Mon. Not. R. Astron. Soc. **000**, 1–2 (2014) Printed 5 March 2015 (MN LATEX style file v2.2)

Vlasolve: numerical method

Sousbie, Thierry[1]

[1] *Institut d'Astrophysique de Paris, CNRS UMR 7095 and UPMC, 98bis, bd Arago, F-75014 Paris, France*

5 March 2015

ABSTRACT

Key words: gravitation – methods: numerical – galaxies: kinematics and dynamics – dark matter

The evolution of the density distribution in a spherically symmetric dark matter halo is described by the Vlasov-poisson equation:

$$\frac{\partial f}{\partial t} + u\frac{\partial f}{\partial r} + \left(\frac{j^2}{r^3} - \frac{GM_r}{r^2}\right)\frac{\partial f}{\partial u}, \tag{1}$$

with $f = f(r, u, j, t)$ the phase-space distribution function at time t, distance to center r, radial velocity u and angular momentum j, and $M_r = M(r' < r)$ the mass inside a sphere of radius r.

In order to solve this equation numerically, we use a semi-Lagrangian method similar to Fujiwara (1983). That is, phase space (r, u, j) is discretized into a rectangular mesh that spans the domain of interest and the value of the initial distribution function is sampled at each mesh site i with coordinates (r_i, u_i, j_i). Considering each mesh site as a particle, the value of $f(r_i, u_i, j_i, t + dt)$ at site i and time $t + dt$ can then be easily deduced by integrating the particle trajectory backward in time at time t and interpolating the distribution function at that location.

For each time step, the integration of the particle trajectory can be carried out using the fact that equation (1) can be split à la Cheng & Knorr (1976) into a free-streaming part:

$$\frac{\partial f}{\partial t} + u\frac{\partial f}{\partial r} + \frac{j^2}{r^3}\frac{\partial f}{\partial u}, \tag{2}$$

and an acceleration part:

$$\frac{\partial f}{\partial t} - \frac{GM_r}{r^2}\frac{\partial f}{\partial u}. \tag{3}$$

Indeed, combining equations (2) and (3) with a leap-frog type numerical integrator, the value of $f(r_i, u_i, j_i, t + dt)$ for each grid site can be deduced at a second order accuracy level from its value one time-step earlier, at time t, through a sequence of simple shifts:

$$f^*(r_i, u_i, j_i) = f(\tilde{r}_i, \tilde{u}_i, j_i, t,), \tag{4}$$

$$f^{**}(r_i, u_i, j_i) = f^*\left(r, u - GM_r/r^2 dt, j\right), \tag{5}$$

$$f(r_i, u_i, j_i, t + dt) = f^{**}(\tilde{r}_i, \tilde{u}_i, j_i), \tag{6}$$

where $(\tilde{r}_i, \tilde{u}_i) = (r_i(t - dt/2), u_i(t - dt/2))$ is the position of grid site i free streamed half a time step backward in time using equation (2), and M_r is computed from $f^*(r_i, u_i, j_i)$. This integration scheme requires two interpolations per time step: a 2D* one in the (r, u) sub-space for the free streaming part† (4) and a 1D interpolation for the acceleration part (5). Similarly to Fujiwara (1983), these interpolations are carried out using third order spline interpolation.

With such a technique, it is very important to carefully define the domain of integration so that it spans the smallest possible phase-space volume (i.e. it needs only contain any region where the distribution function will be non-null at any point in time), while maintaining an adequate resolution everywhere (i.e. phase space distribution features should keep an approximately constant resolution in terms of pixel independently of their position or orientation in phase space). Following Fujiwara (1983), the computational domain is therefore defined as (r, u, j) such that $R_{\min} \leqslant r\_\max$, $|u| \leqslant u_{\max}$ and $0 \leqslant j \leqslant J_{\max}$ and the grid site have a linear distribution along u, a logarithmic distribution along j and the positions of the k^{th} angular momentum site is $j_k = k^2\left(J_{\max}/{N_k}^2\right)$.

Using a logarithmic scale along the radial direction proves particularly advantageous as it allows resolving tiny features close to the center that will expand to larger scales later on as they get further away. One downside of it however is that a small sphere of radius $R_{\min}$ will necessarily be missing from the domain. This shortcoming is usually dealt with by considering the central part as a small reflecting spheres (see e.g. Gott (1973), Fujiwara (1983)). Although simple, this method presents the major drawback of introducing a systematic shift in phase space

* note that due to the spherical coordinates system, $\tilde{u}_i \neq u_i$.
† equations (6) and (4) of the following time-step can indeed be combined into a single interpolation.

distribution: particles reaching the reflective kernel boundary instantly travel the length L through the part of their trajectory contained in the central region below $R_{\rm min}$. As a consequence, a particle i reaching the kernel with velocity u_i will systematically be spuriously shifted in time by $\delta t_u \approx u_i/L_i$ while another particle j with velocity $v_j \neq u_i$ will systematically be shifted in time by $\delta t_v \approx v_j/L_j \neq \delta t_u$, introducing a systematic lag between orbits.

We remedy this issue by taking into account the time spent inside the central sphere for each particle reaching it. We first note that the only quantity that depends on time in equations (4) through (6) is $M_r = M\left(r' \leqslant r\right)$, the mass below r, a strictly increasing function of the radius that tends to 0 as r decreases. For $r \leqslant R_{\rm min}$ sufficiently small, the influence of the acceleration part (3) of the split Vlasov equation over a particle with coordinates $(r_{\leqslant R_m in}, u)$ therefore becomes negligible compared to that of its free-streaming counterpart (2) and we can safely neglect the time dependence. As a consequence, under the assumption that $R_{\rm min}$ is sufficiently small, the trajectory followed by each grid site particle that penetrates into the central sphere does not depend on time, allowing us to pre-compute them for each grid site whose value at previous time step should be interpolated inside the central sphere.

From a technical point of view, this delayed central sphere method is implemented by associating a linked list to each grid site whose coordinates $(\tilde{r}_i, \tilde{u}_i)$ half a time step backward in time is such that $\tilde{r}_i \leqslant R_{\rm min}$. Each linked list contains as many elements as the number of time steps necessary for the corresponding particle to travel throughout the central region and the n^{th} element in the list stores the coordinates of its grid site n time steps backward in time. Before starting the simulation, we initialize each element coordinate and corresponding value of the initial distribution function. For each time step, the value of each element is then simply updated by assigning to it the value of its successor while the last element value, whose coordinates fall outside the central region, is interpolated. A comparison of the results obtained with the reflective central sphere and our improved delayed central sphere is shown on figure .

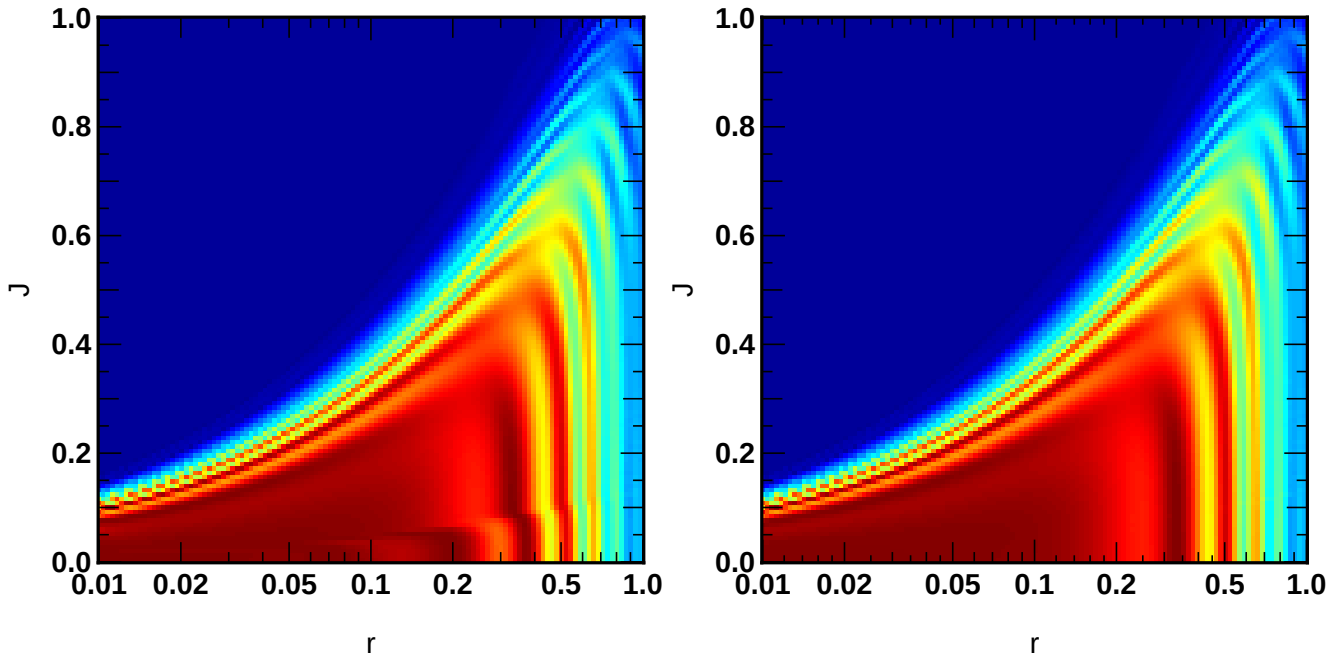
We conclude this section with a word on parallelization. For the purpose of this work, we implemented a hybrid shared and distributed memory version of the algorithm described above via the OpenMP and MPI libraries respectively. Shared memory parallelism is relatively straightforward to implement in the spherically symmetric case, taking advantage of the fact that the angular momentum j is a conserved quantity. Indeed, angular momentum conservation implies that that the spline interpolations required for the numerical integration steps (4) through (6) can be computed independently for each grid slice of constant j. We therefore easily reach an almost perfect‡ parallelization up to a number of tasks equal to the grid resolution N_j the dimension j, which is typically larger than the number of available cores present on a shared memory systems.

Distributed memory parallelization via MPI is not as trivial though as in that case the total number of available computational cores may become very large while spline interpolations are intrinsically non-local, making it non trivial to split computations along dimensions other than j. Sticking with the trivial parallelization described above, the maximum total number of processes running in parallel is therefore limited to N_j, which may become an issue for simulations with high resolution in position and velocity but comparatively low resolution in angular momentum. We overcome this limitation by following Crouseilles & al. (2009) who propose to localize cubic splines interpolation over large enough patches that may not cover the full integration domain along one or more dimensions.

REFERENCES

Cheng C. Z., Knorr G., 1976, JCoPh, 22, 330
Crouseilles N., Latu G., Sonnendrücker, E., 2009, JCoPh, 228, 1429
Fujiwara T., 1983, PASJ, 35, 547
Gott, J.R. III , 1973, ApJ, 186, 481

‡ spline interpolation is by far the most time consuming computation in a time-step



(a) Reflective central sphere

(b) Delayed central sphere

Figure 1. Comparison of the simulated evolution of a spherical top hat profile at $t = 30$ in the $(r, u = 0, j)$ plane, for a reflective central sphere (left) and our improved delayed central sphere (right). The systematic artificial speed increase undergone by orbits that penetrate the central region compared to their higher angular momentum counterparts can clearly be observed at low j on the left picture where a reflective sphere is used, while the distribution function does not exhibit such spurious feature when a delayed kernel is used (right).