improved convergence properties of the proposed velocity sampling scheme suggest that it may be better suited for applications involving cuspy density profiles. A detailed investigation of this issue will be addressed in future work.

The method based on uniform velocity sampling systematically produces slightly overestimated velocity dispersions relative to the analytical solution, whereas the proposed Gaussian-like sampling yields mildly underestimated dispersions. This behavior is observed despite the fact that, in the latter case, trial velocities are drawn using a predefined dispersion derived from the Jeans equations. The discrepancy suggests that, for the adopted parameters, the algorithm preferentially reduces particle velocities in the central region rather than redistributing particles to larger radii.

The overall accuracy of the two approaches is nevertheless comparable. The mean relative deviations of the dispersion profiles from the analytical Hernquist solution are $\varepsilon \simeq 1.2\%$ for the Gaussian-like sampling and $\varepsilon \simeq 1.3\%$ for the uniform sampling.

A direct comparison of the velocity distribution functions further illustrates the improved performance of the proposed velocity sampling scheme. Figure 5 presents the distributions of velocity magnitude (top row) and of the radial and tangential velocity components (middle and bottom rows, respectively) at several representative radii spanning more than four orders of magnitude in r.

Across most radii, the distributions obtained using the Gaussian-like sampling closely follow the analytical Hernquist solution, reproducing both the overall shape and the width of the distributions. In contrast, velocity sampling from a uniform

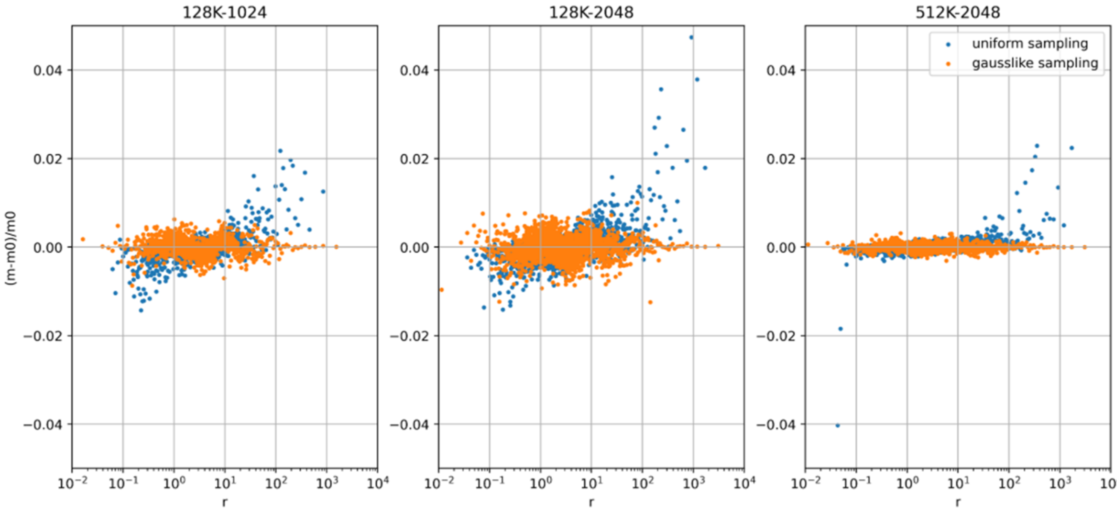

Figure 3: Relative deviation of the time-averaged mass from the target mass in each grid cell as a function of radius for different particle numbers and grid resolutions. Blue symbols correspond to uniform velocity sampling, while orange symbols indicate the proposed Gaussian-like sampling scheme.

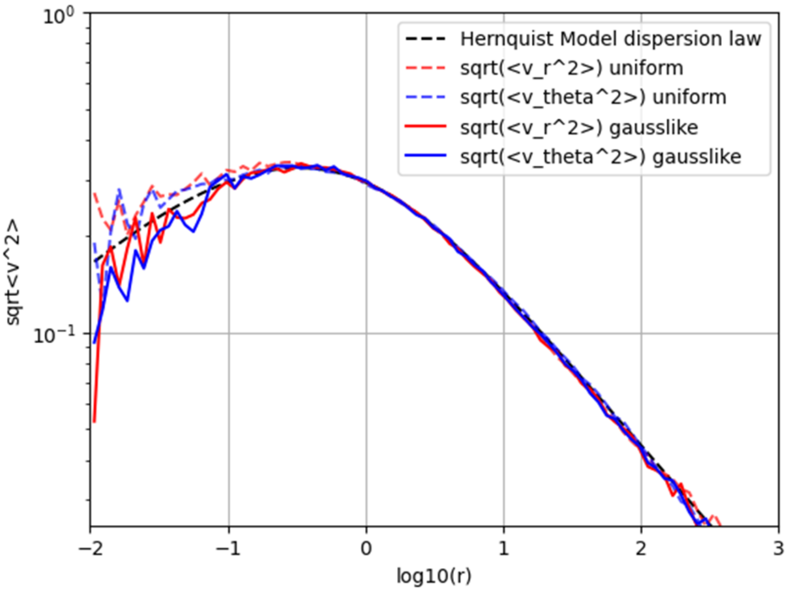

Figure 4: Comparison of the radial and tangential velocity dispersion profiles for the 512K–2048 model. Dashed lines correspond to the original uniform velocity sampling scheme, while solid lines indicate the proposed Gaussian-like sampling. The black dashed curve shows the analytical solution for the isotropic Hernquist model.

distribution leads to noticeable distortions, particularly in the high-velocity tails and in the central regions. These deviations become increasingly pronounced at large radii, where the uniform sampling produces broader and less regular distributions.

The agreement between the proposed method and the analytical solution is especially good for the radial and tangential velocity components, which remain nearly Gaussian and symmetric over a wide range of radii. At the largest radii shown, statistical noise becomes more significant due to the smaller number of particles, yet the Gaussian-like sampling continues to provide a closer match to the theoretical distributions than the uniform approach.

Overall, this comparison demonstrates that sampling velocities from a distribution with a prescribed, physically motivated dispersion not only accelerates convergence but also yields velocity distributions that are consistent with the underlying equilibrium distribution function across the entire system.

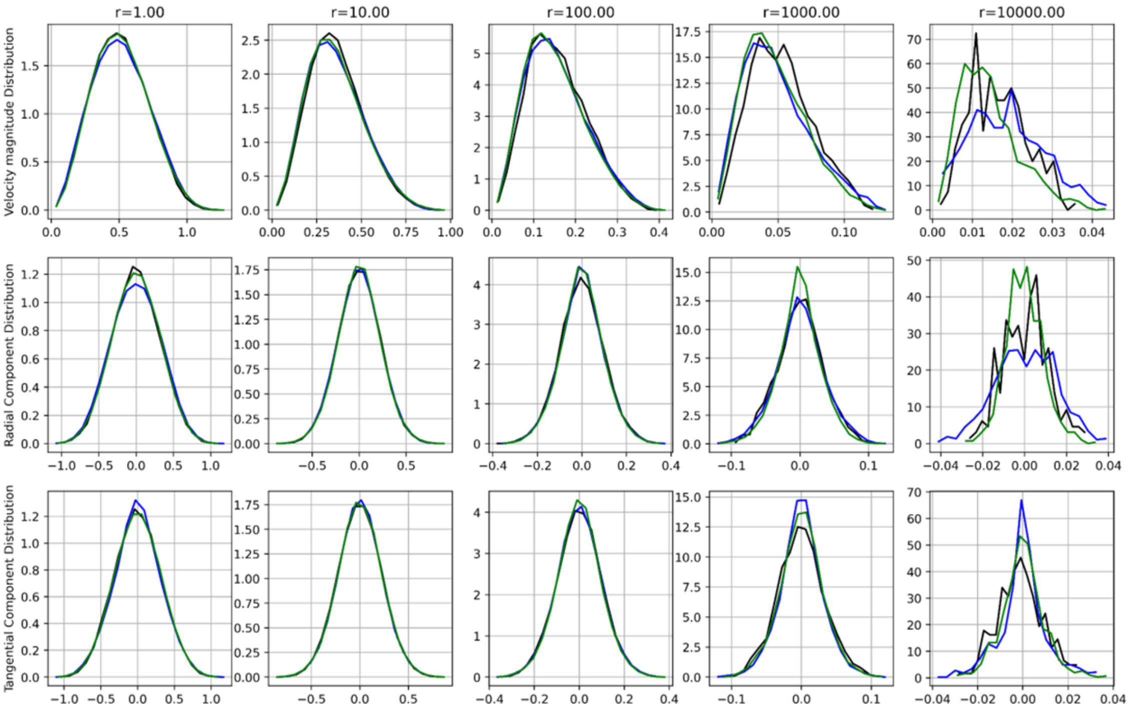

Figure 5: Comparison of velocity distribution functions at different radii. The first row shows distributions of velocity magnitude, while the second and third rows display the radial and tangential velocity components, respectively. The black curves correspond to the analytical Hernquist model, the blue curves to uniform velocity sampling, and the green curves to the proposed Gaussian-like sampling scheme.

4 Conclusion

In this work, we have presented an improved method for assigning initial stellar velocities within the framework of the iterative GalIC algorithm. In contrast to the original approach, which relies on uniform sampling of velocity components within the allowed range, the proposed method employs a Gaussian-like distribution with a dispersion derived from the Jeans equations. This choice provides a more physically motivated initialization of particle velocities.

A comparative analysis demonstrates that the proposed modification significantly accelerates the convergence of the algorithm while improving the accuracy of the resulting equilibrium configurations. In particular, the convergence of the objective function is enhanced both in terms of rate and final value, and the reconstructed mass profiles exhibit a more uniform accuracy across the system. At the same time, the velocity dispersion profiles and the velocity distribution functions produced by the new method show errors that are comparable to those obtained with the original uniform sampling scheme.

This behavior can be attributed to the definition of the objective function, which constrains only the mass distribution through the difference between the target and time-averaged cell masses. As a result, velocity moments are optimized only indirectly. This limitation could be alleviated by including dispersion-dependent terms in the objective function, as suggested by Yurin & Springel [15], without compromising the observed improvement in convergence efficiency.

Overall, the proposed approach achieves comparable accuracy in velocity distributions and their moments while substantially reducing the total computational cost and improving the fidelity of the reconstructed mass profiles. The method therefore provides a more efficient and physically consistent tool for constructing self-consistent equilibrium models of stellar systems. Future work will focus on the further development of the algorithm, including its extension to systems with anisotropic velocity distributions.

Acknowledgments

This work was carried out within the framework of Program No. BR24992807, “Development of the Kazakhstani digital environment for astronomical research of near and deep space objects within the framework of the international network of virtual observatories”, funded by the Ministry of Science and Higher Education of the Republic of Kazakhstan (MSHE RK), as well as within the framework of Grant No. AP23487846, also funded by the MSHE RK.

References

[1] L. Hernquist, Astrophys. J. Suppl. Ser. **86**, 389 (1993).

[2] E. Athanassoula, *Galactic Bulges* (Springer International Publishing, Cham, 2016), p. 391.

[3] L. M. Widrow, B. Pym, and J. Dubinski, Astrophys. J. **679**, 1239 (2008).

[4] A. C. Robin *et al.*, Astron. Astrophys. **409**, 523 (2003).

[5] M. Schwarzschild, Astrophys. J. **232**, 236 (1979).

[6] N. Cretton *et al.*, Astrophys. J. Suppl. Ser. **124**, 383 (1999).

[7] K. Gebhardt *et al.*, Astron. J. **119**, 1157 (2000).

[8] M. Valluri, D. Merritt, and E. Emsellem, Astrophys. J. **602**, 66 (2004).

[9] R. C. E. van den Bosch *et al.*, Mon. Not. R. Astron. Soc. **385**, 647 (2008).

[10] E. Vasiliev and E. Athanassoula, Mon. Not. R. Astron. Soc. **450**, 2842 (2015).

[11] D. Syer and S. Tremaine, Mon. Not. R. Astron. Soc. **282**, 223 (1996).

[12] N. Bissantz, V. P. Debattista, and O. Gerhard, Astrophys. J. **601**, L155 (2004).

[13] F. De Lorenzi *et al.*, Mon. Not. R. Astron. Soc. **385**, 1729 (2008).

[14] S. A. Rodionov, E. Athanassoula, and N. Ya. Sotnikova, Mon. Not. R. Astron. Soc. **392**, 904 (2009).

[15] D. Yurin and V. Springel, Mon. Not. R. Astron. Soc. **444**, 62 (2014).

[16] E. Vasiliev and M. Valluri, Astrophys. J. **889**, 39 (2020).

[17] T. K. Waters *et al.*, Astrophys. J. **971**, 149 (2024).

[18] D. D. Nguyen *et al.*, Astron. Astrophys. **698**, L9 (2025).

[19] E. Jiménez *et al.*, Mon. Not. R. Astron. Soc. **524**, 4346 (2023).

[20] M. Ishchenko *et al.*, Astron. Astrophys. **689**, A178 (2024).

[21] J. Barnes and P. Hut, Nature **324**, 446 (1986).

[22] M. Vargas-Martínez *et al.*, Math. Comput. Appl. **28**, 38 (2023).

[23] S. Zhang *et al.*, Neural Netw. **183**, 106931 (2025).

[24] L. Hernquist, Astrophys. J. **356**, 359 (1990).

[25] W. Dehnen, Mon. Not. R. Astron. Soc. **265**, 250 (1993).

[26] H. C. Plummer, Mon. Not. R. Astron. Soc. **71**, 460 (1911).

[27] I. R. King, Astron. J. **71**, 64 (1966).