DOI: 10.14529/jsfi240302

Study of the Effectiveness of Parallel Algorithms for Modeling the Dynamics of Collisionless Galactic Systems on GPUs

Sergei S. Khrapov[1], *Alexander V. Khoperskov*[1]

N-body model is a common research tool in galaxy physics and cosmology. The transition to the use of computing systems with GPUs can significantly improve the performance and quality of simulation results for gravitational systems. N-body – Particle-Particle algorithm is presented on a hybrid computing platform CPU + multi-GPUs. Using a direct method of calculating gravitational forces by summing the interactions of each particle with each other is resource-intensive, but provides the best accuracy in modeling dynamics at all scales. The main result is an analysis of the efficiency of parallel code depending on the number of GPUs and the choice of single and double precision floating-point arithmetics. The laws of conservation of energy, momentum and angular momentum are tested for a series of models, including major mergers of galaxies and the evolution of galactic stellar disc subject to the most severe gravitational instability. The general conclusion is that conservation laws are poorly implemented when using 4-byte numbers due to the accumulation of arithmetic errors. Calculations with 8-byte numbers ensure that the laws of conservation of momentum and angular momentum are satisfied to the limit of arithmetic accuracy without accumulating errors. The law of conservation of energy is determined primarily by the order of the numerical scheme for integrating the equations of motion. The additional reduction in the error of the conservation law of total energy due to the transition from 4-byte to 8-byte numbers is 1–2 orders of magnitude. Increasing the number of GPUs used helps improve the implementation of conservation laws due to a decrease in the number of particles per graphics processing unit.

Keywords: N-body, GPUs, OpenMP-CUDA, GPUDirect, efficiency.

Introduction

Models of the dynamics of interacting particles are used to describe a wide variety of physical systems and processes from chemistry and plasma physics [2, 19, 20] to astrophysical objects [6, 10, 13, 14, 16, 23]. The properties of the physical medium are determined by the type of field interaction between particles. Improving the quality of modeling, the dynamics of a system requires an increase in the number of particles N, which is limited by computing resources. The N-body model belongs to a class of molecular dynamics models based on the motion simulations of a large number of interacting particles [9, 29]. This approach is effective for studying the dynamics of rarefied gases [7], large atomistic clusters [11], biomolecules and biological systems [2, 9]. Molecular and atomistic models use short-range potentials such as van de Waals' force or Debye screening of charges in a quasi-neutral medium [11]. The gravitational potential is always long-range and the most distant parts in a gravitationally bound system can make a significant contribution to the force [1, 5, 14, 21].

Molecular Dynamics Simulation (or N-body) problems are often critically dependent on the number of particles, so the new computing advantages of GPUs significantly improve modeling efficiency due to price-performance ratio [2, 6, 14]. An important excellence of computing systems with GPUs is the ability to perform massive series of simulations to build large datasets, using the appropriate subsystems of supercomputers [26].

The number of objects $N_\star$ (stars, gas clouds) in real gravitating astrophysical systems, as a rule, significantly exceeds the number of model particles N, which leads to the problem of

[1] Volgograd State University, Volgograd, Russian Federation

ensuring collisionlessness in the numerical model. For example, a typical S-galaxy with a number of stars of the order of $N_\star \sim 10^{11}$ is a collisionless system in which the role of pair interactions is negligible compared to the influence of the mean field. Modeling with a particle number of $N_\star/N \gg 1$ implies the use of macroparticles with a mass $N_\star/N$ times greater than the mass of an average star. The collisionlessness of the gravitating system is ensured by using softening radius r_c at small distances to avoid pair interactions. The choice of the optimal smoothing parameter r_c depends on the number of particles, system configuration and other factors [18, 25]. The characteristic relaxation time in the stellar components of galaxies is $\propto N/\ln(N)$, which preserves collisionless at cosmological times [1, 5].

The traditional approach for calculating the gravitational force from N particles in an astrophysical system is based on various approximate methods, including the fast Fourier transform, various versions of TreeCode, wavelet transforms, etc. [15, 24, 27]. The use of approximate methods for calculating the gravitational potential is dictated by the desire to have as many particles N as possible, which, however, is accompanied by an increase in the error of the interaction force.

The duration of the studied evolution of galactic systems can be long and often reaches $t^{(\max)} \sim 10$ billion years [16]. Moreover, the integration step Δt is limited by the inhomogeneity of the components on small scales and is within approximately $\Delta t \sim 10^5$ years or less. Simultaneous modeling of the gas component can significantly reduce the integration step due to larger gradients of gas density distribution. Let us estimate the errors in calculating the gravitational force for an extended system of size $r^{(\max)}$ and the minimum distance between a closely located pair of particles $r^{(\min)}$. Then the forces for nearby particles $f(r^{(\min)})$ and distant particles are related as the squares of the radii and can reach $(r^{(\min)}/r_c)^2 \simeq 10^7$ inside a typical galaxy. In the case of modeling interacting galaxies, the force ratio between the nearest particles and the most distant particles turns out to be even greater and exceeds 10^8. As a result, the contribution from distant particles adds up with an error or is even lost, depending on the length of the numbers used.

The rapid increase in the performance of modern GPUs provides new opportunities for using direct methods for calculating gravitational forces, when each particle interacts with each other (Particle-Particle algorithm, PP) [1, 16, 23] and allows to perform numerous computational experiments to study galaxies with $N > 10^6$.

The purpose of this work is a detailed analysis of the quality of galaxy simulations within the framework of the N-body method using GPUs for a direct method of calculating gravity by summing the contributions of all particles. Conducting computational experiments under various conditions is aimed at studying the effectiveness of parallel implementations of the N-body numerical algorithm on hybrid computing platforms with multi-GPUs using single and double precision floating-point arithmetics. We focus on the accuracy of the conservation laws of momentum, angular momentum and energy of the total gravitational system, depending on the number of GPUs and the use of single or double precision arithmetic.

The article is organized as follows. Section 1 contains descriptions of the numerical algorithm for the N-body problem and the main characteristics of the models of colliding galactic systems. In Section 2, we discuss the hardware and algorithmic features of the parallel implementation of calculating gravitational forces in a system of N particles. Section 3 is devoted to the analysis of the accuracy of conservation laws in a dynamic system for various ways of organizing parallel computations. Finally, the conclusion summarizes the study and outlines potential future research topics.

1. Algorithms for Integrating Equations of Motion in the N-body Model

The N-body model looks quite simple and attractive, based on a system of ordinary differential equations of motion for a large number of gravitationally interacting points

$$\frac{d^2\mathbf{r}_i}{dt^2} = \sum_{j=1}^{N} \mathbf{f}_{ij}\,, \quad \mathbf{u}_i = \frac{d\mathbf{r}_i}{dt}\,, \quad i = 1, 2, ..., N\,, \tag{1}$$

where $\mathbf{f}_{ij}$ is the force between the i-th particle with mass m_i and the j-th particle ($i \neq j$) with mass m_j. Direct calculation of the force $\mathbf{f}_{ij}$ involves the use of Newton's law in the form

$$\mathbf{f}_{ij} = \frac{Gm_j}{(r_{ij}^2 + r_c^2)^{3/2}}\,(\mathbf{r}_i - \mathbf{r}_j)\,, \tag{2}$$

where G is the gravitational constant, $r_{ij} = |\mathbf{r}_i - \mathbf{r}_j|$ is the distance between two particles, r_c is the softening radius. The value $r_c > 0$ is required to ensure that the system is collisionless in the case of $N \ll N_r$.

Directly calculating all forces in (1) using (2) gives quadratic complexity $O(N^2)$. The approximate hierarchical TreeCode method has $O(N\log(N))$, increasing the error in gravitational force calculation [12, 22].

The three-stage Newton–Störmer–Verlet-leapfrog (or Kick-Drift-Kick, KDK) scheme is traditionally used for numerical integration of system (1) with some modifications [8]. Successive calculations of intermediate velocities at the first stage

$$\widetilde{\mathbf{u}}_i(t + \Delta t) = \mathbf{u}_i(t) + \Delta t \sum_{j=1,\, j\neq i}^{N} \mathbf{f}_{ij}(t) \tag{3}$$

give the positions of all particles at time $t + \Delta t$ at the second step

$$\mathbf{r}_i(t + \Delta t) = \mathbf{r}_i(t) + \frac{\Delta t}{2}\,\left[\widetilde{\mathbf{u}}_i(t + \Delta t) + \mathbf{u}_i(t)\right]\,. \tag{4}$$

Then, we calculate the velocities at time $t + \Delta t$ in the third stage

$$\mathbf{v}_i(t + \Delta t) = \frac{\mathbf{u}_i(t) + \widetilde{\mathbf{u}}_i(t + \Delta t)}{2} + \frac{\Delta t}{2} \sum_{j=1,\, j\neq i}^{N} \mathbf{f}_{ij}(t + \Delta t)\,. \tag{5}$$

The forces at time $t + \Delta t$ are used at the next iteration, so the main advantage of KDK is only a one-time calculation of the forces on the right side of the equations (1), which, however, provides second order accuracy $O(\Delta t^2)$ at one time step.

We study the evolution of the initial states of gravitating systems related to severe tests in which complex flows are formed with the development of strong gravitational instability. Table 1 contains the number of particles in the disc $N^{(d)}$, the number of particles in the dark hot spheroidal halo $N^{(h)}$ and the Toomre parameter Q_T in the central region of the disc, characterizing the effective temperature of the matter (particle velocity dispersion). The quantity

Table 1. Main parameters of the models

Model Name	$N^{(d)}$ Discs	$N^{(h)}$ Halo	Q_T $r < 0.5$	Comment
D100	2^{18}	2^{19}	$Q_T \simeq 1.25$	Disc + halo
D101	2^{19}	2^{19}	$Q_T \simeq 0.8$	Disc + halo
D102	2^{19}	—	$Q_T \simeq 1$	Disc without halo
D103	2^{19}	—	$Q_T \simeq 0.1$	Disc without halo
D201	2^{19}+2^{19}	2^{19}+2^{19}	$Q_T \simeq 0.8$	Collision of two galaxies

$Q_T = c_r/(3.36G\sigma/\kappa)$ is traditionally used to determine the limit of gravitational stability of the stellar disc (c_r is the radial velocity dispersion, σ is the surface density, κ is the epicyclic frequency) [5]. We consider the crash tests in models D103 and D201, during which the fastest processes occur inside small sizes at large density gradients and discs destruction occurs.

The model D201 reproduces the collision of two disc systems almost flat, leading to a large merger through the passage of two galaxies through each other. Figures 1 and 2 show the dynamics of two models in three perpendicular planes. Model D103 describes an initial very cold disc with small Toomre parameter ($Q_T \simeq 0.1$), which leads to the rapid development of strong gravitational instability. This model does not contain a dark halo, which has a stabilizing effect on gravitational instability. As a result, the disc matter is divided into several massive clamps, which are slowly destroyed during the heating of the system with the formation of a hot extended disc with a significantly reconstructed radial density profile. Model D103 is not of physical interest due to the condition $Q_T \simeq 0.1$, but it is a good touchstone for checking calculations. The proximity of the parameter Q_T to zero leads to the development of the most powerful gravitational instability. As a result, a dynamically very cold disc breaks up into several isolated, long-lived, small, high-density clamps that actively interact with each other (see two bottom panels in Fig. 1). The rotating disc lies in the plane (x, y) in all models at the initial time.

Model D201 includes two identical galaxies embedded in a dark halo. We push them together at an angle of 45° (Fig. 2). This simulation of centrally colliding two-component disks with a dark, massive halo is an example of strong interaction. It ends with a large merging of the two systems. Note that the observed galaxies type Taffy for the pairs UGC12914/UGC12915, NGC7733/NGC7734 apparently goes through such a stage of evolution [3].

We use a system of dimensionless quantities to conveniently represent galactic characteristics so that all dimensionless parameters are of the order of unity under typical conditions. Conversion from standard astronomical characteristics of length (1 pc $\simeq 3.086 \cdot 10^{16}$ m), mass (1 $M_\odot = 1.989 \cdot 10^{30}$ kg, solar mass) to dimensionless quantities is carried out by the factors $\ell_r = 9000\,\mathrm{pc}$ and $\ell_M = 3.72 \cdot 10^{10}\,M_\odot$, respectively [16]. The units of time ℓ_t and velocity ℓ_V are equal to $\ell_t = 63.2\,\mathrm{Myr}$, $\ell_V = 133.7\,\mathrm{km\,s^{-1}}$.

2. Features of Parallel Code Implementations

We used hybrid computing platforms with multiple GPUs (CPU+2GPU and CPU+4GPU). The simulations on the computing architecture CPU+2GPU were carried out on Lomonosov-2 – Volta-1 (Lomonosov Moscow State University) supercomputer: CPU (Intel Xeon Gold 6142) + 2GPU (Nvidia Tesla V100). The computing architecture CPU + 4GPU was used on VolSU –

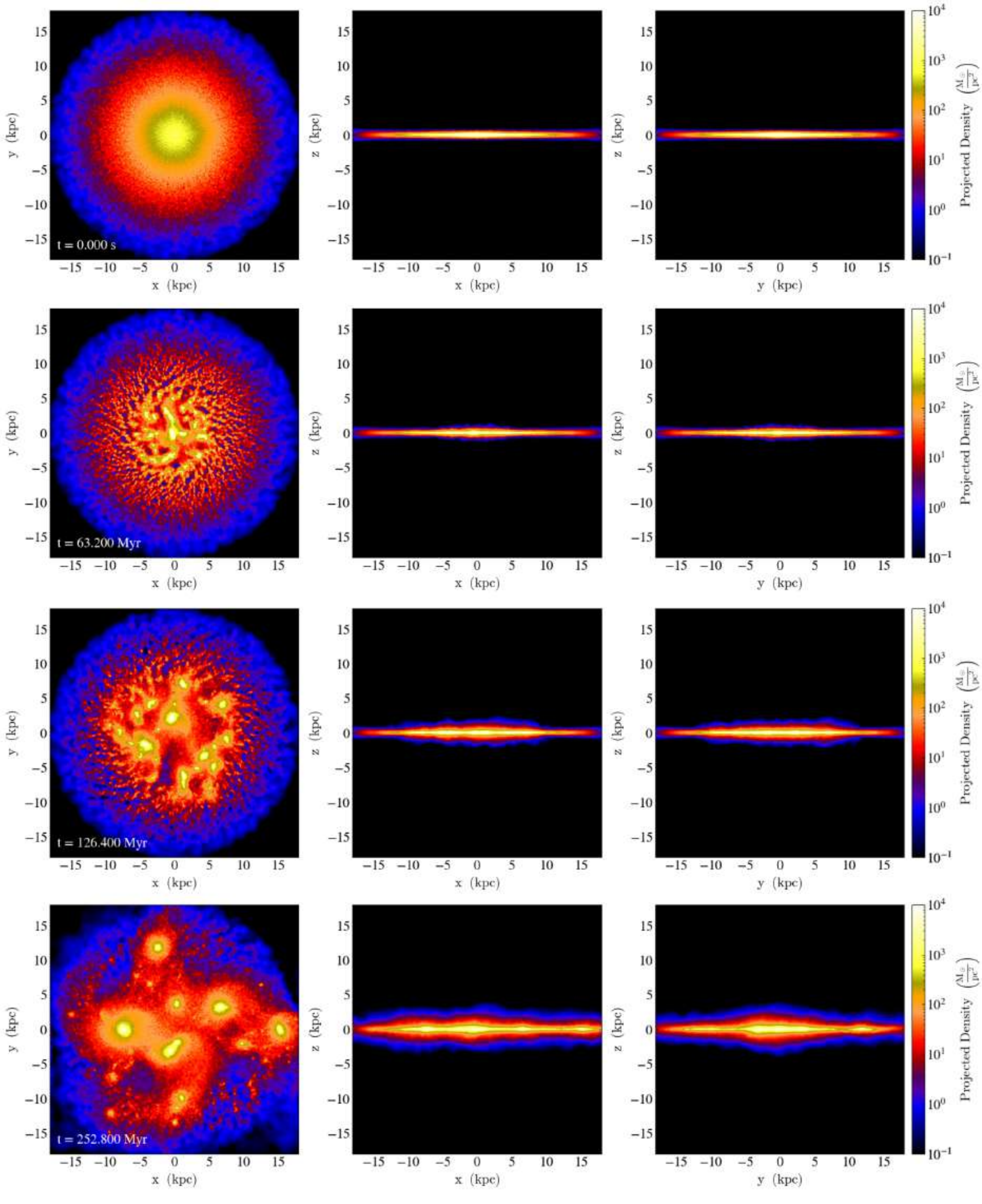


Figure 1. Evolution of an isolated very cold disc without a dark halo (model D103)

Nvidia DGX-1 supercomputer: 2CPU (Intel Xeon E5-2698) + 8GPU (Nvidia Tesla V100). We parallelized our algorithm N-body – PP based on technologies OpenMP + CUDA + GPUDirect, which allow parallel calculations to be performed on one computing node with several GPUs. OpenMP technology is used to parallel run CUDA-kernel on multiple GPUs. GPUDirect technology creates a common memory address space for multi-GPUs and allows CUDA-threads to communicate directly through the NVLINK or PCIe interface, bypassing CPU memory. Figure 3 shows the principle of organizing calculations using our algorithm N-body – PP on a hybrid com-

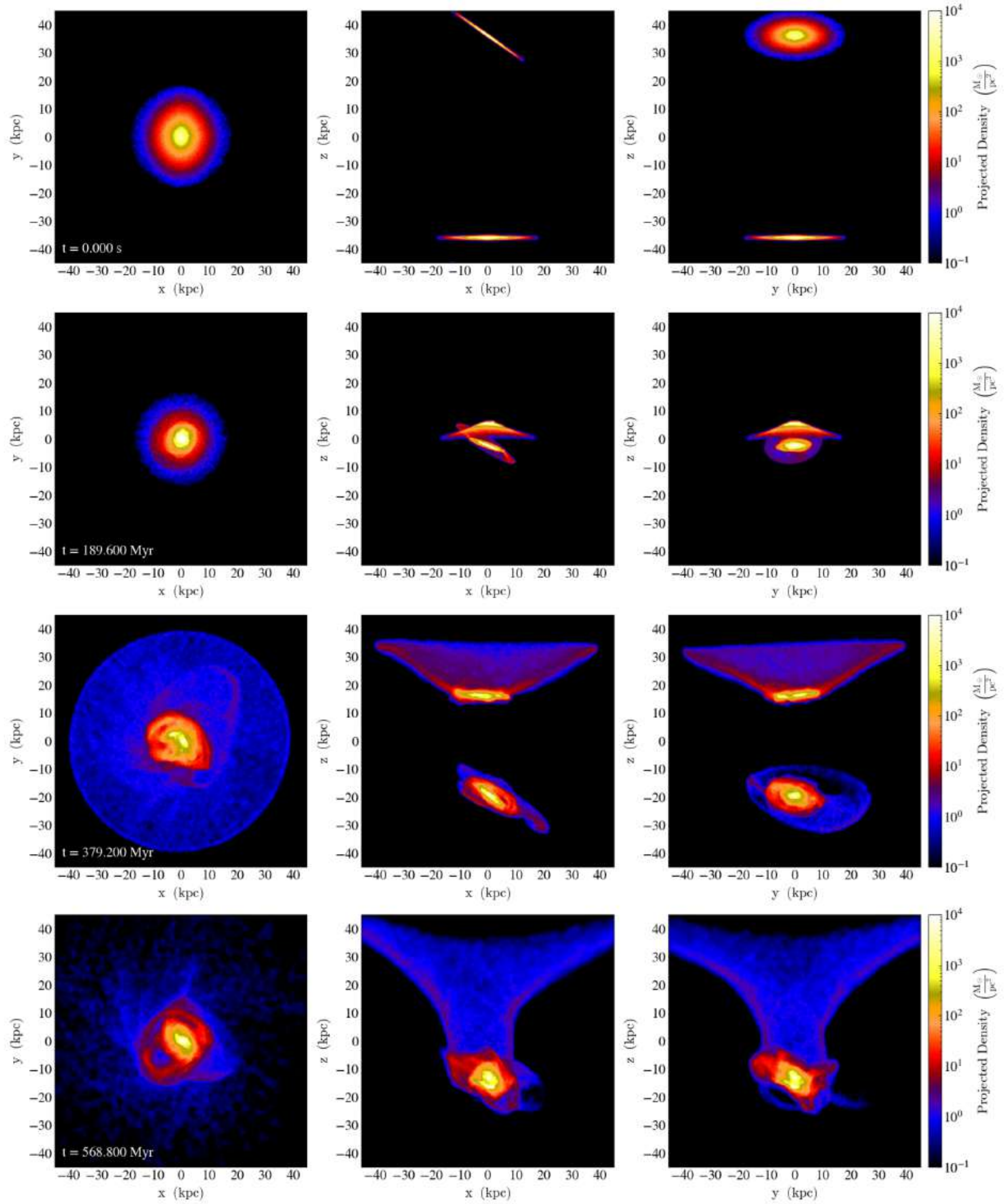
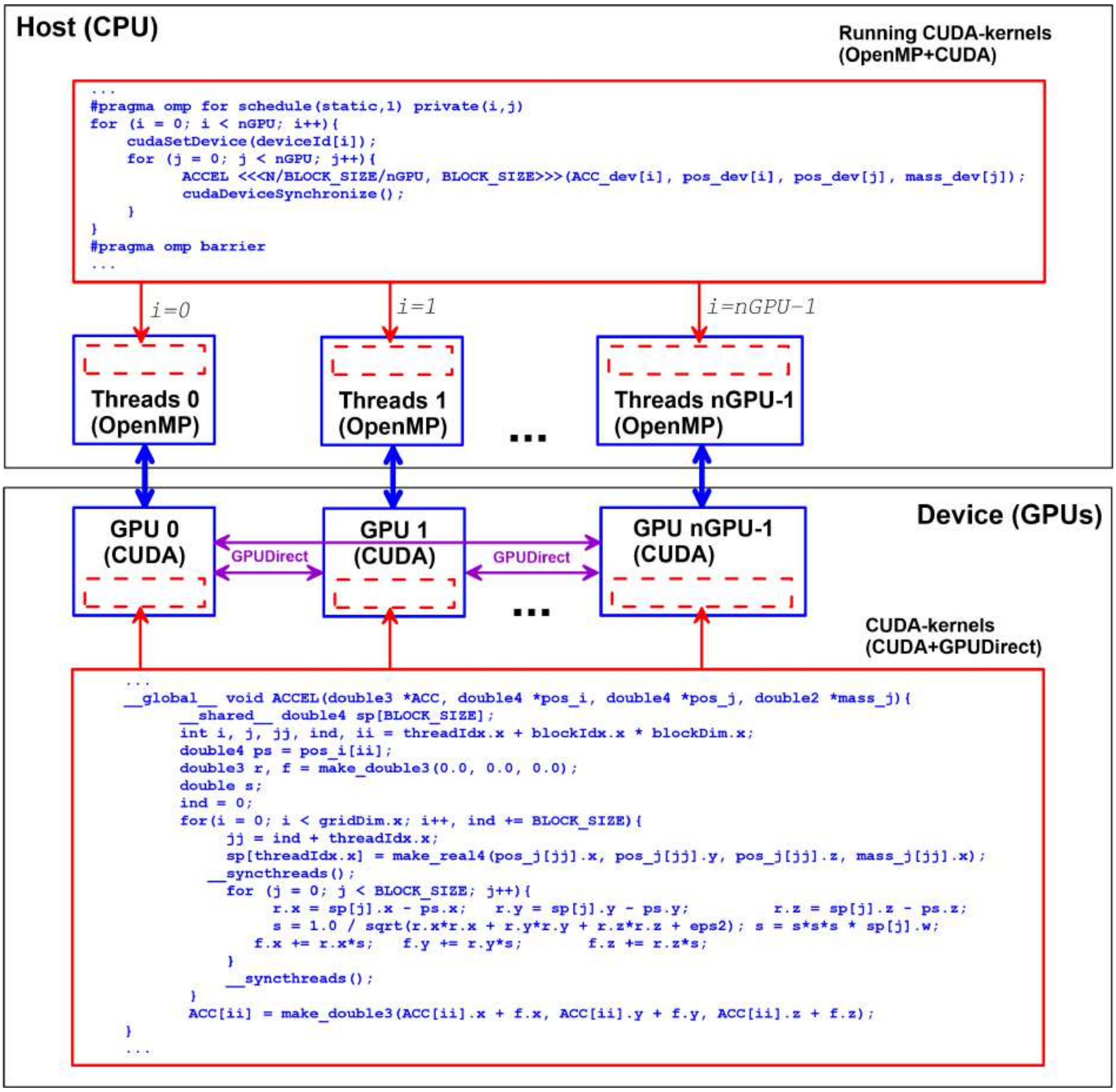


Figure 2. Evolution of two colliding Taffy-type disc galaxies (model D201)

puting platform CPU + multi-GPUs. Scheme in Fig. 3 also shows the most resource-intensive part of the code associated with calculating the gravitational interaction (PP).

The problem of the quality of numerical N-body models using high performance GPUs single-precision performance is relevant [6]. Figure 4 shows the results of analyzing the efficiency of code parallelization under various conditions. We carry out calculations with different numbers of GPUs: $n_G = 1, 2, 4$, n_GGPU. All simulations are duplicated using 4-byte (FP32) and 8-byte (FP64) numbers.



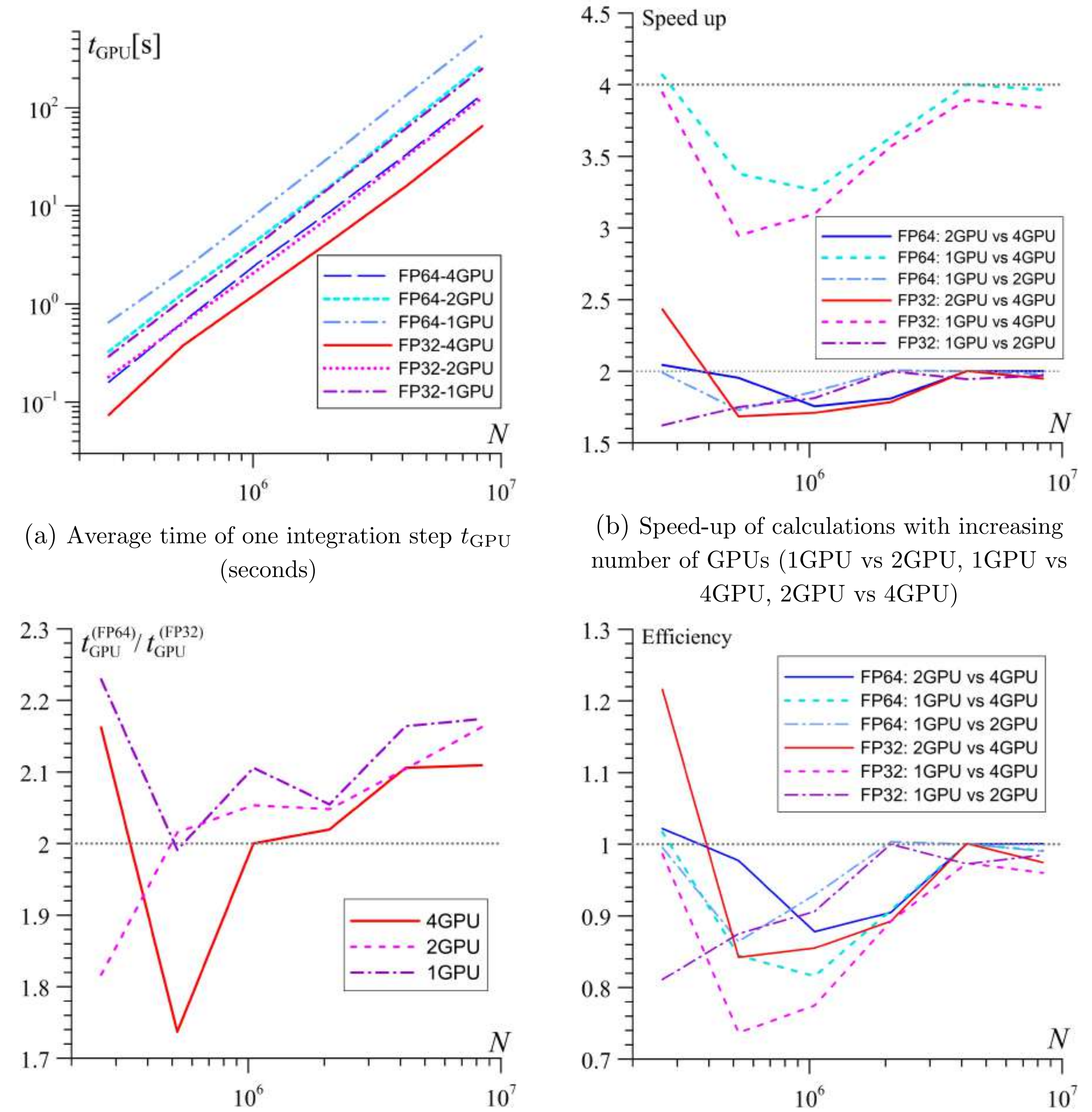
```
...
#pragma omp for schedule(static,1) private(i,j)
for (i = 0; i < nGPU; i++){
    cudaSetDevice(deviceId[i]);
    for (j = 0; j < nGPU; j++){
          ACCEL <<<N/BLOCK_SIZE/nGPU, BLOCK_SIZE>>>(ACC_dev[i], pos_dev[i], pos_dev[j], mass_dev[j]);
          cudaDeviceSynchronize();
    }
}
#pragma omp barrier
...
```

```
...
__global__ void ACCEL(double3 *ACC, double4 *pos_i, double4 *pos_j, double2 *mass_j){
      __shared__ double4 sp[BLOCK_SIZE];
      int i, j, jj, ind, ii = threadIdx.x + blockIdx.x * blockDim.x;
      double4 ps = pos_i[ii];
      double3 r, f = make_double3(0.0, 0.0, 0.0);
      double s;
      ind = 0;
      for(i = 0; i < gridDim.x; i++, ind += BLOCK_SIZE){
          jj = ind + threadIdx.x;
          sp[threadIdx.x] = make_real4(pos_j[jj].x, pos_j[jj].y, pos_j[jj].z, mass_j[jj].x);
        __syncthreads();
          for (j = 0; j < BLOCK_SIZE; j++){
              r.x = sp[j].x - ps.x;   r.y = sp[j].y - ps.y;          r.z = sp[j].z - ps.z;
              s = 1.0 / sqrt(r.x*r.x + r.y*r.y + r.z*r.z + eps2); s = s*s*s * sp[j].w;
            f.x += r.x*s;   f.y += r.y*s;        f.z += r.z*s;
          }
          __syncthreads();
       }
       ACC[ii] = make_double3(ACC[ii].x + f.x, ACC[ii].y + f.y, ACC[ii].z + f.z);
}
...
```

Figure 3. Scheme of implementation of the algorithm N-body – PP with code fragments on hybrid computing platform CPU + multi-GPUs

When analyzing the performance of parallel computing on various platforms CPU+n_GGPU, we consider only the execution time of the parallel part of our N-body–PP code (see Fig. 3), in which gravitational forces between particles are calculated by the direct method (2) and integration of the equations of motion (1) is carried out using the method (3)–(5). The time it takes to copy data from the GPU to the CPU and write it to disc is not taken into account. Figure 4a shows the dependence of the computation time for the parallel N-body – PP algorithm on the number of particles in various computational models. There is a quadratic law in the form $t_{\rm GPU} \propto N^2/n_c$, where n_c is the total number of computing cores of GPU.

The Nvidia Tesla V100 GPU contains $n_c = 2560$ cores for double precision (FP64) and $n_c = 5120$ cores for single precision (FP32), so the computational performance for numbers FP32 is approximately 2 times faster than with FP64 (see Fig. 4a,c). Figures 4b,d demonstrate the characteristic features of parallelization of the N-body – PP algorithm on multi-GPUs. The



(a) Average time of one integration step t_{GPU} (seconds)

(b) Speed-up of calculations with increasing number of GPUs (1GPU vs 2GPU, 1GPU vs 4GPU, 2GPU vs 4GPU)

(c) Ratio $t_{\mathrm{GPU}}^{(FP64)}/t_{\mathrm{GPU}}^{(FP32)}$, where $t_{\mathrm{GPU}}^{(FP64)}$, $t_{\mathrm{GPU}}^{(FP32)}$ are the average times of one integration step with FP64 and FP32, respectively

(d) Parallelization efficiency of calculations with increasing number of GPUs (1GPU vs 2GPU, 1GPU vs 4GPU, 2GPU vs 4GPU)

Figure 4. Dependence of the performance of multi-GPU calculations on the number of particles N in various computational models for the N-body – PP algorithm

speed-up and efficiency curves have a minimum near $N = 2^{19}$–2^{20} when comparing different computational models. The parallelization efficiency of our algorithm on multi-GPUs tends to 1 as the number of particles increases after $N \geq 2^{22}$. Note the anomalous behavior of speed-up and efficiency at $N = 2^{18}$, which may be associated with an increase in the data transfer rate between GPUs via NVLINK interface when the data volume is less than a certain threshold value.

The performance of two hybrid computing platforms CPU+2GPU and 2CPU+8GPU was compared for our algorithm N-body–PP. Supercomputer Lomonosov-2 – Volta-1 for computing on 1GPU and 2GPU with numbers FP64 is approximately 4–5 percent more productive than VolSU – DGX-1.

Data copy time between CPU (Device) and GPU (Host) depends on the memory bus bandwidth and the amount of data being copied. Therefore, the copying time is proportional to the number of particles N. The calculation time of gravitational forces in our N-Body-Particle-Particle algorithm is $\propto N^2$, so replacing the GPU-Direct technology with direct copying of data between the GPU and CPU in our code does not lead to a significant decrease in speed-up and parallelization efficiency on multi-GPU at $N > 10^5$. These times can be comparable only for very small N. Our estimates of the speed-up degradation for different values of N show that the speed-up of computations without using GPU-Direct is less than 1% for $N > 2^{18}$ (Tab. 2). The use of GPU-Direct technology in numerical algorithms with lower computational complexity, for example, $\propto N \ln N$ for treecode or $\propto N$ in the case of hydrodynamic simulations, should lead to more significant gains in speed-up and parallelization efficiency on multi-GPUs. Direct data copying between GPU and CPU has another drawback, which is the duplication of arrays as the number of GPUs increases, since all particle positions must be stored on each GPU at each computational time. This results in an increase in memory space by a factor of k (where $k \simeq 0.58 + 0.42 \cdot n\text{GPU}$) on each GPU compared to GPU-Direct.

Table 2. Average time of one integration step on multi-GPU without using GPU-Direct (t^*_{nGPU}) and using GPU-Direct (t_{nGPU}) for different N

	$N = 2^{18}$	$N = 2^{19}$	$N = 2^{20}$	$N = 2^{21}$	$N = 2^{22}$	$N = 2^{23}$
t^*_{2GPU}, [s]	0.6484×2^{-1}	0.6475×2^{1}	0.5661×2^{3}	0.5276×2^{5}	0.5253×2^{7}	0.5254×2^{9}
t_{2GPU}, [s]	0.6444×2^{-1}	0.6450×2^{1}	0.5648×2^{3}	0.5268×2^{5}	0.5249×2^{7}	0.5253×2^{9}
t^*_{4GPU}, [s]	0.3238×2^{-1}	0.3303×2^{1}	0.3289×2^{3}	0.2877×2^{5}	0.2679×2^{7}	0.2681×2^{9}
t_{4GPU}, [s]	0.3210×2^{-1}	0.3285×2^{1}	0.3280×2^{3}	0.2872×2^{5}	0.2678×2^{7}	0.2680×2^{9}

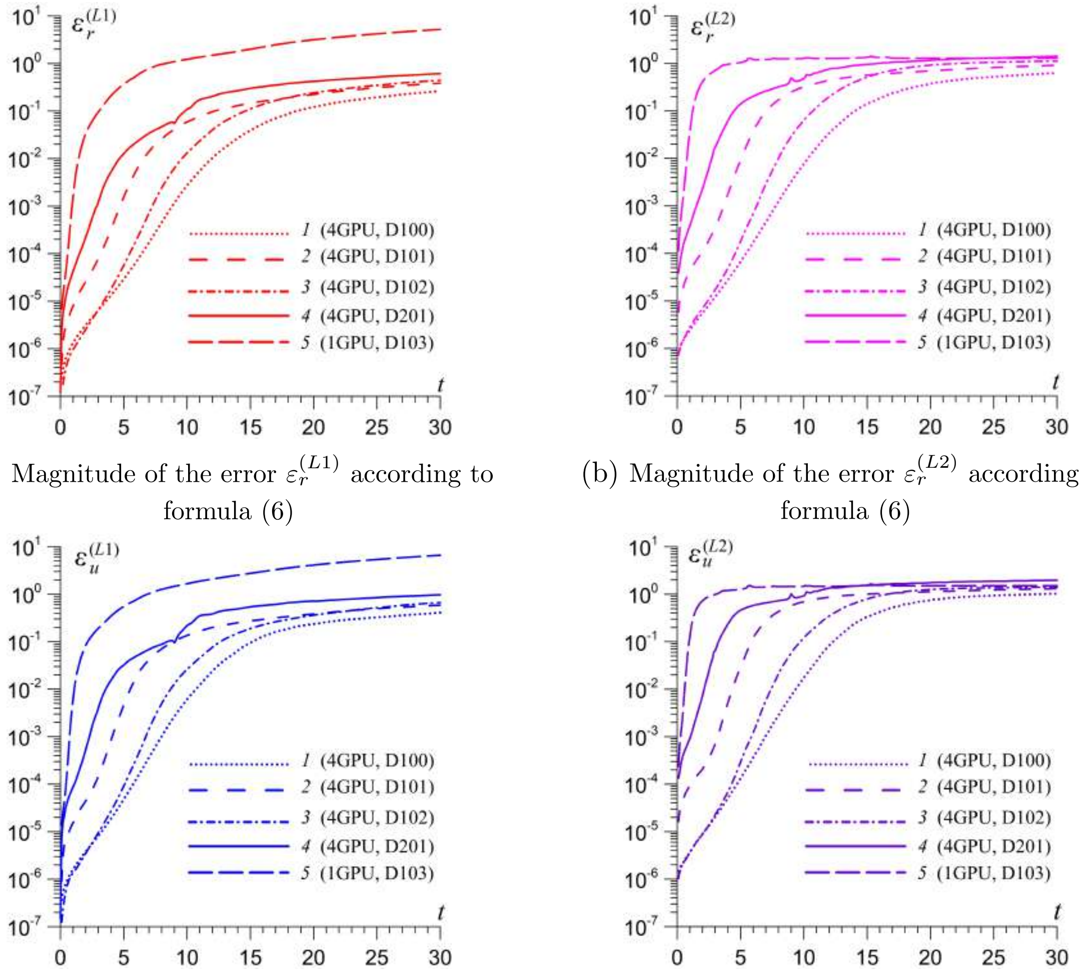
The complete set of trajectories in the phase space $\{\mathbf{r}_i, \mathbf{u}_i\}$ $(i = 1, 2, ..., N)$ depends on the length of the numbers (FP32 or FP64), all other things being equal. The average divergence of such trajectories is determined by calculating the parameters

$$\varepsilon_r^{(L1)} = \frac{1}{N}\sum_{i=1}^{N}\left|\mathbf{r}_i^{(FP32)} - \mathbf{r}_i^{(FP64)}\right| , \qquad \varepsilon_r^{(L2)} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left|\mathbf{r}_i^{(FP32)} - \mathbf{r}_i^{(FP64)}\right|^2} , \tag{6}$$

$$\varepsilon_u^{(L1)} = \frac{1}{N}\sum_{i=1}^{N}\left|\mathbf{u}_i^{(FP32)} - \mathbf{u}_i^{(FP64)}\right| , \qquad \varepsilon_u^{(L2)} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left|\mathbf{u}_i^{(FP32)} - \mathbf{u}_i^{(FP64)}\right|^2} \tag{7}$$

for the metrics $L1$ and $L2$ at each time instant t during the simulations.

Figure 5 shows the average integral divergence of trajectories in phase space in accordance with (6) and (7) in different models (see Tab. 1) on 1GPU and 4GPU. The accumulation of errors occurs primarily at the initial stage of evolution, when powerful spiral structures are formed in an isolated disk due to gravitational instability (models D100–D103). A similar dependence is obtained in the process of a large merger of two galaxies into one in the D201 model. The evolution of all models over long times ends in quasi-stationary states, when macroscopic characteristics (density, velocities, dispersions of velocity components) practically cease to change. Such quasi-stationary systems are characterized by a very slow increase in the parameters $\varepsilon_{r,u}^{(L1)}$, $\varepsilon_{r,u}^{(L2)}$ or no changes.



(a) Magnitude of the error $\varepsilon_r^{(L1)}$ according to formula (6)

(b) Magnitude of the error $\varepsilon_r^{(L2)}$ according to formula (6)

(c) Magnitude of the error $\varepsilon_u^{(L1)}$ according to formula (7)

(d) Magnitude of the error $\varepsilon_u^{(L2)}$ according to formula (7)

Figure 5. Dependences of errors $\varepsilon_{r,u}^{(L1)}$, $\varepsilon_{r,u}^{(L2)}$ on time for different experiments with different numbers of GPUs

A stronger initial nonstationarity of the gravitating system leads to a more rapid growth of $\varepsilon_{r,u}^{(L1)}$, $\varepsilon_{r,u}^{(L2)}$, which stops at more high level. The smallest discrepancies between the phase trajectories are obtained in model D100, in which the initial disc is only marginally unstable and the slow formation of spiral arms of small amplitude is observed.

3. Problems of Fulfilling Conservation Laws

The law of conservation of energy E for a system of N interacting particles in the absence of dissipation and external forces is determined by the following expression:

$$E = \sum_{i=1}^{N} \frac{m_i |\mathbf{v}_i|^2}{2} - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \frac{G\, m_i m_j}{\left(r_{ij}^2 + r_c^2\right)^{1/2}} \,, \tag{8}$$

where m_i is the mass of the i-th particle, $j \neq i$, $r_{ij} = |\mathbf{r}_i - \mathbf{r}_j|$ is the distance between two points.

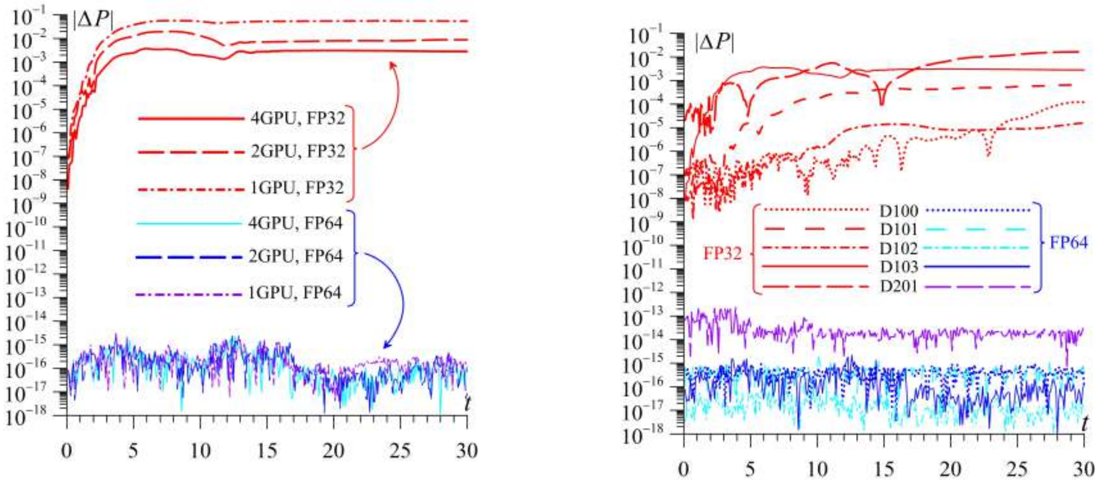
Conservation of total momentum

$$\mathbf{P} = \sum_{i=1}^{N} m_i \mathbf{u}_i \,, \tag{9}$$

and angular momentum

$$\mathbf{L} = \sum_{i=1}^{N} m_i [\mathbf{r}_i \times \mathbf{u}_i]_z \tag{10}$$

in explicit form does not depend on the softening radius r_c.



(a) Changes in total momentum $|\Delta\mathbf{P}|$ in model D103

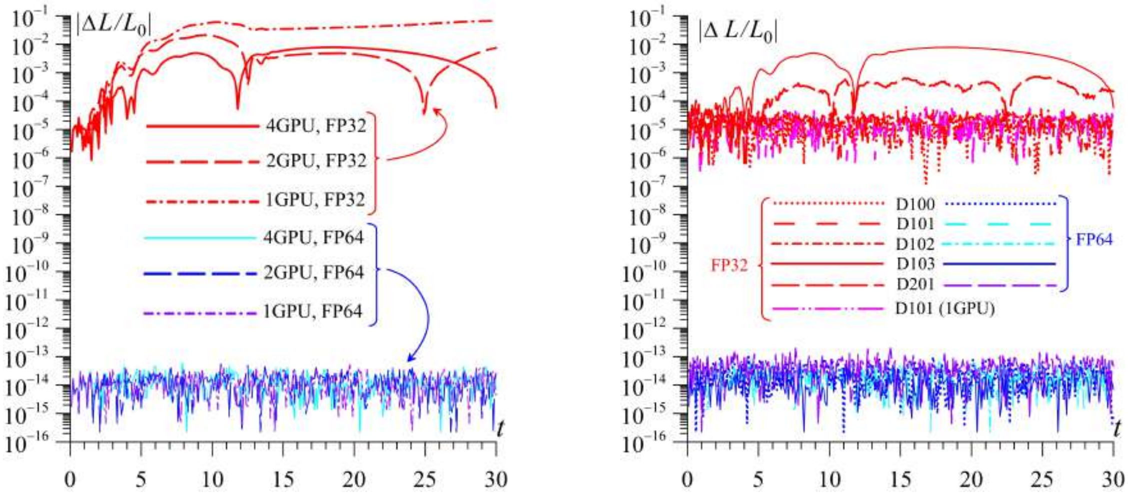
(b) Models from Tab. 1 were calculated on 4GPU

Figure 6. Changes in total momentum in the system $|\Delta\mathbf{P}|$ under different conditions

Modeling of a gravitating system in accordance with (1) should ensure the fulfillment of the laws of conservation of total momentum **P** (9), angular momentum **L** (10) and energy (8). We consider in detail the problem of conservation of (8)–(9) when using numbers of different lengths in parallel calculations with different numbers of GPUs. Figure 6 shows the accuracy of total momentum conservation depending on the calculation conditions. Analysis of motion trajectories gives the worst results for model D103 in Fig. 5, therefore, the dependencies $|\Delta\mathbf{P}(t)|$ for this model are constructed separately (Fig. 6b), where two features stand out. Firstly, calculations with FP32 give a rapid increase in error and $|\Delta\mathbf{P}|$ increases by 5–6 orders of magnitude. Using FP64 keeps $|\Delta\mathbf{P}|$ approximately at the entry level within 10^{-17}–10^{-15}. This finding holds true for any model and number of GPUs. The second feature is more subtle and is related to the number of GPUs used. Momentum is better preserved as the number of GPUs increases, and this effect can be significant (compare the red lines in Fig. 6a).

Analysis of the simulation results of various models from Tab. 1 confirms the inadmissibility of using FP32, which leads to large errors for **P** (see Fig. 6b for 4GPU). The error only increases when using 2GPU or 1GPU. Higher starting level $|\Delta\mathbf{P}| \sim 10^{-5}$ in model D201 is associated with the peculiarity of constructing the initial state for a pair of colliding galaxies. However, it is important to emphasize that calculations with FP64 keep $|\Delta\mathbf{P}|$ within the initial limits ($\sim 10^{-13}$).

Disc galactic subsystems rotate rapidly and the azimuthal velocity significantly exceeds the characteristic thermal velocities of particles in the disc. This rotation velocity is comparable to the thermal velocities of the dark matter in the halo. The total initial angular momentum of the



(a) Changes $\Delta L/L_0$ in model D103

(b) Models from Tab. 1

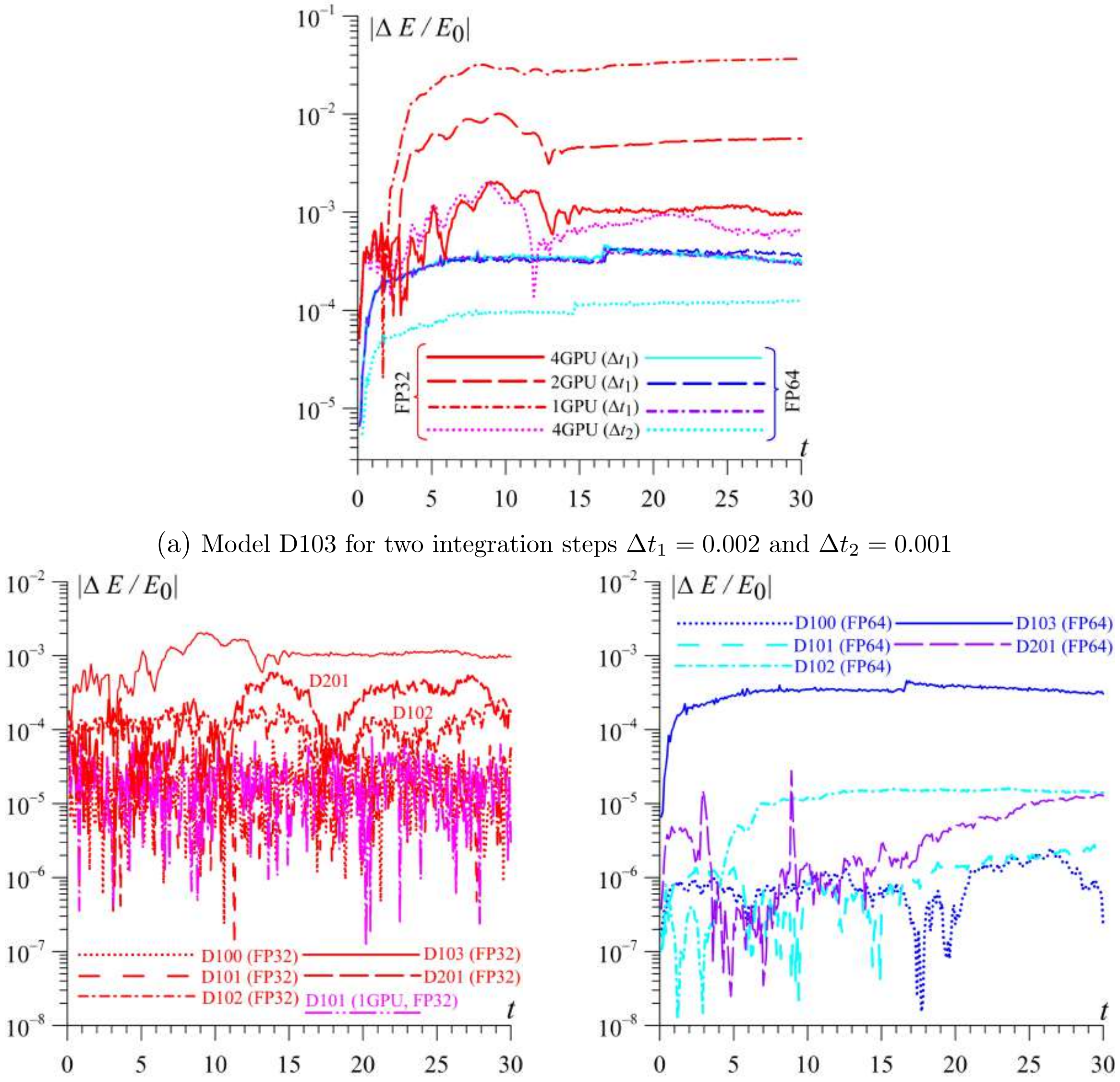
Figure 7. Changes in total angular momentum in the system $\Delta L/L_0$ (L_0 is the angular momentum at time $t = 0$)

dark halo is close to zero in our models and can then arise from tidal interactions of the halo with disturbances in the disc. Noticeable rotation of the halo can only occur in model D201 at long times after merging.

Figure 7 shows the relative changes in total angular momentum $\Delta L/L_0$, where L_0 is the initial angular momentum. All models from Tab. 1 are calculated on 4GPU. Model D101 on 1GPU is additionally shown as a magenta curve in Fig. 7b. Conservation of angular momentum plays an important role since the disc is in the balance of primarily gravitational and centrifugal forces. Therefore, even small disturbances lead to radial imbalances of forces in the disc, which is accompanied by radial movements of the matter. Behavior of curves in Fig. 7a is generally similar to the results of calculations of $|\Delta \mathbf{P}|$. The evolution of $\Delta L(t)/L_0$ for five models on 4GPU with FP32 shows that if single disc are close to the stability limit $Q_T \simeq 1$ (models D100, D101, D102), then the relative angular momentum error does not increase and remains within $< 10^{-4}$. Only very cold discs or merging models give an increase in error in the case of FP32. All our models with FP64 retain angular momentum up to 13 digits.

The results of checking the law of conservation of energy are shown in Fig. 8. Total energy is less well conserved compared to momentum and angular momentum, since the velocities in the kinetic part of the energy in (8) are calculated by approximately solving the equations of motion (1). Curves $\Delta E(t)$ in Fig. 8a describe the worst-case model D103, in which the difference in calculations using FP32 and FP64 is only 3:1 for step $\Delta t_1 = 0.002$. We have strong differences between the curves ($\Delta E(t)$) when using 1GPU, 2GPU and 4GPU with FP32, as in the case of momentum and angular momentum in Fig. 6a, 7a. Calculations with FP64 are slightly dependent on the choice of 1GPU/2GPU/4GPU. Thus, the accumulation of error due to arithmetic rounding on short 4-byte numbers significantly depends on the number of GPUs used. Increasing the number of GPUs reduces error very effectively, bringing the results closer to DF64 calculations, which are almost independent of n_G (see Fig. 8a for calculations with FP64).

Reducing the integration step by half from $\Delta t_1 = 0.002$ to $\Delta t_2 = 0.001$ naturally reduces the error (compare the solid and dotted light-blue lines in Fig. 8a). This decrease is $n_t^2 = 4$ times at the beginning of evolution in accordance with scheme (3)–(5) and then reaches 2.5 due to the



(a) Model D103 for two integration steps $\Delta t_1 = 0.002$ and $\Delta t_2 = 0.001$

(b) All models from Tab. 1 were calculated on 4GPU

(c) Same as panel (b) for FP64

Figure 8. Changes in total relative energy $\Delta E/E_0$ (E_0 is the energy at time $t = 0$). Model D101 on 1GPU with FP32 is shown additionally by magenta line in panel (b)

accumulation of arithmetic error in end of calculations ($t = 30$). The error under consideration is determined by the order of the numerical scheme n_t and the integration step Δt: $O(\Delta t^{n_t})$. The transition from Δt_1 to Δt_2 in the case of 4-byte numbers almost does not reduce the relative energy error.

Models D100, D101, D102 with typical galactic spiral patterns have an energy error approximately an order of magnitude smaller for FP32 compared to model D103, all other things being equal (Fig. 8b). Calculations with FP64 give an acceptable error already at Δt_1 (Fig. 8c).

Accumulating errors in conservation laws are reflected in the evolution of macroscopic characteristics. Figures 9, 10 compare surface density distributions in three projections, constructed in model D103 with FP32 and FP64. There are comparable differences in velocity fields, distributions of velocity dispersion components, etc. The distributions of matter along the line of sight in Fig. 9 with FP32 (bottom) and FP64 (top) give qualitatively similar structures at

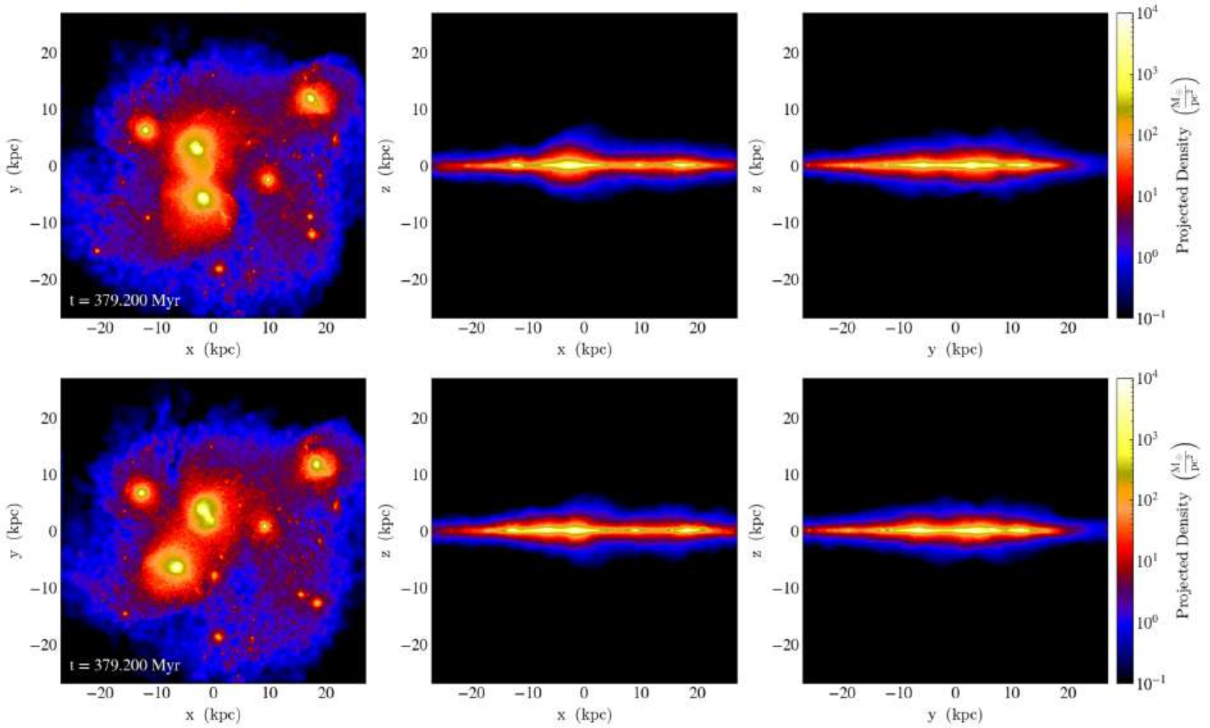


Figure 9. Density distributions along the line of sight, demonstrating the differences in numerical solutions in model D103 for FP64 (top) and FP32 (bottom) at $t = 6$

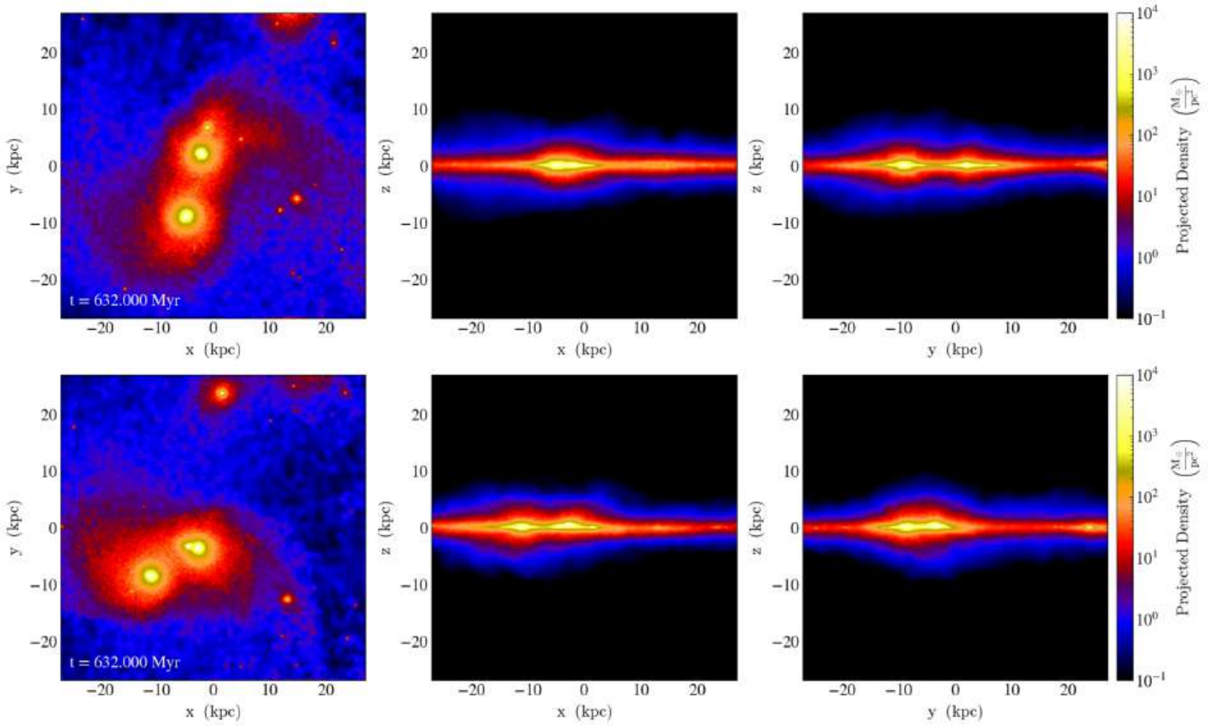


Figure 10. As in Fig. 9 at time $t = 10$

time $t = 379$ Myr. However, we already see noticeable differences in the positions of the density clamps and their relative orientation. These deviations quickly increase and the picture in Fig. 10 is already qualitatively different when comparing between FP32 and FP64 at time $t = 632$ Myr.

The differences in models D100, D101, D102 are weaker, since the amplitudes of disturbances are smaller in these models compared to D103. However, the conclusion remains that it is impossible to quantitatively study galactic systems within the framework of 4-byte arithmetic.

Discussion and Conclusion

We analyzed the implementation of the laws of conservation of momentum, angular momentum and energy in models of the dynamics of gravitationally interacting N-bodies. Such models are a traditional tool for studying globular clusters, open clusters, galaxies and galaxy clusters [1, 13, 16, 18, 25]. The system of gravitationally interacting points simulates the movements of both stars and dark matter. Our models contain both of these components. Adding gas to the model is possible when using smoothed-particle hydrodynamics, since it allows an end-to-end method for calculating gravitational forces [17].

The direct method of calculating the gravitational force by summing the contributions of all particles from each other "Particle–Particle" provides the most accurate result for a fixed number of particles N. However, some features of the organization of parallel computing on GPUs can have a significant impact on the error in N-body modeling even for an accurate method.

There are two factors that we investigated. Firstly, this is the number of significant digits. In practice, there is a choice between 4-byte and 8-byte numbers. The efficiency of operations with numbers of different lengths is very sensitive to the microarchitecture of modern GPUs. For example, the execution time of an operation with FP32 and FP64 on the V100 GPU differs by 2 times. Similar calculations on NVIDIA RTX4090 GPU differ by almost an order of magnitude. The second factor is related to the use of different numbers of n_G, in particular, calculations on 1GPU, 2GPU and 4GPU are considered, which also affects the error of long-term modeling of complex structures. Increasing the number of n_G leads to a decrease in the number of particles processed on one GPU, which in turn reduces the accumulation of error when using numbers FP32.

Graphics cards are designed for single precision arithmetic, and implementing double precision for many types of graphics accelerators requires disproportionate time resources, as is the case, for example, with the RTX4090. The considered solution to the N-body problem on RTX4070/RTX4090 with FP32 and FP64 differs by approximately an order of magnitude in execution time. Therefore, only the NVIDIA GPU Kx0/Pascal/Volta/Ampere line provides an acceptable transition to double-precision. The area of application of GPUs with FP32 are machine learning algorithms mainly [21, 28].

The law of conservation of energy always has an error due to the approximate method of integrating the equations of motion. This error can accumulate at each subsequent integration step under conditions where the number of iterations is on the order of 10^5, and contributions to the gravitational force from different particles can differ by 6 orders of magnitude or more.

In principle, we can provide the laws of conservation of total momentum $\mathbf{P}$ and total angular momentum $\mathbf{L}$ close to the limit of arithmetic resolution at the level of 13 digits or even better. The conservation of the value $\mathbf{P}$ is a reflection of the accuracy of the execution of Newton's third law $\mathbf{f}_{ij} = -\mathbf{f}_{ji}$. In the case of a sequential version of the program, it is easy to achieve exact fulfillment of this condition and further reduce the number of operations by 2 times thanks to optimization of the algorithm. In the case of CUDA parallelization, the requirement to satisfy the condition $\mathbf{f}_{ij} = -\mathbf{f}_{ji}$ is always accompanied by an increase in computation time due to an increase in the complexity of the algorithm [4].

Thus, we highlight three main results.

1) The parallelization efficiency of the N-body – PP algorithm on multi-GPU varies between 0.8–1.2 depending on the number of particles N, the number of GPUs and the choice of single or double precision floating-point numbers. Qualitative modeling of galaxy dynamics requires the use of a number of gravitating particles $N > 10^6$. The efficiency of parallelizing such numerical models on a multi-GPU tends to unity.

2) The test of the laws of conservation of energy, momentum and angular momentum for the long-term evolution of gravitating systems showed a strong dependence of errors on the digits of the floating-point numbers used. The decrease in the accuracy of conservation laws for single-precision operations is due to the accumulation of arithmetic errors due to two factors. Firstly, the sum of gravitational forces from different particles contains terms that differ by several orders of magnitude, which leads to a loss of accuracy. Effective implementation of the CUDA algorithm on high-performance GPUs requires the use of a very large number of parallel threads (10^5–10^6). The execution order of these threads is determined by the built-in CUDA scheduler and cannot be determined in the source code. Secondly, studying the dynamics of galaxies over cosmological time corresponds to more than 10^5 integration steps, which also requires calculations with 8-byte numbers.

3) An increase in the number of GPUs used contributes to a more accurate implementation of conservation laws in the case of 4-byte arithmetic due to a decrease in the number of particles per GPU. Conservation laws in double-precision models are always fulfilled with high accuracy and do not depend on the number of GPUs.

Acknowledgements

This work is supported by the Russian Science Foundation (grant No. 23-71-00016, https://rscf.ru/project/23-71-00016/. The research also relied on the shared research facilities of HPC computing resources at Lomonosov Moscow State University.

References

1. Aarseth, S.J.: Gravitational N-Body Simulations: Tools and Algorithms. Cambridge University Press, Cambridge (2009)

2. Abraham, M.J., Murtola, T., Schulz, R., *et al.*: Gromacs: High performance molecular simulations through multi-level parallelism from laptops to supercomputers. SoftwareX 1-2, 19–25 (2015). `https://doi.org/10.1016/j.softx.2015.06.001`

3. Appleton, P.N., Emonts, B., Lisenfeld, U., *et al.*: The CO Emission in the Taffy Galaxies (UGC 12914/15) at 60 pc Resolution. I. The Battle for Star Formation in the Turbulent Taffy Bridge. Astrophysical Journal 931(2), 121 (2022). `https://doi.org/10.3847/1538-4357/ac63b2`

4. Belleman, R.G., Be dorf, J., Zwart, S.F.P.: High performance direct gravitational N-body simulations on graphics processing units II: An implementation in CUDA. New Astronomy 13, 103–112 (2008). https://doi.org/10.1016/j.newast.2007.07.004

5. Binney, J., Tremaine, S.: Galactic Dynamics. Princeton University Press, Princeton (2008)

6. Brasser, R., Grimm, S.L., Hatalova, P., Stadel, J.G.: Speeding up the GENGA N-body integrator on consumer-grade graphics cards. Astronomy & Astrophysics 678, A73 (2023). https://doi.org/10.1051/0004-6361/202347071

7. Bruno, D., Capitelli, M., Longo, S., *et al.*: Particle kinetic modelling of rarefied gases and plasmas. Plasma Sources Science and Technology 12(4), S89 (2003). https://doi.org/10.1088/0963-0252/12/4/024

8. Eckmann, J.P., Hassani, F.: The detection of relativistic corrections in cosmological N-body simulations. Celestial Mechanics and Dynamical Astronomy 132(2) (2020). https://doi.org/10.1007/s10569-019-9943-z

9. Fedorov, V.A., Kholina, E.G., Gudimchuk, N.B., Kovalenko, I.B.: High-performance computing of microtubule protofilament dynamics by means of all-atom molecular modeling. Supercomputing Frontiers and Innovations 10(4), 62–68 (2023). https://doi.org/10.14529/jsfi230406

10. Greengard, L.: The Numerical Solution of the N-Body Problem. Computers in Physics 4, 142–152 (1990). https://doi.org/10.1063/1.4822898

11. Grigoriev, F.V., Sulimov, V.B., Tikhonravov, A.V.: Study of thin optical films properties using high-performance atomistic simulation. Supercomputing Frontiers and Innovations 11(1), 97–108 (2024). https://doi.org/10.14529/jsfi240105

12. Hopkins, P.F., Nadler, E.O., Grudic, M.Y., *et al.*: Novel conservative methods for adaptive force softening in collisionless and multispecies N-body simulations. Monthly Notices of the Royal Astronomical Society 525(4), 5951–5977 (2023). https://doi.org/10.1093/mnras/stad2548

13. Ishchenko, M., Kovaleva, D.A., Berczik, P., *et al.*: Star-by-star dynamical evolution of the physical pair of the Collinder 135 and UBC 7 open clusters. Astronomy & Astrophysics 686, A225 (2024). https://doi.org/10.1051/0004-6361/202348978

14. Kamlah, A.W.H., Leveque, A., Spurzem, R., *et al.*: Preparing the next gravitational million-body simulations: evolution of single and binary stars in NBODY6++GPU, MOCCA, and MCLUSTER. Monthly Notices of the Royal Astronomical Society 511(3), 4060–4089. https://doi.org/10.1093/mnras/stab3748

15. Khan, R., Kandappan, V.A., Ambikasaran, S.: HODLRdD: A new black-box fast algorithm for N-body problems in d-dimensions with guaranteed error bounds: Applications to integral equations and support vector machines. Journal of Computational Physics 501, 112786 (2024). https://doi.org/10.1016/j.jcp.2024.112786

16. Khoperskov, A.V., Khrapov, S.S., Sirotin, D.S.: Formation of transitional cE/UCD galaxies through massive disc to dwarf galaxy mergers. Galaxies 12(1), 1 (2024). https://doi.org/10.3390/galaxies12010001

17. Khrapov, S.S., Khoperskov, A.V.: Retrograde infall of the intergalactic gas onto S-galaxy and activity of galactic nuclei. Open Astronomy 33(1), 20220231 (2024). https://doi.org/10.1515/astro-2022-0231

18. Khrapov, S.S., Khoperskov, A.V., Zaitseva, N.A., *et al.*: Formation of spiral dwarf galaxies: observational data and results of numerical simulation. St. Petersburg State Polytechnical University Journal. Physics and Mathematics 16(1.2), 395–402 (2023). https://doi.org/10.18721/JPM.161.260

19. Li, Y., Pinto, M.C., Holderied, F., *et al.*: Geometric Particle-In-Cell discretizations of a plasma hybrid model with kinetic ions and mass-less fluid electrons. Journal of Computational Physics 498, 112671 (2024). https://doi.org/10.1016/j.jcp.2023.112671

20. Liseykina, T.V., Dudnikova, G.I., Vshivkov, V.A., *et al.*: MHD-PIC Supercomputer Simulation of Plasma Injection into Open Magnetic Trap. Supercomputing Frontiers and Innovations 10(3), 11–17 (2024). https://doi.org/10.14529/jsfi230302

21. Navarro, C.A., Hitschfeld-Kahler, N., Mateu, L.: A Survey on Parallel Computing and its Applications in Data-Parallel Problems Using GPU Architectures. Communications in Computational Physics 15(2), 285–329 (2014). https://doi.org/10.4208/cicp.110113.010813a

22. Ong, B.W., Dhamankar, S.: Towards an Adaptive Treecode for N-body Problems 82, 72 (2020). https://doi.org/10.1007/s10915-020-01177-1

23. Rantala, A., Naab, T., Rizzuto, F.P., *et al.*: Bifrost: simulating compact subsystems in star clusters using a hierarchical fourth-order forward symplectic integrator code. Monthly Notices of the Royal Astronomical Society 522(4), 5180–5203 (2023). https://doi.org/10.1093/mnras/stad1360

24. Romeo, A.B., Horellou, C., Bergh, J.: N-body simulations with two-orders-of-magnitude higher performance using wavelets. Monthly Notice of the Royal Astronomical Society 342(2), 337–344 (2003). https://doi.org/10.1046/j.1365-8711.2003.06549.x

25. Smirnov, A.A., Sotnikova, N.Y., Koshkin, A.A.: Simulations of slow bars in anisotropic disk systems. Astronomy Letters 43(2), 61–74 (2017). https://doi.org/10.1134/S1063773717020062

26. Voevodin, V.V., Chulkevich, R.A., Kostenetskiy, P.S., *et al.*: Administration, Monitoring and Analysis of Supercomputers in Russia: a Survey of 10 HPC Centers. Supercomputing Frontiers and Innovations 8(3), 82–103 (2021). https://doi.org/10.14529/jsfi210305

27. Yokota, R., Barba, L.A.: Treecode and Fast Multipole Method for N-Body Simulation with CUDA. Springer (2011). https://doi.org/10.1016/B978-0-12-384988-5.00009-7

28. Zhang, H., Si, S., Hsieh, C.J.: Gpu-acceleration for large-scale tree boosting. Eprint arXiv 1706.08359 (2017). https://doi.org/10.48550/arXiv.1706.08359

29. Zhou, K., Liu, B.: Molecular Dynamics Simulation: Fundamentals and Applications. Elsevier, Amsterdam (2022)