Parallel Realization of the Hybrid Model Code for Numerical Simulation of Plasma Dynamics

M A Boronina, I G Chernykh, E A Genrikh, V A Vshivkov
Institute of Computational Mathematics and Mathematical Geophysics SB RAS, Lavrentiev avenue 6, Novosibirsk, 630090, Russia
E-mail: boronina@ssd.sscc.ru

Abstract. The work is devoted to a parallel realization of a hybrid model for study of plasma dynamics in axially symmetric open magnetic traps. The model is based on MHD approximation for electron component of the plasma and on the kinetic approach for ion component. In the model the particle-in-cells method (PIC) with explicit numerical schemes on staggered grids for the cylindrical coordinate system is applied. The permanent injection of the particles, the long characteristic times, the necessary grid resolution and the conditional stability of the method required development of parallel version of the algorithm. In the parallelization we use the mixed decomposition with static load balancing. We present the parallel algorithm as well as results of the computational experiments on plasma dynamics in regime of diamagnetic confinement in an open magnetic trap.

1. Introduction
The present-day leaders in the field of devices for plasma confinement with purpose of controlled thermonuclear fusion are tokamaks. Experiments on JET tokamak yielded fusion power 16MW and fusion power gain Q around 0.65 [1]. The ratio of the plasma pressure to the magnetic field pressure, $\beta$, in tokamaks does not exceed 5-10%. This fact together with the complicated tokamak design, big sizes, high experimental costs brings into consideration open plasma traps. A new concepts of plasma confinement are proposed and needed to be verified. For example, formation of “diamagnetic bubble” allows creating a compact linear open traps for the fusion purposes [2].

Mathematical modelling is the key instrument for the physical effects analysis. The appropriate model for the plasma simulation is based on kinetic Vlasov equation for electrons and ions. However, the differences in masses of electrons and ions make this model inapplicable due to the enormously high computer requirements. Our 2D hybrid model is based on MHD approximation for electron component of the plasma and on the kinetic approach for the ion component [3]. Despite of its economy compared with fully kinetic models the parallel version of the numerical algorithm is natural necessity for the reasonable computational times. The aim of the work is the parallel realization of the hybrid model code for the simulations of the plasma flows interaction as well as testing it on example of beam injection into open magnetic trap. In the paper the algorithm description and the finite-difference schemes are written out and the results of the numerical tests are presented.
2. Problem statement

We consider a cylindrical open trap of radius $R$ and length $L$ with background hydrogen plasma with density $n_0$ at the initial time moment. Two coaxial round coils on the ends of the trap produce the magnetic field of the mirrors, the magnetic field at the center on the axis $(0, L/2)$ is $H_0$. In the same point a beam particle source is located. The question is to study the self-consistent beam-plasma interaction.

As we consider an axially symmetric case, we use cylindrical coordinate system in the two dimensional domain and eliminate derivatives $\partial/\partial \varphi$ from the equations.

For the definition of the dimensionless variables we use the characteristic values $H_0$ for the magnetic field and $n_0$ for the density. The characteristic time is $t_0 = 1/\omega_{ci}$, where $\omega_{ci} = eH_0/cm_i$ is the ion gyrofrequency. The characteristic length is $L_0 = c/\omega_{pi}$, $\omega_{pi} = \sqrt{4\pi n_0 e^2/m_i}$ is the ion plasma frequency. The characteristic velocity is Alfvén velocity $V_A = H_0/\sqrt{4\pi m_i n_0}$.

The ion dynamics can be described with the Vlasov equation, we solve it with the method of characteristics. The equations for the ion trajectories coincide with the equations of the characteristics of kinetic Vlasov equation:

$$\frac{d\vec{r}_i}{dt} = \vec{v}_i,$$

$$\frac{d\vec{v}_i}{dt} = \vec{F}_i,$$

where $\vec{r}_i, \vec{v}_i$ are the ion coordinates and the velocities,

$$\vec{F}_i = \vec{E} + [\vec{v}_i, \vec{H}] - \kappa \vec{j}/n,$$

The force $\vec{F}_i$ includes Lorentz force and the ion-electron friction force with $\kappa = cm_e/eH_0\tau_{ei}$, where $\tau_{ei}$ is the characteristic ion-electron collision time [4]. Since we consider the quasi-neutral plasma $n = n_i = n_e$. The current is

$$\vec{j} = n_i \vec{V}_i - n_e \vec{V}_e,$$

where $n_e$ is electron density, $\vec{V}_e$ is electron velocity. The ion velocity $\vec{V}_i$ and the ion density $n_i$ are defined by the ion distribution function $f_i(t, \vec{r}, \vec{v})$:

$$n_i(\vec{r}) = \int f_i(t, \vec{r}, \vec{v})d\vec{v},$$

$$\vec{V}_i(\vec{r}) = \frac{1}{n_i(\vec{r})} \int f_i(t, \vec{r}, \vec{v})\vec{v}d\vec{v}.$$ 

The massless electron component is described by the transport equation:

$$\frac{d}{dt}[\vec{V}_e, \vec{H}] - \nabla \frac{p_e}{2n} - \kappa \frac{\vec{j}}{n} = 0.$$

The electron temperature $T_e$ is defined by the heating due to the ion-electron friction, the heat flux, the changes due to the compression/expansion and obeys the equation:

$$n \left( \frac{\partial T_e}{\partial t} + (\vec{V}_e \nabla)T_e \right) = (\gamma - 1) \left( 2\kappa \frac{T_e^2}{n} + \kappa_1 \text{div} \nabla T_e - p_e \text{div} \vec{V}_e \right)$$

with $\kappa_1 = 4\pi e\lambda/H_0c$, where $p_e = n_e T_e$ is the pressure of the electrons, $\lambda$ is the thermal conductivity coefficient. The adiabatic index $\gamma = 5/3$. We consider $\kappa = \text{const}, \kappa_1 = \text{const}$. 
The electric field $\vec{E}$ and magnetic field $\vec{H}$ follow Maxwell’s equations for the case of negligible displacement currents:

$$\frac{\partial \vec{H}}{\partial t} = -\text{rot} \vec{E},$$

$$\text{rot} \vec{H} = \vec{j}.$$  \hspace{1cm} (9)

At the initial time moment the electric field $\vec{E} = 0$ and the magnetic field is defined from potential of the coils in a doubled domain with zero boundary conditions. On the present stage of the model realization we consider the perturbations of all values at the borders are negligibly small and we assume the particles reflect from the external boundaries.

3. Algorithm description

We use particle-in-cell method on staggered uniform grids to solve the equations (1–10) [5]. The values $H_r(i-1/2, k), H_z(i, k), E_r(i, k-1/2), E_z(i-1/2, k-1/2), E_r(i-1/2, k), V_c(i, k-1/2), V_e(i-1/2, k-1/2), V_a(i-1/2, k), V_p(i-1/2, k-1/2), T_r(i-1/2, k-1/2), V_p(i-1/2, k-1/2), n(i-1/2, k-1/2)$ are defined in the spatial mesh nodes. The values of particle data $r, z, v_r, v_p, v_z$ are defined on the Lagrangian mesh.

Since in equations (3) (4), (7), (8) the currents appear with a factor as $\vec{j}/n$, we substitute $\vec{U} = \vec{j}/n$ and use the arrays for $\vec{U}$ during the computations. The only equation requiring values $\vec{j}$ is (10) and we use the multiplication $n\vec{U}$ there. The result of the changes is decrease of the number of the operations performed by the processors and increase of the algorithm performance. The definitions $\vec{E}_e = -[\vec{V}_e, \vec{H}], \vec{E}_c = -\nabla p_e/2n, \vec{E} = \vec{E}_e + \vec{E}_c$ yield $\vec{E} = \vec{E} + \kappa \vec{U} = \vec{E}_e + \vec{E}_c + \kappa \vec{U}$. Here $\vec{E}_\varphi = 0$ and we use only two arrays for $\vec{E}_\varphi$. For equations (3) and (7) rewritten in terms of $\vec{E}$ we obtain the reduction of the term $k\vec{U}$ and a computation speed-up due to the use of arrays only for $\vec{E}$. We use the idea and calculate and store the array data of $\vec{E}_e$ and $\vec{E}_c$ separately for the calculation of the magnetic field $H_\varphi$ using equation (9). The additional arrays allow using more compact scheme template and more accurate calculation of the pressure gradient contributions into the magnetic field.

The stationary structure of the diamagnetic bubble implies attainment of the bubble saturation with the particles and further steady plasma flux in the trap, defined by the particle injection and losses. The characteristic times of the system evolution are expected to be $10^4 \omega_{ci}^{-1}$, so, for the time unit with $\sim 10^4$ steps it requires $\sim 10^8$ time steps.

Since the solution accuracy is determined by the grid resolution, the bubble boundary layer of the size $\lambda_b \sim 1$ cm has to be described with minimum 10 grid nodes. The bubble sizes $10 \times 100$ cm require grids with $100 \times 1000$ nodes. The particle-in-cell accuracy is also defined by the number of the model particles in each cell [6], and more accurate results require bigger number of the model beam particles and model background particles. The millions of the injected particles require sufficient processor memory. The grid refinement leads to proportional increase of the number of the background ions. Besides, smaller time step must be taken for finer grids according to the stability condition for the explicit schemes. All these factors necessitate using of a parallel version of the described algorithm for both faster computations and larger data processing.

The domain is divided in equal parts and each subdomain is assigned to one of $n_p_0$ groups of processor cores. The particles of the subdomain are distributed evenly among the corresponding group cores [7]. From $n_p$ processor cores the ones with ranks $0..n_p_0 - 1$ are denominated as master cores, the exchanges of the auxiliary grid nodes are performed with the master cores. The rest of the cores are denominated as slaves, they receive the same grid data from their masters, process the particles and send the results back. The static core distribution for the particles and the domain decomposition are presented schematically in figure 1.
The block-scheme of the algorithm is presented in figure 2. Let us consider one step of the time cycle. It consists from the two stages and switching between them. On the Lagrangian stage the particle velocities and coordinates are calculated. The grid values are calculated on the Eulerian stage.

In the beginning of the cycle the master cores distribute the field values \( B, E^*, E_\varphi \) to their slaves using function MPI \(_{Bcast}\).

### 3.1. Ion motion

We use the scheme [8] for the values \( r_i, v_i \) in Cartesian coordinates, and then recalculate them in the cylindrical coordinates:

\[

v_{ir}^* = v_{ir}^m + \tau \left( \mathcal{E}_{r}^{sm} + \mathcal{E}_{r}^{m} + v_{i\varphi}^m H_z^m - v_{iz}^m H_\varphi^m \right)
\]

\[

v_{i\varphi}^* = v_{i\varphi}^m + \tau \left( \mathcal{E}_{\varphi}^{sm} + v_{iz}^m H_r^m - v_{ir}^m H_z^m \right)
\]

\[

v_{iz}^{m+1} = v_{iz}^m + \tau \left( \mathcal{E}_{z}^{sm} + \mathcal{E}_{z}^{m} + v_{ir}^m H_\varphi^m - v_{i\varphi}^m H_r^m \right),
\]

(11)
\[ z_i^{m+1} = z_i^m + \tau v_{ir}^{m+1}, \quad x = v_i^m + \tau v_{ir}, \quad y = \tau v_{i\phi}, \quad v_i^{m+1} = \sqrt{x^2 + y^2}, \] (12)

\[ \cos \alpha = z/r_i^{m+1}, \quad \sin \alpha = y/r_i^{m+1} \]

\[ v_{ir}^{m+1} = v_{i\varphi}^m \cos \alpha + v_{i\varphi}^m \sin \alpha \quad v_{i\varphi}^{m+1} = v_{i\varphi}^m \cos \alpha - v_{i\varphi}^m \sin \alpha \] (13)

Each processor core computes its own particle velocities and coordinates using bilinear interpolation for the electromagnetic fields. We form buffer arrays for the particles, which move away from their subdomain, to be sent to one of the cores of the neighbour group. The local rank of the receiving core in the neighbour group is defined as \( \text{mod}(n + 1, C_g) \), where \( C_g \) is number of cores in group \( g \). This way of particles exchanges yields a uniform particle distribution among the cores in each group.

### 3.2. Average ion velocities, densities

The switching from the Lagrangian to Eulerian stage implies that each processor core computes the average velocity in cell and density. First, the square of the cell \([r_{a-1/2}, r_{a+1/2}] \times [z_{b-1/2}, z_{b+1/2}] \) is calculated:

\[ S_{cell} a = \pi r_{a+1/2}^2 - \pi r_{a-1/2}^2 = 2\pi h_r r_a, \]

and then the volume of the cell:

\[ V_{cell} a = S_{cell} a h_z = 2\pi h_r h_z r_a. \]

The density values on the spatial grid are

\[ n_{i-1/2,k-1/2} = \frac{\sum_j \left( q_j R_{CIC}(\vec{r}_{i-1/2,k-1/2}, \vec{r}_{j}) \right)}{V_{cell} i-1/2}. \] (14)

Here \( j = 1..N \) is the number of model particle, \( q_j \) is the full charge of the ion \( j \), \( R_{CIC} \) is cloud-in-cell shape-function of the particle (bilinear interpolation). Similarly, from the velocities of ions \( \vec{v}_j \) the average ion velocity is computed at the nodes of the grid:

\[ \vec{V}_{i,a,b} = \left( \frac{\sum_j q_j \vec{v}_j R_{CIC}(\vec{r}_{a,b}, \vec{r}_{j})}{n_{a,b} V_{cell} a} \right) \] (15)

Since the density is calculated at points \( i - 1/2, k - 1/2 \), for the velocity components \( V_r \) and \( V_z \) in (15) the values \( n_{i,k-1/2}, n_{i-1/2,k} \) are densities, calculated as arithmetic average from two corresponding density values.

The masters collect the processed arrays using MPIReduce from their group and then exchange with each other with the boundary grid values.

### 3.3. Currents

The further computations in the time cycle are performed only by the master cores on the spatial grids, so only the grid boundary exchanges are needed.

From Maxwell equation (10) values \( U_r, U_{\varphi}, U_z \) are determined:

\[ U_r^{m+1}_{i,k-1/2} = -\frac{2}{h_z} \frac{H_r^{m+1}_{i,k} - H_r^{m+1}_{i,k-1}}{(n_{i+1/2,k-1/2} + n_{i-1/2,k-1/2})}, \]

\[ U_{\varphi}^{m+1}_{i-1/2,k-1/2} = \frac{1}{n_{i-1/2,k-1/2}} \left( \frac{H_{\varphi}^{m+1}_{i-1/2,k} - H_{\varphi}^{m+1}_{i-1/2,k-1}}{h_z} \right) \] (16)

\[ U_z^{m+1}_{i-1/2,k} = \frac{2}{h_r} \frac{r_i H_{\varphi}^{m+1}_{i,k} - r_{i-1} H_{\varphi}^{m+1}_{i-1,k}}{r_{i-1/2}(n_{i-1/2,k+1/2} + n_{i-1/2,k-1/2})}. \]
3.4. Electron velocities
The electron velocities are defined from relation (4):
\[ \vec{V}_e = \vec{U} - \vec{V}_i. \] (17)

3.5. Electric field
Equation (7) is used to compute the electric field:
\[
E_{\varphi i-1/2,k-1/2}^{m+1} = \frac{(V_{m+1}^{i,k-1/2} + V_{m+1}^{i-1,k-1/2})}{4} \left( \frac{H^m_{z i,k-1/2} + H^m_{z i-1,k-1/2}}{2} - \frac{V_{m+1}^{i-1/2,k} + V_{m+1}^{i-1/2,k-1}}{4} \right).
\]

The values of \( E^* \) are computed according to the following schemes:
\[
E_{\varphi i-1/2,k}^* = \frac{V_{m+1}^{i-1/2,k+1/2} + V_{m+1}^{i-1/2,k-1/2}}{2} - \frac{H^m_{\varphi i-1/2,k} + H^m_{\varphi i-1,k}}{2},
\]
where the overline denotes the arithmetic average of the four values in the nearest nodes:
\[
\overline{C}_{i,k} = \frac{C_{i+1/2,k+1/2} + C_{i-1/2,k+1/2} + C_{i+1/2,k-1/2} + C_{i-1/2,k-1/2}}{4}.
\]

For the value of \( E_\nabla \) the following schemes are used:
\[
E_{\nabla i-1/2,k}^{m+1} = -\frac{n_{i+1/2,k-1/2}T_{m+1}^{i+1/2,k-1/2} - n_{i-1/2,k-1/2}T_{m}^{i-1/2,k-1/2}}{h_r(n_{i+1/2,k-1} + n_{i-1/2,k-1/2})},
\]
\[
E_{\nabla i-1/2,k}^{m+1} = -\frac{n_{i-1/2,k+1/2}T_{m+1}^{i-1/2,k+1/2} - n_{i-1/2,k-1/2}T_{m}^{i-1/2,k-1/2}}{h_z(n_{i-1/2,k+1} + n_{i-1/2,k-1/2})}.
\]

3.6. Magnetic field
The formulae for the magnetic field are of the following form:
\[
H_{r i-1/2,k}^{m+1} = H_{r i-1/2,k}^m + \frac{\tau}{h_z} \left( E_{\varphi i-1/2,k+1/2}^{m+1} - E_{\varphi i-1/2,k-1/2}^{m+1} \right) + \frac{\tau\kappa}{h_z} \left( U_{\varphi i-1/2,k+1/2}^m - U_{\varphi i-1/2,k-1/2}^m \right) - \frac{\tau}{h_z} \left( E_{\varphi i+1/2,k}^* - E_{\varphi i,k-1/2}^* \right) - \frac{2\tau\kappa}{h_z} \left( U_{\varphi i,k+1/2}^m - U_{\varphi i,k-1/2}^m \right),
\]
\[
H_{z i,k-1/2}^{m+1} = H_{z i,k-1/2}^m + \frac{\tau}{h_r} \left( E_{\varphi i-1/2,k}^* - E_{\varphi i-1/2,k}^* \right) + \frac{\tau\kappa}{h_r} \left( U_{\varphi i,k+1/2}^m - U_{\varphi i,k-1/2}^m \right) - \frac{\tau\kappa}{h_r} \left( U_{\varphi i+1/2,k}^m - U_{\varphi i-1/2,k}^m \right).
\]
Here \( G_{i,k} \) correspond to \( rotE_\varphi \) and computed from the relation:
\[
G_{i,k} = \frac{1}{4h_z h_x} \left[ \left( T_{i+1/2,k+1/2} - T_{i-1/2,k-1/2} \right) \left( n_{i+1/2,k-1/2} - n_{i-1/2,k+1/2} \right) + \left( T_{i-1/2,k+1/2} - T_{i+1/2,k-1/2} \right) \left( n_{i+1/2,k+1/2} - n_{i-1/2,k-1/2} \right) \right].
\]
3.7. Temperature

We use the up-wind scheme for the electron temperature computation:

\[
\frac{T_{e}^{m+1}}{\tau} = \frac{T_{e}^{m} - T_{e}^{m}}{\tau} + \frac{V_{er} i,k-1/2 + V_{er} i-1,k-1/2}{2} \Delta_{r} T_{e} |_{i-1/2,k-1/2} + \frac{V_{ez} i-1/2,k + V_{ez} i-1/2,k-1}{2} \Delta_{z} T_{e} |_{i-1/2,k-1/2} = 0
\]

(20)

\[
= -(\gamma - 1) T_{e} i-1/2,k-1/2 \left[ \frac{1}{r_{i-1/2}} \frac{r_{i} V_{er} i,k-1/2 - r_{i-1} V_{er} i-1,k-1/2}{h_{r}} + \frac{V_{ez} i-1/2,k - V_{ez} i-1/2,k-1}{h_{z}} \right]
\]

\[
+ 2\kappa (\gamma - 1) \left[ (U_{r} i,k-1/2 + U_{r} i-1,k-1/2)^2 + (U_{e r} i-1/2,k-1/2)^2 + (U_{e z} i-1/2,k + U_{e z} i-1/2,k-1)^2 \right] +
\]

\[
+ \frac{\kappa_{1} (\gamma - 1)}{n_{e} i-1/2,k-1/2} \left[ \frac{r_{i} (T_{e} i+1/2,k-1/2 - T_{e} i-1/2,k-1/2) - r_{i-1} (T_{e} i-1/2,k-1/2 - T_{e} i-3/2,k-1/2)}{h_{r}^2 r_{i-1/2}} + \frac{T_{e} i-1/2,k+1/2 - 2T_{e} i-1/2,k-1/2 - T_{e} i-1/2,k-3/2}{h_{z}^2} \right].
\]

(21)

\[
\Delta_{r} T_{e} |_{i-1/2,k-1/2} = \begin{cases} 
\frac{T_{e} i-1/2,k-1/2 - T_{e} i-3/2,k-1/2}{h_{r}^2}, & \text{if } V_{er} i,k-1/2 + V_{er} i-1,k-1/2 \leq 0 \\
\frac{T_{e} i+1/2,k-1/2 - T_{e} i-1/2,k-1/2}{h_{r}^2}, & \text{if } V_{er} i,k-1/2 + V_{er} i-1,k-1/2 > 0
\end{cases}
\]

(22)

\[
\Delta_{z} T_{e} |_{i-1/2,k-1/2} = \begin{cases} 
\frac{T_{e} i-1/2,k-1/2 - T_{e} i-2,k-3/2}{h_{r}^2}, & \text{if } V_{ez} i-1/2,k + V_{ez} i-1/2,k-1 \leq 0 \\
\frac{T_{e} i+1/2,k-1/2 - T_{e} i-1/2,k-1/2}{h_{r}^2}, & \text{if } V_{ez} i-1/2,k + V_{ez} i-1/2,k-1 > 0
\end{cases}
\]

For the computations (3.3-3.7) only exchanges with the auxiliary grid nodes between the master cores are required after each value computation. The magnetic fields on the boundary (1, k) are computed from the scheme and do not require the exchanges with the right neighbour.

4. Numerical experiments

In the numerical tests we consider the computational domain sizes R=6, L=30. The magnetic field on the axis $H(0,L/2) = 0.1$, the mirror ratio is 3.2. The beam density is $10^9$, the injecting ion speed is 0.1, the ion temperature is 0.01. The coefficient of electrical conductivity is 0.2, the coefficient of thermal conductivity is k=0.1. These parameters correspond to the dimensional values $R = 137 cm$, $L = 684 cm$, $H(0,L/2) = 0.2 kG$. The dimensional background ion density is $10^{12} cm^{-3}$, the beam density $10^{14} cm^{-3}$, the average beam speed $4.3 \cdot 10^7 cm/sec$, the ion temperature 10eV. The normalizing parameters are: $H_0 = 2 kG$, $n_0 = 10^{12} cm^{-3}$, $V_0 = 4.3 \cdot 10^8 cm/sec$, $L_0 = 23 cm$, $t_0 = 5 \cdot 10^{-8} sec$.

In figure 3 and figure 4 the absolute values of the magnetic field at the initial time moment and at t=40 are presented. The simulation results demonstrate formation of the magnetic cavern with high field gradients on its boundary.
Figure 3. The magnitude of the magnetic field $|B|$ at $t = 0$.

Figure 4. The magnitude of the magnetic field $|B|$ at $t = 40$.

Figure 5 and figure 6 demonstrate the particle velocities $(V_z, V_r)$ for the time moments $t = 10$ and $t = 40$. The beam ions are marked with black colour, the background ions are marked with red. The initially motionless background ions and the injecting beam ions increase their speeds, and later concentrate on the beam front forming background plasma cavern. The decrease of

Figure 5. The particle velocities at $t = 10$.

Figure 6. The particle velocities at $t = 40$. 
the radial velocities of the beam ions with time and the evolution observed make possible the theoretically expected plasma confinement due to the diamagnetic bubble regime.

For the numerical experiments spatial grid with $N_r = 60$ and $N_z = 300$ nodes and $N_\tau = 10^5$ time steps $\tau = 5 \cdot 10^{-4}$ were used. The number of background ions used $J_b = 7.2 \cdot 10^4$, the total number of ions $J_p = 3.4 \cdot 10^6$. The computations took $\sim 9.5$ hours.

For the grids $N_r = 30$, $N_z = 150$ and time step $\tau = 8 \cdot 10^{-4}$ from 10 to 72 processor cores in 10..30 groups were used. We denote the core distribution as $np_0 - np$. The two central groups have maximal number of cores, for example, for the case of $10 - 30$ the number is $C_4 = C_5 = 5$. The number $C_g$ falls linearly with the factor $\delta$ as the number of group $g$ approaches the domain ends: $C_{g\pm i} = \pm i\delta + C_g$, where $c$ is the number of one or two central groups. For the case 10 – 30 the number of cores falls according to $\delta = 1$: $C_3 = C_6 = 4$, etc. Another example is distribution 15 – 30, the central group has $C_7 = 8$ cores and $\delta = 2$: $C_6 = C_8 = 4$, $C_5 = C_9 = 2$. The remote groups have 1 core.

In figure 7 and figure 8 the average numbers of particles $J_c = J_g/C_g$ in a core of a group for the time moments $t = 40$ and $t = 100$ are shown. Here we normalize the groups number $g$ on $np_0$ and stretch the value along the longitudinal coordinate $z$ to compare the data for different number of groups $np_0$.

At the beginning of the injection (figure 7) the central cores contain the majority of particles and force the other cores to stand idle increasing the computational time. The cores of

![Figure 7](image1.png)  
**Figure 7.** The average number of particles $J_c$ in core at $t = 40$.

![Figure 8](image2.png)  
**Figure 8.** The average number of particles $J_c$ in core at $t = 100$. 
distribution $15 - 30$ are loaded almost uniformly at $t=80$ due to the highest number of central cores. However, further increase of the beam size leads an effective distribution to the one with the underloaded central cores and the overloaded periphery cores (see $10 - 14, 15 - 30, 30 - 72$ in figure 8). The core distribution $30 - 72$ is most effective as it has high number of cores in the center: $C_7 = 8, \delta = 1$ and smallest of all considered subdomains with particles. The configuration $10 - 14$ and $10 - 16$ have the same number of central cores $C_4 = C_5 = 4$, but higher $\delta = 2$ for $np = 16$ corresponds to an additional core in groups $g = 3$ and $g = 6$. The more uniform core load yields decreasing of the computation times for $3\%$ for $T = 100$. The linear decomposition $np_0 = np$ is the most inefficient case, but bigger number of processors $np_0$ allows processing smaller domains and, as consequence, smaller number of particles, providing faster computations.

The computation times measured in seconds for the evolution time $T = 1.25 \cdot 10^5 \tau$ are presented in Tab. 1. The table demonstrates the strong dependence of the computational time on the particle processing. The calculation of the velocities, coordinates and currents takes $\sim 98\%$ of the total time $T_{total}$. The largest subdomain with the biggest number of particles in core $J_c$ corresponds to the distribution $10 - 10$ and the longest computations. The minimal number of particles $J_c$ corresponds to the configuration $30 - 72$ and yields the best results. Since number of the grid nodes is smaller then the number of particles, the computations on the Eulerian stage take significantly smaller time (less than $1\%$ of $T_{total}$). The spatial grid size is slowly growing parameter in the simulations, whereas the number of particles is defined by the injection intensity and the evolution time $T$. In the present simulations the particle exchanges take $\sim 10^{-4} T_{total}$ or less, but for $10^3 \omega_{ci}^{-1}$ these times must be taken into account. The times for 2D array exchanges grow with the maximal number of cores in group. A certain distribution within the given $np$ cores may significantly accelerate the computations, for example, for $np = 30$ the most effective configuration in the table corresponds to $np_0 = 10$ and requires about 5 times smaller computation time than for $10 - 10$. The test simulations demonstrate the mixed decomposition advantages, but a dynamical load balancing may considerably improve it.

The computations were performed on Intel Xeon Phi 7290 processors of Siberian Supercomputer Center cluster (ICM&MG SB RAS, Novosibirsk).

5. Conclusion
A 2D hybrid model realization for study of plasma dynamics in open magnetic systems is presented. The basis of the code is a model with MHD approach for the electron component of the plasma and the kinetic approach for the ions. The parallel PIC code for the cylindrical coordinate system combines domain and particle decomposition. The numerical experiments were performed and the results correspond to the theoretical expectations. The mixed
decomposition is a powerful method for acceleration of the computation times, and appropriate core distribution may significantly decrease the times. For the problems with highly non-uniform particle distribution in time a dynamic load balancing is required for efficient computations.

Acknowledgements
The work is supported by Russian Science Foundation (project N 19-71-20026).

References
[1] Syme D, Popovichev S, Conroy S, Lengar I and Snoj L 2011 Nuclear Engineering and Design 246 185–190
[2] Beklemishev A 2016 Physics of Plasmas 23 082506
[3] Vshivkova L and Dudnikova G 2017 Hybrid model of particle acceleration on a shock wave front Numerical Analysis and Its Applications: 6th International Conference, NAA 2016, Revised Selected Papers pp 737–743
[4] Braginskii S 1965 Reviews of Plasma Physics 1 205–311
[5] Birdsall C and Langdon A 1985 Plasma physics via computer simulation (McGraw-Hill)
[6] Lotov K V, Timofeev I V, Snytnikov A and Vshivkov V 2014 Physics of Plasmas 22
[7] Boronina M and Vshivkov V 2015 Journal of Plasma Physics 81 495810605
[8] Boris J 1970 Proc. Fourth Conference on Numerical Simulations of Plasmas, Washington D.C., USA 3–67