Bull. Nov. Comp. Center, Num. Anal., 16 (2013), 81–92

The advantage of the GPU-based supercomputer simulation of plasma phenomena*

A.V. Snytnikov, A.A. Romanenko

Abstract. A 3D kinetic study of the plasma relaxation processes caused by the propagation of an electron beam in high-temperature plasma was carried out. The mathematical model is built on the basis of the Particle-in-Cell (PIC) method. The performance for supercomputers powered by both Intel Xeon processors and Nvidia Tesla GPUs (Graphic Processor Units) is given. It is shown that the particle motion stage is computed much faster with Tesla. Also, the optimization techniques are given for the PIC simulation with GPUs.

Keywords: parallel computing, high-temperature plasma, PIC method, GPU.

1. Introduction

Currently, Graphic Processor Units are widely used to accelerate the simulations of various natural phenomena. The GPUs were found to be especially efficient in the simulation of plasma processes [1, 2].

This research was initiated by the effect of anomalous heat conductivity observed with the GOL-3 facility in the Budker Institute of Nuclear Physics [3]. The GOL-3 facility is a long open trap where the dense plasma is heated up in an intensive magnetic field during the injection of a powerful relativistic electron beam of a microsecond duration. The effect is expressed in a decrease in the plasma electron heat conductivity by 100 or 1,000 times as compared to the classical value for plasma with temperature and density observed in the experiment. An anomalous heat conductivity arises because of the turbulence that is caused by the relaxation of a relativistic electron beam in the high-temperature Maxwellian plasma. The physical problem is to define the origin and mechanism of the heat conductivity decrease. This is of great importance for the fusion devices because the effect in question contributes in heating the plasma and confining it. The problem of the heat transport in fusion devices was widely discussed (e.g. [4–6]).

The novelty of the present study has two aspects: the physical aspect and the numerical aspect. From the physical point of view, the beam heating of plasma was being considered for a long time by now, but the details of the process, namely, parameters of the arising plasma instability are still unknown. The existing theory of the beam heating uses too many approximations like a strictly monochromatic beam, the Maxwellian plasma, etc.

*Supported by RFBR under Grants 12-07-00065, 11-01-00249.

Our objective is to determine such an instability and to evaluate its parameters. The numerical aspect of the novelty is that such problems are usually solved by means of the direct Boltzmann equation solution (e.g. [7]). The PIC method is expected to give a better picture of turbulence and the underlying plasma instability, although, in order to obtain a physically realistic picture. it requires much more computer costs, as was pointed out in [4].

This problem needs a high-performance computation because of the necessity of having a large enough grid to simulate the resonance interaction of a relativistic electron beam with plasma. The beam interacts with plasma through the electric field (similar to the Landau damping), thus it is necessary to simultaneously observe two different scales. The first one is the plasma Debye length and the second is the beam-plasma interaction wavelength, which is 10 or 100 times larger than the Debye length. Since one must provide, at least, 8 grid cells at the Debye length, it is possible to estimate the size of the grid.

It is also necessary to provide a large number of superparticles for each cell of the grid for the simulation of turbulence. The level of non-physical statistical fluctuations is inversely proportional to the number of superparticles per cell. So, with a very small number of superparticles, all the physical plasma waves and oscillations will be suppressed by the non-physical noise.

This research also aimed at a more efficient use of cluster supercomputers. In order to attain this objective it is necessary to evaluate their performance with some programs for solving real physical problems. It would be incorrect to restrict the performance testing to the general purpose tests like the LinPack package since there is a difference between the cluster performance of declared in Top 500 list [8] (either the peak or the LinPack performance) and the performance attained by a particular user with his particular program. For example, if one employs one fourth of a cluster with the peak performance of 5.4 Tflop/s (1 Tflop/s is 10^{12} floating point operations per second), the peak performance should be about 1 Tflop/s. But in fact, with the above program, simulating the beam relaxation in plasma, only 0.18 Tflop/s was obtained. A difference between the declared and the attained performance is not only because of the features of the program, but also because cluster systems do not fit for this kind of a problem.

2. Model description

2.1. Basic equations. The mathematical model employed for solving of the problem of beam relaxation in plasma consists of the Vlasov equations for ion and electron components of plasma and, also, of the Maxwell equation system. These equations have the following form:

$$
\begin{aligned}
\frac{\partial f_{i,e}}{\partial t} + \vec{v}\frac{\partial f_{i,e}}{\partial \vec{r}} + \vec{F}_{i,e}\frac{\partial f_{i,e}}{\partial \vec{p}} = 0, &\qquad \vec{F}_{i,e} = q_{i,e}\Big(\vec{E} + \frac{1}{c}[\vec{v},\vec{B}]\Big),\\
\operatorname{rot}\vec{B} = \frac{4\pi}{c}\vec{j} + \frac{1}{c}\frac{\partial \vec{E}}{\partial t}, &\qquad \operatorname{div}\vec{B} = 0,\\
\operatorname{rot}\vec{E} = -\frac{1}{c}\frac{\partial \vec{B}}{\partial t}, &\qquad \operatorname{div}\vec{E} = 4\pi\rho.
\end{aligned}
\tag{1}
$$

In the present study, this equation system is solved by the method described in [10]. All the equations will be further given in the non-dimensional form. The following basic quantities are used for the transition to the dimensionless form:

- characteristic velocity is the light speed $\tilde{v} = c = 3 \times 10^{10}$ cm/s;
- characteristic plasma density $\tilde{n} = 10^{14}$ cm^{-3}; and
- characteristic time $\tilde{t}$ is the plasma period (the value inverse to the electron plasma frequency) $\tilde{t} = \omega_p^{-1} = \left(\frac{4\pi n_0 e^2}{m_e}\right)^{-0.5} = 5.3 \times 10^{-12}$ s.

The Vlasov equations are solved by the PIC method. This method implies the solution of the equation of motion for model particles, or superparticles. The quantities with the subscript i are related to ions and those with the subscript e – to electrons:

$$
\begin{aligned}
\frac{\partial \vec{p}_e}{\partial t} &= -(\vec{E} + [\vec{v}_e,\vec{B}]), \qquad \frac{\partial \vec{p}_i}{\partial t} = \kappa(\vec{E} + [\vec{v}_i,\vec{B}]),\\
\frac{\partial \vec{r}_{i,e}}{\partial t} &= \vec{v}_{i,e}, \qquad \kappa = \frac{m_e}{m_i}, \qquad \vec{p}_{i,e} = \gamma\vec{v}_{i,e}, \qquad \gamma^{-1} = \sqrt{1-v^2}.
\end{aligned}
$$

The leapfrog scheme is employed to solve these equations as follows:

$$
\begin{aligned}
\frac{\vec{p}^{\,m+1/2}_{i,e} - \vec{p}^{\,m-1/2}_{i,e}}{\tau} &= q_i\left(\vec{E}^m + \left[\frac{\vec{v}^{\,m+1/2}_{i,e} - \vec{v}^{\,m-1/2}_{i,e}}{2}, \vec{B}^m\right]\right),\\
\frac{\vec{r}^{\,m+1}_{i,e} - \vec{r}^{\,m}_{i,e}}{\tau} &= \vec{v}^{\,m+1/2}_{i,e}.
\end{aligned}
$$

Here τ is the time step. The scheme proposed by Langdon and Lasinski is used to obtain the values of electric and magnetic fields. The scheme employs the finite difference form of the Faradey and the Ampere laws. A detailed description of the scheme can be found in [10]. The scheme gives the second order of approximation with respect to space and time.

2.2. Problem statement. The 3D computational domain has the shape of a cube with the following dimensions:

$$0 \le x \le L_X, \qquad 0 \le y \le L_Y, \qquad 0 \le z \le L_Z.$$

Within this domain there is model plasma. The model plasma particles (superparticles) are uniformly distributed within the domain. The density of plasma is set by the user as well as the electron temperature. The temperature of ions is considered to be zero. Beam electrons are also uniformly distributed along the domain. Thus, a beam is considered to be already present in plasma, and the effects that occur while the beam is entering the plasma, are beyond the scope of this study.

The superparticles simulating beam electrons differ from those simulating plasma electrons by the value of their energy. The beam electrons initially have the energy of about 1 MeV, and the plasma electrons have the energy of about 1 keV. Moreover, the beam electrons have one direction of movement strictly along the axis X, the plasma electrons having the Maxwellian velocity distribution for all the three dimensions.

There is one more difference between the superparticles simulating beam electrons and plasma electrons. They have different weights when computing the current and charge density. Let us consider the ratio of the beam density to the plasma density, α (usually α varies from 10^{-3} to 10^{-6}), then the contribution of a beam electron superparticle is α from the contribution of a plasma electron superparticle. In such a way it is possible to provide a large number of beam superparticles.

The main physical parameters of the problem under study are the following: the density and the temperature of the plasma electrons, the ratio of the beam density to the plasma density and the energy of the beam.

3. Parallel implementation

The program was parallelized by the domain decomposition method. The computational domain is divided into parts along the direction orthogonal to the direction of the beam (along the axis Y, the beam moving along the axis X). The computational grid in the whole domain is divided into equal parts (subdomains) along the axis Y. Each subdomain is assigned to a group of processors (in the case of a multicore system a single core is called a processor, since no hybrid parallelization like MPI+OpenMP is employed, just a mere MPI). Furthermore, the superparticles of each subdomain are uniformly distributed among processors of the group with no regard to their position, as is shown in Figure 1.

Every processor in the group solves the Maxwell equations in the whole subdomain and exchanges boundary values of the fields with processors assigned to the adjacent subdomains. Then the motion equations of superpar-

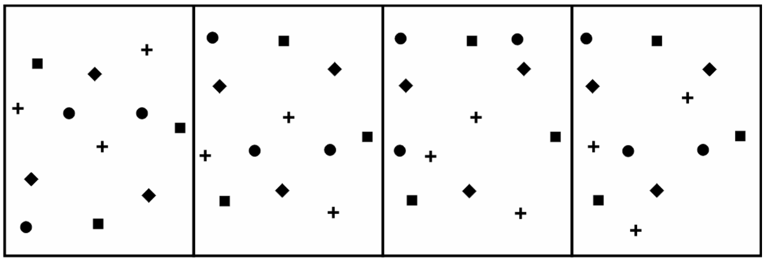

Figure 1. The scheme of domain decomposition. The computational domain is divided into 4 subdomains. The superparticles of each subdomain are distributed among four processors uniformly with no regard to their position. Different symbols (a circle, a square, a diamond, a star) denote superparticles belonging to different processors in the same subdomain

ticles are solved, and the 3D matrix of the current density and the charge density are evaluated by each processor. But since the processor has only a part of the superparticles located inside the subdomain, it is necessary to sum the matrices through all the processors of the group in order to obtain the whole current density matrix in the subdomain. The interprocessor data exchange is performed by the MPI subroutines.

3.1. Parallelization efficiency. A parallel program was primarily developed for the simulation of the beam interaction with plasma on large computational grids and with large numbers of superparticles. That is why the parallelization efficiency was computed in the following way:

$$k = \frac{T_2}{T_1} \cdot \frac{N_1}{N_2} \cdot \frac{S_2}{S_1} \cdot 100\ \%. \tag{2}$$

Here T_1 is the computation time with N_1 processors, T_2 is the computation time with N_2 processors, S is the characteristic size of the problem in each case. Here the characteristic size is the grid size along the axis X. In this section, the characteristic size S is proportional to the number of processors N. This means that the workload of a single processor is constant. The purpose of such a definition of efficiency is to find out what the communication overhead is when the number of processors is increased with a constant workload for each processor. In the ideal case, the computation time must remain the

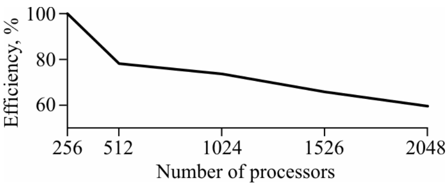

Figure 2. Parallelization efficiency measured with MVS-100K cluster. The grid size along the axes Y and Z is 64 nodes, the grid size along the axis X is equal to the number of processors, 150 superparticles per cell are used in all the cases

same (the ideal $k = 100$ %). In the computations dealing with the efficiency evaluation only the grid size of the axis X was increased, all the other parameters remaining unchanged. The results are shown in Figure 2.

3.2. Cluster performance comparison. Every time step consists of the following procedures:

- Computation of electric and magnetic fields;
- Computation of movement of superparticles;
- Evaluation of the new values of the current and the charge densities.

In addition, during the time steps selected (usually one time step from a hundred) the physical data are filed for the future analysis. The most important part of the data are the Fourier transforms of the main physical quantities (the current and the charge densities, absolute values of electric and magnetic fields).

The worktime of the above listed procedures was measured with the GnuProf (gprof) profiler. In each case, the worktime of a single procedure call is given.

The program was tested with four cluster supercomputers. The first is installed in the Tomsk State University. It has 564 2xXeon 5150 processors and is called SKIF Cyberia. In 2007, the SKIF Cyberia was number 199 in Top-500 list. The SKIF is the name of the Joint Belorussian–Russian Supercomputer Program [9]. The latter is the most powerful in Russia (by the time the profiles were obtained) and number 38 in the world MVS-100K. It is installed in the Joint Supercomputer Center and equipped with different Xeon Processors. Most of them are 4xXeon E5450, the total number of cores being 7920. The third one is called the SKIF MSU. It is installed in the Research Computing Center of the Moscow State University. It has 1,250 processors (mostly 2xXeon E5472) and it is number 103 according to the Top-500 list. The fourth cluster is installed in the Novosibirsk State University, so it will be called here the NSU cluster. It is equipped with 2xXeon 5355 processors with the total number of cores 512. A more detailed description for these clusters could be found at http://supercomputers.ru (the list of most powerful supercomputers in Russia), and for MVS-100K and SKIF MSU, also, at http://top500.org. All the clusters are equipped with similar processors, but with different network, and this results in a dramatic performance difference.

Particle movement computation and memory bandwidth. In order to evaluate new values of the position and impulse of a superparticle, it is necessary to know the values of electric and magnetic fields at the present position of the superparticle. Each of the three components of the field is

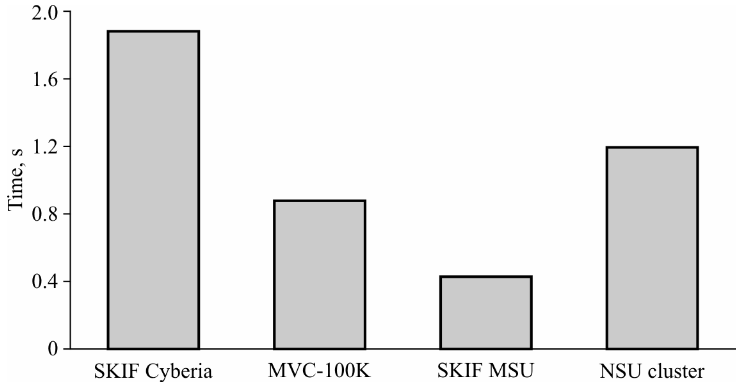

Figure 3. The worktime of the superparticle movement computation procedure

stored in a separate 3D array. In such a way six 3D arrays are accessed at each time step for each superparticle. Since superparticles are situated randomly in the subdomain, the access to the arrays is also disordered. This means that the use of the cache memory cannot reduce the computation time. If a part of the field array was fetched to the cache while computing with a superparticle, it would be impossible to use this part of the field array for the computation with the next superparticle, because it is (most likely) situated in a completely different part of the subdomain.

Since the cache memory cannot store all the six arrays for fields, one has to access the RAM for computing of the superparticle movement. And since the speed of a processor is usually much faster than the memory bandwidth, it is the memory bandwidth that determines the speed of the computation with superparticles and the performance of a program as a whole (superparticles take more than 60 % of the total time). Figure 3 presents the computation time with superparticles during one time step.

The major influence of the memory bandwidth on the superparticle computation time is confirmed by the comparison of the times obtained with the MVS-100K and the SKIF MSU clusters. Both clusters are equipped with similar processors, thus the resulting time difference (almost twofold) can only be explained by the difference in the memory bandwidth.

Figure 3 also shows that there is a possibility of optimizing the performance. One of the possible ways is to sort supersuperparticles along their position to enable the efficient use of cache memory. The computation with supersuperparticles takes most of the time (from 92 % with the SKIF Cyberia up to 64 % with the SKIF MSU). This is the part of the program that is parallelized with the highest efficiency. So, the optimization of this procedure might spoil the parallelization efficiency, but a decrease of the total time appears to be more important.

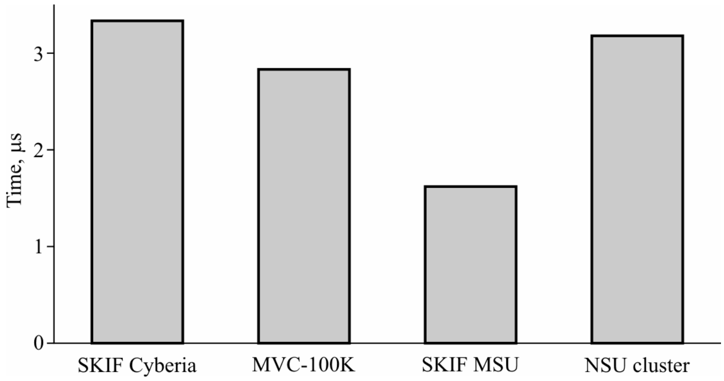

Figure 4. The worktime of 1D Fast Fourier Transform procedure

Fourier transform and computation speed. In order to separate the computation time from the memory access time, the Fourier transform procedure is considered in Figure 4. This procedure takes a 1D complex array with the size of either 512 or 64 as the input and performs the fast Fourier Transform. All the local variables of this procedure fit well in the cache memory. In such a way, with an example of this procedure one can measure the speed of the "fast" computation with no access to the RAM using only the cache memory.

Superparticle transfer and interprocessor communication speed. The speed of an interprocessor communication was measured by way of an example of the procedure implementing the transfer of supersuperparticles between processors. This procedure involves the search for supersuperparticles that have flown out of the subdomain and that are now situated in the additional buffer layer. These superparticles are then excluded from the superparticle array and put into the transfer buffer. Next the buffers are transmitted to the processors assigned to the adjacent subdomains. If the rank of the current subdomain is even, the superparticles are first transmitted "to the left", that is, to the subdomain with coordinates Y lower than in the current subdomain, and then "to the right". If the number of the current subdomain is odd, just the opposite happens—first to the right, and then to the left. Consequently, the time shown in Figure 5 includes the search through the list of superparticles and four transmitting operations MPI_SendRecv.

The number of superparticles being transmitted cannot be large due to physical considerations. In plasma physics, no large differences in density may occur within the Debye length, 10 % being a large difference in this case. The Debye length, as pointed above, must be not less than 8 grid cells for the correct wave dynamics simulation. With the grid size of $512 \times 64 \times 64$ nodes, and the domain being divided into 32 subdomains, the width of a subdomain

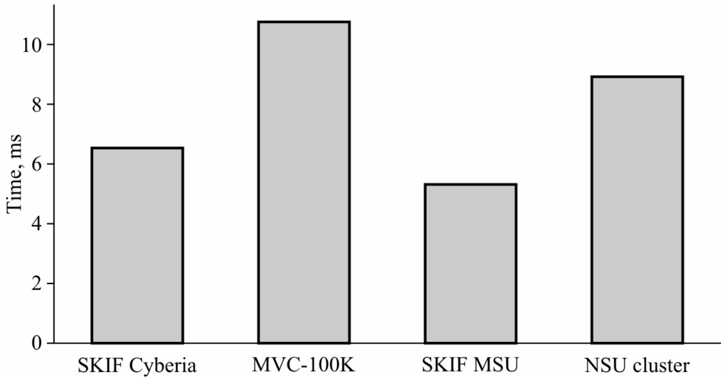

Figure 5. The worktime of the particle transfer procedure

along the axis Y is just 2 cells. Thus, the densities in adjacent subdomains may not differ by more than 10 %. This means that if a superparticle flux from one processor to another suddenly occurs to be large, the computation is physically incorrect. In such a way it is possible to set the size of the transfer buffer to 5% of the size of the superparticle array, the number of superparticles sent being usually much less.

Figure 5 shows that a minimal time for the transfer is being wasted with the SKIF Cyberia and the SKIF MSU. A possible cause is the ServNet technology installed on the SKIF machines [9]. A maximal time is wasted for the transfer with the MVS-100K. This is probably because of a large size and heterogenous nature of this supercomputer.

3.3. Computation time reduce by means of GPU computation. Particle motion computation as performed with the GeForce 9400 (16SMx8) has already shown a significant performance increase as compared to both Xeon and Nehalem processors (the table). It is necessary to note that the particle computation time includes the memory access time, thus, the par-

Particle motion computation time on GPUs as compared to CPUs. For Intel Xeon processor, the lowest time obtained with the SKIF-MGU cluster is given. The performance in flops is estimated based on the fact that 250 floating point operations are made for each particle at each time step

Computational device	Computation for 1 million particles, ms	Performance, Gflops
GeForce 9400	16	2.7
Tesla c2090	0.3	800
Xeon	204.08	1.1
Nehalem	316.32	0.79

ticle computation time drastically differs on different systems. In the same way, the advantage of GPUs is achieved due to the fast memory access.

In such a way the computation with one core of the GPU is faster than magnitude by two orders. In the general case, the present numbers are valid only for Particle-In-Cell method implementation and cannot be used for the performance evaluation of the above-mentioned systems.

3.4. Optimization of computation with GPU. In the above table, the performance given was obtained after several stages of the program optimization. The details of the GPUs in use were taken into account as well as peculiarities of particle pushing algorithms. Let us recall that the computation by the PIC method consists of the two stages: field computation and particle motion. Profiling the program has shown that the particle motion takes most of the time and requires optimization.

After the source code of the program was adopted for GPUs and debugged, the GPU performance was only 27 times as much as the CPU performance (Intel Xeon processor). The analysis of the program with the profiling tools has shown that there are a lot of cache misses for both reading and writing. There are also branches resulting in a significant slowdown and some other problems.

A large number of cache misses is due to the non-steady memory access. To remove this drawback, the particles were reordered, i.e., assigned to definite cells, so that all the particles that belong to one cell be stored in one place in the memory. It is necessary to note that sorting is not used. The reason is that the algorithm complexity for sorting is not lower than $O(N \log N)$, which is unacceptable for the PIC method (for the particle number $N \sim 10^9$). The complexity of the reordering algorithm is $O(N)$.

The use of the reordering algorithm the worktime reduces twice even for Xeon. But it does not help in avoiding cache misses completely. So, the particles are stored in the texture memory, thus greatly reducing the number of cache misses and giving additional acceleration up to 5–6 times (Figure 6).

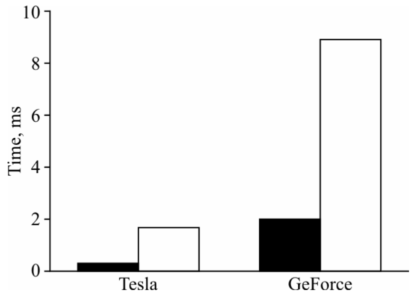

Figure 6. Runtime of the particle motion procedure on Tesla and GeForce using texture memory (the black bar) and without texture memory (the white bar)

In the course of computation, a lot of values are used that remain constant during the kernel execution. These values were moved to the constant memory.

The arithmetical operations were also optimized by replacing them with faster analogs. In all the places, where it is possible, division was replaced by multiplication, giving the performance increase of 15 %.

The use of the CUDA Occupancy Calculator has shown that the performance growth is restricted by the number of registers required by the kernel for execution. The number of required registers was reduced by rewriting the computational algorithm. In such a way, the number of registers available for the kernel was explicitly restricted at the compilation stage, resulting in 200 % acceleration.

Finally, the particle motion stage was in 600 times accelerated.

4. Conclusion

The parallel implementation of the numerical model of an interaction of the electron beam with plasma is described, and the parallel efficiency is presented. The worktime of the program has been measured on various clusters and analyzed. It was detected that the memory bandwidth of the cluster is the most important feature for the PIC programs to achieve good performance. For the program presented it has been shown that there is a good possibility for the optimization and decrease of the computation time.

The performance for supercomputers powered by both Intel Xeon processors and Nvidia Tesla GPUs is given. It is shown that particle motion stage is computed much faster with GPUs.

References

[1] Kong Xianglong, Huang Michael C., Ren Chuang, Decyk Viktor K. Particle-in-cell simulations with charge-conserving current deposition on graphic processing units // J. Computational Physics. — 2011. — Vol. 230, Iss. 4. — P. 1676–1685.

[2] Mertmann Ph., Eremin D., Mussenbrock T., Brinkmann R.P., Awakowicz P. Fine-sorting one-dimensional particle-in-cell algorithm with Monte-Carlo collisions on a graphics processing unit // Computer Physics Communications. — 2001. — Vol. 182, Iss. 10. — P. 2161–2167.

[3] Astrelin V.T., Burdakov A.V., Postupaev V.V. Generation of ion-acoustic waves and suppression of heat transport during plasma heating by an electron beam // Plasma Physics Reports. — 2010. — Vol. 24, No. 5. — P. 414–425.

[4] Cohen B.I., Barnes D.C., Dawson J.M., et al. The numerical Tokamak project: simulation of turbulent transport // Computer Physics Communications. — 1995. — Vol. 87, Iss. 1–2. — P. 1–15.

[5] Jaun A., Appert K., Vaclavik J., Villard L. Global waves in resistive and hot tokamak plasmas // Computer Physics Communications. — 1995. — Vol. 92, Iss. 2–3. — P. 153–187.

[6] Gardarein J.-L., Reichle R., Rigollet F., et al. Calculation of heat flux and evolution of equivalent thermal contact resistance of carbon deposits on Tore Supra neutralizer // Fusion Engineering and Design. — 2008. — Vol. 83, Iss. 5–6. — P. 759–765.

[7] Wright J.C., Bonoli P.T., D'Azevedo E., Brambilla M. Ultrahigh resolution simulations of mode converted ion cyclotron waves and lower hybrid waves // Computer Physics Communications — 2004. — Vol. 164, Iss. 1–3. — P. 330–335.

[8] TOP500 Supercomputing Sites. — http://www.top500.org.

[9] "SKIF" – Joint Belorussian–Russian Program. — http://www.skif.bas-net.by/index_en.htm.

[10] Vshivkov V.A., Grigoryev Yu.N.,Fedoruk M.P. Numerical "Particle-in-Cell" Methods. Theory and Applications. — Utrecht–Boston: VSP, 2002.

[11] Krall N., Trivelpiece A. Principles of Plasma Physics. — McGraw–Hill, 1973.

[12] Bidsall Ch., Langdon B. Plasma Physics via Computer Simulation. — McGraw–Hill, 1985.