TRANSACTIONS OF THE INSTITUTE OF AVIATION
No. 3(248), pp. 24-45, Warsaw 2017

ISSN 0509-6669
eISSN 2300-5408

PARTICLE-IN-CELL ELECTROSTATIC NUMERICAL ALGORITHM

Wojciech Konior*
* Institute of Aviation, Aleja Krakowska 110/114, 02-256 Warsaw, Poland
wojciech.konior@ilot.edu.pl

Abstract

Existing global models of interaction between the solar wind (SW) and the local interstellar medium (LISM) describe the heliosphere that arises as a result of this interaction. There is a strong motivation to develop a kinetic model using the Particle-in-Cell (PIC) method to describe phenomena which appear in the heliosphere. This is however a long term scientific goal. This paper describes an electrostatic Particle-in-Cell numerical model developed in the Institute of Aviation in Warsaw, which includes mechanical and charge exchange collisions between particles in the probabilistic manner using Direct Simulation Monte Carlo method. This is the first step into developing simulations of the heliosphere incorporating kinetic effects in collisionless plasmas. In this paper we focus only on presenting the work, which have been done on the numerical PIC algorithm.
Keywords: plasma, neutral particles, Particle-in-Cell, Monte-Carlo Method.

1. INTRODUCTION

Until now, theoretical [1-3] as well as numerical models have been developed which utilize hydrodynamical (HD) [e.g. 4-6] or magnetohydrodynamical (MHD) [e.g.7-11] description of two supersonically counter-flowing SW and LISM plasmas (resulting as the heliosphere), in combination with HD [5] or kinetic description [11] of LISM neutral component or taking into account such aspects of the SW-LISM interaction as e.g. mixing of both plasmas [12]. The fluid description may provide useful approximation and answers to some questions, but in collisionless plasmas kinetic effects may introduce some, even quite significant deviations in comparison with fluid approach. Therefore kinetic models are necessary, not only in description of LISM neutral component, but also in both plasms description. Developing the full kinetic PIC model of the SW-LISM interaction is a very ambitious long term scientific goal. There are some approaches that can be conceptually considered as something between fluid description and kinetic models e.g. gyrokinetics or kinetic MHD [13-14] and can help on achieving the general long term scientific goal, but in the frame of this work we want to get rid of the approximations assuming medium as a continuum.

There are several approaches of kinetically solving the dynamics of medium, which consists of charged and neutral particles, interacting with each other, without collisions due to very low densities, through electromagnetic forces (ions and electrons) or through binary interactions between charged and neutral particles. The first method is a direct computation of the interaction between each particle and the second one is a description of plasma (only charged particles) kinetics with Vlasov equation, which could be solved by many methods, e.g. computing the distribution function on a grid in velocity space [15] or PIC. For this study the PIC method have been chosen because of the possibility of tracking individual particles or macroparticles in continuous phase space, along with the Direct Simulation Monte Carlo (DSMC) method which in probabilistic manner allow direct binary particle-particle interactions between charged particles and neutrals (e.g. charge exchange), which could be described by fluid models also, but with a big approximation. In the direct method (Fig. 1), which is ineffective for very large systems, the number of computations is approximately proportional to the number of particles squared ($\sim 100\, N_{part}^2$). That means, for example, that with one hundred millions particles (10^8) and the computational power of one hundred teraflop per second $100\, Tflop/s$, one time step will last approximately 3 hours [16]. In comparison, in the PIC method (Fig. 2) the number of computations is about $\sim \left(2\, N_{part} + N_{cells}\right)$, which gives us for 10^8 particles, ~ 260000 cells and $100\, Tflop/s$ of computational power, and a time of one step of about $\sim 0.3s$ [16]. Such improved speed in the PIC method motivates its employment in computations of flow dynamics of rarefied gases with partially or fully ionized components, where collisions may be neglected. This method is used to solve some sort of partial differential equations, especially in plasma physics. In PIC the velocity and position for each particle are tracked, while at the same time densities and currents are computed on domain grid nodes. Equations of motion for particles, as well as field equations in mesh points are solved simultaneously. The method includes typically integration of the motion equations, scattering charge density to the field mesh, integration of the field equations on the grid nodes and weighting fields quantities from the mesh to the particles located in it. More details about PIC can be find in the next paragraph and chapter 2.

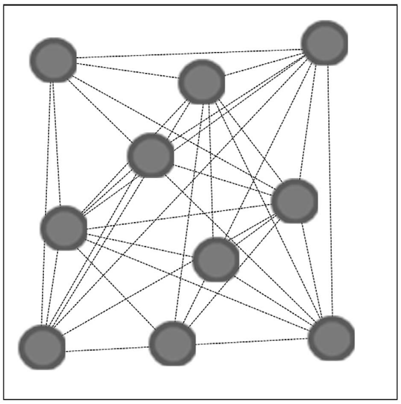
Fig. 1. Direct kinetic method (N-body simulation, dashed lines symbolize interactions between particles).

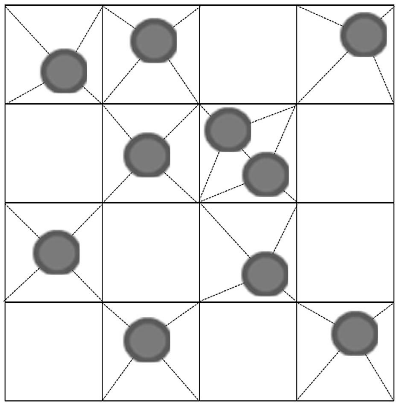
Fig. 2. Particle-in-Cell method with linear charge deposition (dashed lines symbolize interaction between particles and the field.

The computational cycle of the full electromagnetic Particle-in-Cell (EM-PIC) method is as follows: (1) according to the particle positions *($\boldsymbol{x}_n$)*, their charges and velocities (q_n, $\boldsymbol{u}_n$) are scattered to grid nodes *(i,j,k)* of the cell (weighting) in which they are located, resulting in distribution of the charge and electrical current density *(ρ, $\boldsymbol{j}$)$_{i,j,k}$* over the whole grid; (2) the integration of field equations on each grid node, which is essentially solving Maxwell's equations for electric field intensity $\boldsymbol{E}$ and magnetic field induction $\boldsymbol{B}$; (3) Lorentz's forces $\boldsymbol{F}_n$ for all particles are computed from the values of the field electric intensity and magnetic induction *($\boldsymbol{E}_{i,j,k}$, $\boldsymbol{B}_{i,j,k}$)*, in each cell; (4) integration of the equations of motion, in this time step, to obtain new velocities *($\boldsymbol{u}_n$)* of particles and pushing them into new positions *($\boldsymbol{x}_n$)*. Here, the indexes *(i,j,k)* determine grid nodes and index *n* determines particles. The cycle repeats until preset time of the simulation is achieved (see Fig. 3).

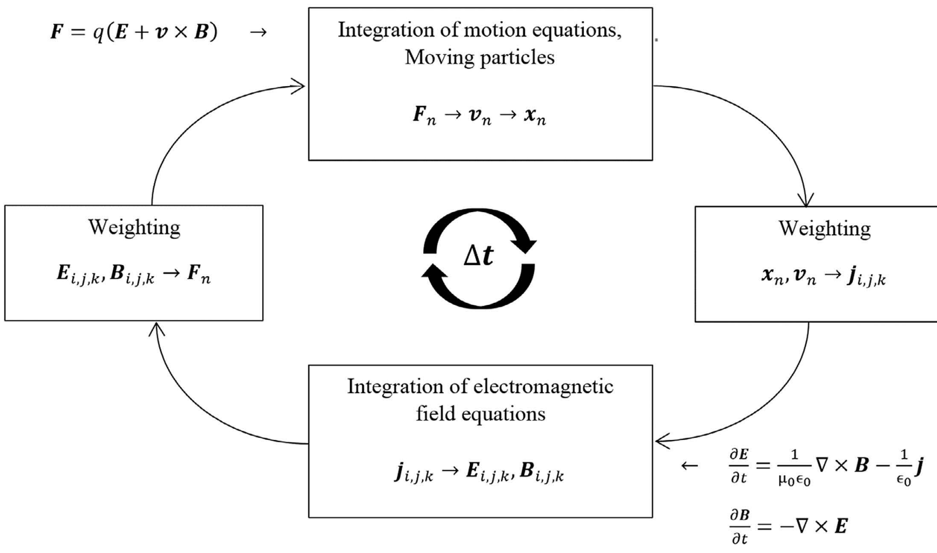
$\boldsymbol{F} = q(\boldsymbol{E} + \boldsymbol{v} \times \boldsymbol{B})$ →
Integration of motion equations, Moving particles
$\boldsymbol{F}_n \to \boldsymbol{v}_n \to \boldsymbol{x}_n$
Weighting
$\boldsymbol{E}_{i,j,k}, \boldsymbol{B}_{i,j,k} \to \boldsymbol{F}_n$
Δt
Weighting
$\boldsymbol{x}_n, \boldsymbol{v}_n \to \boldsymbol{j}_{i,j,k}$
Integration of electromagnetic field equations
$\boldsymbol{j}_{i,j,k} \to \boldsymbol{E}_{i,j,k}, \boldsymbol{B}_{i,j,k}$
← $\frac{\partial \boldsymbol{E}}{\partial t} = \frac{1}{\mu_0 \epsilon_0} \nabla \times \boldsymbol{B} - \frac{1}{\epsilon_0} \boldsymbol{j}$
$\frac{\partial \boldsymbol{B}}{\partial t} = -\nabla \times \boldsymbol{E}$

Fig. 3. A simple and straightforward EM-PIC scheme [17].

The most time-consuming operations in the PIC method are those which operate directly on particles. It means that weighting the charge and current densities, weighting the electric and magnetic field intensities and solving the motion of each particle takes most of the computing time. Nevertheless, the advantage from solving field equations on the grid is so significant that the method might be used in modeling all physical phenomena, including particles and interactions between them. In the frame of this work the PIC model of interaction between two counter-flowing streams of mixtures of charged and neutral particles was developed including mechanical and charge exchange collisions. Also electric effects were included into algorithm. Further works assumes incorporating magnetic effects and other particle interactions as electron impact and photoionization. In the next section, the PIC algorithm for simple electrostatic problems is described.

2. ELECTROSTATIC PIC ALGORITHM

The motivation of developing new fully kinetic code to solving plasma physics oriented to SW-LISM interaction is lack of such code which can operate in three dimensions, simulating globally processes occurring in the heliosphere, in which we can freely include variety of physical phenomena and numerical methods to solve them, optimized to work on the specific cluster of computers using Message Passing Interface (MPI) and Compute Unified Device Architecture (CUDA) technologies to increase the speed of computations. It was decided by authors that modifying existing codes to meet the all requirements will be too much time consuming.

The first step of developing a hybrid-kinetic model for heliospheric processes was building a model with no time-variant electric fields and with no magnetic fields. At this step numerical methods of solving the Poisson equation were intended to be applied in order to test electrostatic PIC cycle algorithm in its simplest form (Fig. 4). In this particular case, the following electrostatic seven-step PIC scheme was utilized:

I. Program initialization: simulation parameters setup, definition of the geometry of the model, grid parameters as well as initial and boundary conditions determination and particle generating.
II. Particles weighting to the grid: charge density computation in each cell and scattering the values to the grid nodes.
III. Electric field computation: solving Poisson equation for electric potential and computing potential gradient for **E**.
IV. Electric field weighting to particles: computing the electric force acting on each particle in a cell, from the field values in adjacent nodes in the cell.
V. Pushing particles: integration of equations of motion for each particle, updating velocities and moving particles to new locations.
VI. Additional operations: generating new particles, removal of those particles which left the domain and performing charge exchange and mechanical collisions.
VII. Outputting the results, moving to the next iteration (going to step II.) or ending the program.

The program runs until stationary state of global density is achieved and this condition should be fulfilled before preset number of time steps is gained.

In step (I.) there are definitions of constants and variables, also physical and numerical. Further steps are described in the next subsections. It is worth to mention that two kinds of boundary conditions (BC) were applied, i.e. Dirichlet, for those nodes which are required to have a fixed electric potential, and Neumann, for those which are assumed to have the same potential as the adjacent nodes normal to the boundary face. The core of the entire electrostatic solver is the solution of Poisson equation for electric potential:

$$\nabla^2\phi = b_{rhs} \tag{1}$$

where: ϕ – solution of partial differential equation, b_{rhs} – right hand side of the equation.

After getting potential electric field and electric forces are computed to push particles to new positions (for details see chapter 2.2-2.8). The full electrostatic PIC (ES-PIC) cycle is shown on Fig. 4. In electromagnetic problem the solver is more complicated and allows to compute not only electric but also magnetic forces. This is the main difference between electrostatic and electromagnetic code (which is not the topic of this work).

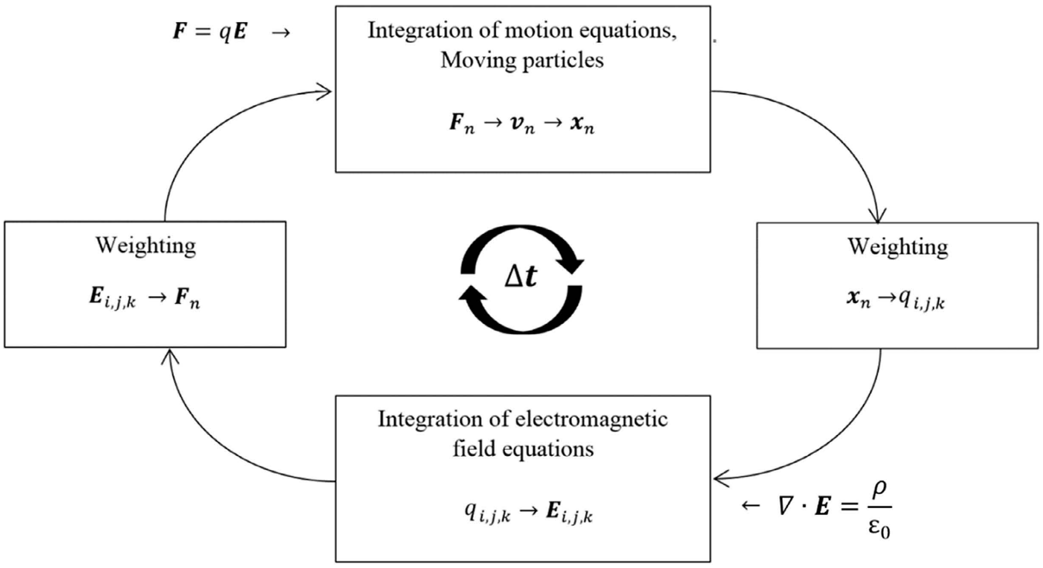
$\boldsymbol{F} = q\boldsymbol{E} \rightarrow$
Integration of motion equations, Moving particles
$\boldsymbol{F}_n \rightarrow \boldsymbol{v}_n \rightarrow \boldsymbol{x}_n$
Weighting
$\boldsymbol{E}_{i,j,k} \rightarrow \boldsymbol{F}_n$
Δt
Weighting
$\boldsymbol{x}_n \rightarrow q_{i,j,k}$
Integration of electromagnetic field equations
$q_{i,j,k} \rightarrow \boldsymbol{E}_{i,j,k}$
$\leftarrow \nabla \cdot \boldsymbol{E} = \frac{\rho}{\varepsilon_0}$

Fig. 4. A simple and straightforward ES-PIC scheme.

2.1. Particles weighting to the grid

The step (II.) consists of a procedure for charge scattering to the grid nodes. The procedure described here is for the case of a two-dimensional grid. A particular cell, containing a group of particles, is a member of the entire domain field. Every particle in that cell needs to be scattered to the grid nodes in terms of its charge. For example, let us perform a linear weighting of only one particle of charge q_p, localized inside a two-dimensional cell in point (x_n, y_n). Now let us index each node of this cell: $(i, j),(i+1, j),(i, j+1),(i+1, j+1)$. It is worth noticing, that there could be more than one method of assigning particle charge to nodes. The first one, 0th order method, is the simplest and the least accurate, and consist in assigning the entire charge to the nearest grid node. Second one, consist in assigning partial charge, linearly approximated to all four nodes, and this method is applied in the example presented here. There are also methods of 2nd and higher orders, but here we focus only on the 1st order one. To assign a part of the charge, we need to compute a proper weight and multiply it by the charge. The method of computing weights is the following: as we divide the entire area of the cell into four smaller areas (Fig. 5), the charge is localized in one vertex of each of them, so one should noticed that each area is a part of the entire cell area.

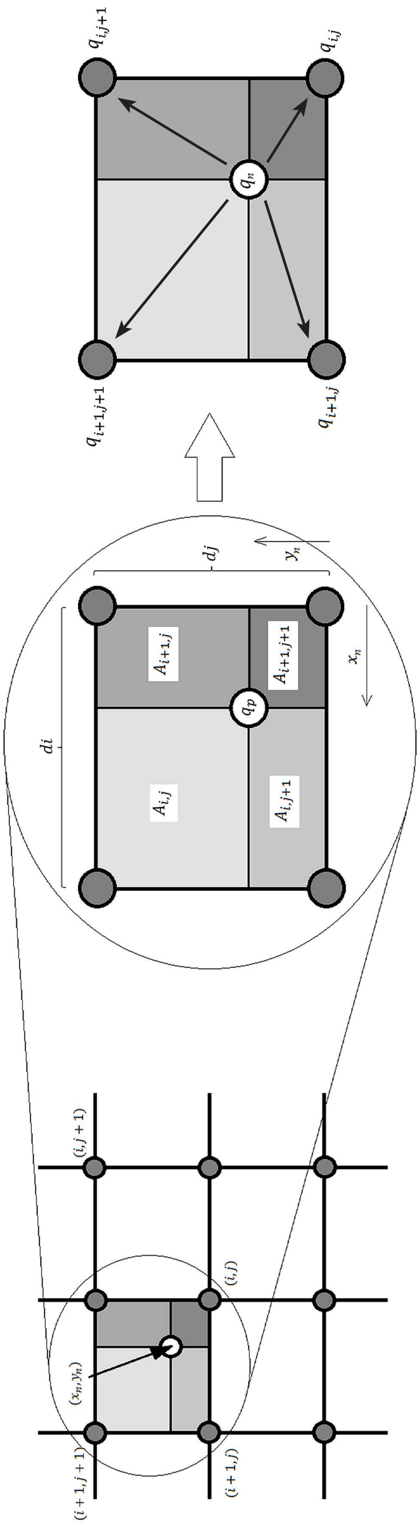

Fig. 5. Weighting scheme for 1st order interpolation method [18].

For example, let's try to assign a certain partial charge to the grid node (i,j). Let's take the opposite partial area $A_{i,j}$ (the brightest) and divide it by the whole cell area A.

$$w_{i,j} = \frac{A_{i,j}}{A} = \frac{(di - x_n)\cdot(dj - y_n)}{di \cdot dj} \tag{2}$$

where: x_n, y_n – position of particle, di, dj – width and height of the cell.

We'll get the weight $w_{i,j}$ of a value between 0 and 1. Multiplying the obtained value by the particle charge, we'll get the partial charge $q_{i,j}$, which should be assigned to the (i,j) node.

$$q_{i,j} = w_{i,j} \cdot q_n \tag{3}$$

where: q_n – particle charge.

The same should be done for the $(i+1,j),(i,j+1),(i+1,j+1)$ nodes.

$$\begin{aligned}
w_{i+1,j} &= \frac{A_{i+1,j}}{A} = \frac{x_n\cdot(dj - y_n)}{di \cdot dj} &\rightarrow\quad q_{i+1,j} &= w_{i+1,j}\cdot q_n \\
w_{i,j+1} &= \frac{A_{i,j+1}}{A} = \frac{(di - x_n)\cdot y_n}{di \cdot dj} &\rightarrow\quad q_{i,j+1} &= w_{i,j+1}\cdot q_n \\
w_{i+1,j+1} &= \frac{A_{i+1,j+1}}{A} = \frac{x_n\cdot y_n}{di \cdot dj} &\rightarrow\quad q_{i+1,j+1} &= w_{i+1,j+1}\cdot q_n
\end{aligned} \tag{4}$$

The sum of all four weights should be equal to 1. If the number of particles inside this cell is more than one, then this procedure should be performed for each particle, and its resulting weighted partial charge should be added to the particular node.

$$q_{i,j}^{tot} = \sum_{n=1}^{N} q_{i,j}^{n} \tag{5}$$

where: $q_{i,j}^{n}$ – n-th particle partial charge at (i, j) node, N – number of particles in cell.

This means that each cell node contains the sum of weighted partial charges from each particle in this cell. The same process needs to be done for all the cells in the domain. Charge density is computed by division of net charge in a particular node by the cell area (in 2D case) and by the cell volume V (in 3D case).

$$\rho_{i,j} = \frac{q_{i,j}}{A} \leftarrow 2D \qquad\qquad \rho_{i,j,k} = \frac{q_{i,j,k}}{V_{i,j,k}} \leftarrow 3D \tag{6}$$

The method described above is perfectly applicable in a three-dimensional domain. The only difference is due to the introduction of a 3rd dimension. Then instead of a rectangular cell, there is a cubic cell with depth dk, and each particle has an additional positon coordinate z_p. During computing weights partial volumes $V_{i,j,k}$ must be considered instead of areas, and there are eight nodes per cell.

2.2. Poisson equation

In the step (III.) the equation of the field quantity must be solved. In our case we need to derive a relation only with the electric potential i.e. Poisson equation. The motion of charged particles in absence of a magnetic field ($\boldsymbol{B} = 0$) is governed only by the electric field $\boldsymbol{E}$. This is expressed through the following Lorentz's force reduction [19]:

$$\boldsymbol{F} = q\left(\boldsymbol{E} + \boldsymbol{v} \times \boldsymbol{B}\right) \quad \rightarrow \quad \boldsymbol{F} = q\boldsymbol{E} \tag{7}$$

where: q – electric charge, $\boldsymbol{E}$ – electric field vector, $\boldsymbol{B}$ – magnetic induction vector, $\boldsymbol{v}$ – particle velocity vector.

In terms of electric and magnetic potential, the electric field formula reduces as follows [19]:

$$\boldsymbol{E} = -\nabla\phi - \frac{\partial \boldsymbol{A}}{\partial t} \quad \rightarrow \quad \boldsymbol{E} = -\nabla\phi \tag{8}$$

where: ϕ – electric potential, $\boldsymbol{A}$ – magnetic vector potential.

Assuming a stationary electric field, it can be stated that the motion of charged particles complies only with Gauss' law for electricity.

$$\nabla \cdot \boldsymbol{E} = \frac{\rho}{\varepsilon_0} \tag{9}$$

where: ρ – total charge density, ε_0 – vacuum permittivity.

Differentiating (8) and combining with (9) it can be obtained the Poisson equation [19].

$$\nabla^2\phi = -\frac{\rho}{\varepsilon_0} \tag{10}$$

Solving (10) for potential, the electric field can be simply obtained by differentiating solution of Poisson equation and taking it with a negative sign. For simple one-dimensional problems, equation (10) can be solved analytically, but for more complex three-dimensional geometry a numerical method of solution must be introduced. Let's consider geometry of some cuboid (Fig. 6). Firstly, the domain should be discretized in order to transfer it from infinite small differences to finite ones, then to numerically indexed. Let's assume the domain is divided into n_i, n_j, n_k cells in x, y, z directions accordingly (Fig. 6). Then let's index every node location using the i, j, k indexes.

$$dx \rightarrow di = x_{i+1,j,k} - x_{i,j,k}\ , \quad dy \rightarrow dj = x_{i,j+1,k} - x_{i,j,k}\ , \quad dz \rightarrow dk = x_{i,j,k+1} - x_{i,j,k} \tag{11}$$

where: di, dj, dk – numerical increments corresponding to those physical dx, dy, dz in x-, y-, z-direction accordingly, $x_{i,j,k}$ – position of *(i, j, k)*-th node.

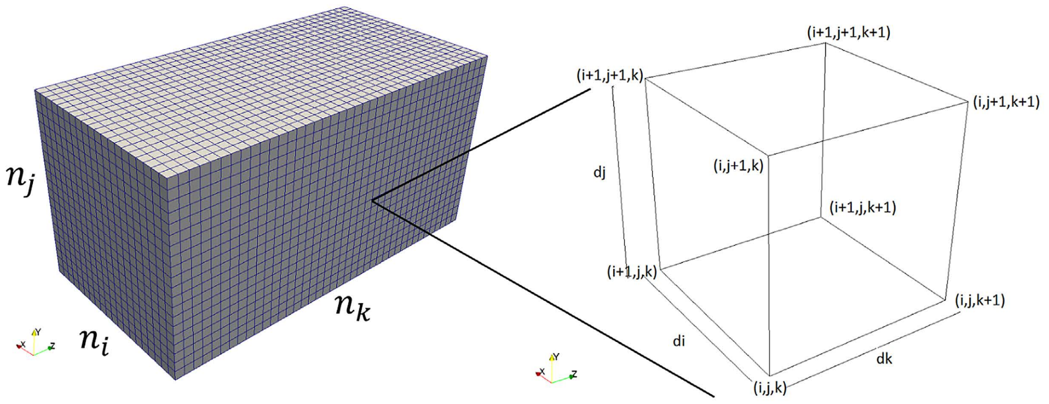

Fig. 6. Example of domain discretized into 20x20x40 cells and indexing of the particular cell (i,j,k).

Now, the Poisson equation (10) can be formulated in terms of numerical coordinates using central difference scheme:

$$\frac{\phi_{i-1,j,k}-2\phi_{i,j,k}+\phi_{i+1,j,k}}{(di)^2}+\frac{\phi_{i,j-1,k}-2\phi_{i,j,k}+\phi_{i,j+1,k}}{(dj)^2}+\frac{\phi_{i,j,k-1}-2\phi_{i,j,k}+\phi_{i,j,k+1}}{(dk)^2}=-\frac{\rho_{i,j,k}}{\varepsilon_0} \qquad (12)$$

where: $\phi_{i,j,k}$ – potential at *(i, j, k)* node, $\rho_{i,j,k}$ – total charge density at *(i, j, k)* node.

Equation (12) might be solved numerically. In the following subsections the methods for composing and solving nonlinear systems of equations are going to be described.

2.3. Poisson equation as algebraic system of equations

In sections 2.3 – 2.6 the procedure of numerically solving the Poisson equation is shown. This section shows how to compose a nonlinear system, which can be solved by chosen method. The equation $\nabla^2\phi = b_{rhs}$, in a discrete domain, is actually a system of algebraic equations in the form of:

$$M\vec{\phi}=\vec{B} \qquad (13)$$

where: M – square, band matrix of $n_u \times n_u$ size, $\vec{\phi}$ – vector of potentials of n_u size, $\vec{B}$ – vector of right sides of n_u size, $n_u = n_i \cdot n_j \cdot n_k$ – total number of domain grid nodes.

In (13), matrix M is composed of coefficients beside indexed potentials, in each equation (12). The vector $\vec{\phi}$ contains potentials in each node and $\vec{B}$ is composed of negative charge densities divided by permittivity in each internal node and boundary conditions in boundary nodes. To draw a picture of the matrix equation composition, let's consider a one-dimensional example containing in its domain only eight nodes ($n_i = 8$). Nodes are indexed by $i = 0,1,\ldots,7$ and the interval *di* between them is constant (Fig. 7).

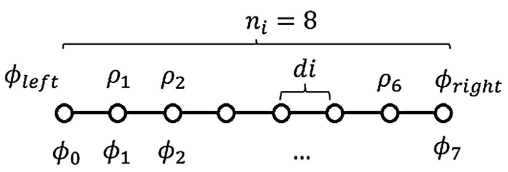

Fig. 7. One-dimensional example of the domain, where: ϕ_i – potential at i-th node, ρ_i – total charge density at i-th node [20].

For the considered domain the following Poisson equations can be formulated [20]:

$$\frac{\phi_{i-1} - 2\phi_i + \phi_{i+1}}{(di)^2} = -\frac{\rho_i}{\varepsilon_0} \xrightarrow{\text{rearranging}} \frac{1}{(di)^2}\phi_{i-1} + \frac{-2}{(di)^2}\phi_i + \frac{1}{(di)^2}\phi_{i+1} = B_i \tag{14}$$

And after transformation, we'll get a system of linear equations in the matrix form $M\vec{\phi} = \vec{B}$:

$$\underbrace{\frac{1}{di^2}\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -2 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}}_{M} \cdot \underbrace{\begin{bmatrix} \phi_0 \\ \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \\ \phi_5 \\ \phi_6 \\ \phi_7 \end{bmatrix}}_{\vec{\phi}} = \underbrace{\begin{bmatrix} \phi_{left}/di^2 \\ -\rho_1/\varepsilon_0 \\ -\rho_2/\varepsilon_0 \\ -\rho_3/\varepsilon_0 \\ -\rho_4/\varepsilon_0 \\ -\rho_5/\varepsilon_0 \\ -\rho_6/\varepsilon_0 \\ \phi_{right}/di^2 \end{bmatrix}}_{\vec{B}} \tag{15}$$

To generalize the above composing method for three-dimensions, let's rearrange (12) to isolate the coefficients beside the potentials:

$$\begin{gathered} \frac{1}{(di)^2}\phi_{i-1,j,k} + \frac{1}{(dj)^2}\phi_{i,j-1,k} + \frac{1}{(dk)^2}\phi_{i,j,k-1} + \left(\frac{-2}{(di)^2} + \frac{-2}{(dj)^2} + \frac{-2}{(dk)^2}\right)\phi_{i,j,k} \\ + \frac{1}{(di)^2}\phi_{i+1,j,k} + \frac{1}{(dj)^2}\phi_{i,j+1,k} + \frac{1}{(dk)^2}\phi_{i,j,k+1} = -\frac{\rho_{i,j,k}}{\varepsilon_0} \end{gathered} \tag{16}$$

To compose matrix M, reindexing must be done to transfer from three indexes to only one, according to the scheme:

$$\phi_{i,j,k} \to \phi_u \tag{17}$$

where: $u = i + n_i j + n_i n_j k$ – new node index ($u = 0,1,\ldots,n_u - 1$), $n_u = n_i n_j n_k$ – total number of nodes.

Operation (17) yields a new form of the Poisson equation, namely:

$$\frac{1}{(di)^2}\phi_{u-1}+\frac{1}{(dj)^2}\phi_{u-ni}+\frac{1}{(dk)^2}\phi_{u-ni*nj}+\left(\frac{-2}{(di)^2}+\frac{-2}{(dj)^2}+\frac{-2}{(dk)^2}\right)\phi_u$$
$$+\frac{1}{(di)^2}\phi_{u+1}+\frac{1}{(dj)^2}\phi_{u+ni}+\frac{1}{(dk)^2}\phi_{u+ni*nj}=-\frac{\rho_u}{\varepsilon_0} \quad (18)$$

To simplify (18), let's set the names of the coefficients next to the potentials, i.e. *a, b, c, d, e, f, g* and right-side *B,* then:

$$a_{u-1}\phi_{u-1}+b_{u-ni}\phi_{u-ni}+c_{u-ni*nj}\phi_{u-ni*nj}+d_u\phi_u+e_{u+1}\phi_{u+1}+f_{u+ni}\phi_{u+ni}+g_{u+ni*nj}\phi_{u+ni*nj}=B_u \quad (19)$$

After rewriting (19) into a matrix form, the following example equation is obtained:

$$\underbrace{\begin{bmatrix} d_0 & e_1 & 0 & f_3 & 0 & 0 & g_6 & 0 & 0 & \cdots & 0 \\ c_0 & d_1 & e_2 & 0 & f_4 & 0 & 0 & g_7 & 0 & \ddots & \vdots \\ 0 & c_1 & d_2 & e_3 & 0 & f_5 & 0 & 0 & g_8 & \ddots & \vdots \\ b_0 & 0 & c_2 & d_3 & e_4 & 0 & f_6 & 0 & 0 & \ddots & \vdots \\ 0 & b_1 & 0 & c_3 & d_4 & e_5 & 0 & f_7 & 0 & \ddots & \vdots \\ 0 & 0 & b_2 & 0 & c_4 & d_5 & e_6 & 0 & f_8 & \ddots & \vdots \\ a_0 & 0 & 0 & b_3 & 0 & c_5 & d_6 & e_7 & 0 & \ddots & \vdots \\ 0 & a_1 & 0 & 0 & b_4 & 0 & c_6 & d_7 & e_8 & \ddots & \vdots \\ 0 & 0 & a_2 & 0 & 0 & b_5 & 0 & c_7 & d_8 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & d_{nu-1} \end{bmatrix}}_{M} \cdot \underbrace{\begin{bmatrix} \phi_0 \\ \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \\ \phi_5 \\ \phi_6 \\ \phi_7 \\ \phi_8 \\ \vdots \\ \phi_{nu-1} \end{bmatrix}}_{\vec{\phi}} = \underbrace{\begin{bmatrix} B_0 \\ B_1 \\ B_2 \\ B_3 \\ B_4 \\ B_5 \\ B_6 \\ B_7 \\ B_8 \\ \vdots \\ B_{nu-1} \end{bmatrix}}_{\vec{B}} \quad (20)$$

In the scope of the current algorithm only cubic mesh is considered, but in the future the intention is to incorporate Finite Element Method (FEM) to support non-cubic elements. Note that some of the right-side coefficients are boundary conditions, not only the first and the last. This depends on the number of divisions in x-, y- and z-directions. A complex numerical code programmatically set these conditions in the proper places of the $\vec{B}$ vector. There are two ways of solving (20), namely the iterative Gauss-Seidel (GS) method and the Newton-Raphson method for nonlinear systems.

2.4. Gauss-Seidel method of solving Poisson equation

The first method to be described is Gauss-Seidel [20]. In it, to find the solution of (12), let's separate potentials $\phi_{i,j,k}$ and rearrange the equation in the following form:

$$\frac{\phi_{i-1,j,k}+\phi_{i+1,j,k}}{(di)^2}+\frac{\phi_{i,j-1,k}+\phi_{i,j+1,k}}{(dj)^2}+\frac{\phi_{i,j,k-1}+\phi_{i,j,k+1}}{(dk)^2}+\frac{\rho_{i,j,k}}{\varepsilon_0}=\left(\frac{2}{(di)^2}+\frac{2}{(dj)^2}+\frac{2}{(dk)^2}\right)\phi_{i,j,k} \tag{21}$$

After dividing (21) by the right-hand term in brackets, we get formula for $\phi_{i,j,k}$:

$$\phi_{i,j,k}=\left(\frac{\phi_{i-1,j,k}+\phi_{i+1,j,k}}{(di)^2}+\frac{\phi_{i,j-1,k}+\phi_{i,j+1,k}}{(dj)^2}+\frac{\phi_{i,j,k-1}+\phi_{i,j,k+1}}{(dk)^2}+\frac{\rho_{i,j,k}}{\varepsilon_0}\right)/\left(\frac{2}{(di)^2}+\frac{2}{(dj)^2}+\frac{2}{(dk)^2}\right) \tag{22}$$

Subsequent values of $\phi_{i,j,k}$ are being computed iteratively until convergence is achieved. This scheme might be generalized to nonlinear problems. Let's assume the right-hand side of (10) is expressed by Boltzmann relation [21], describing electrons as background and ions in terms of charge density, dependent exponentially on potential ϕ.

$$\nabla^2\phi=-\frac{1}{\varepsilon_0}\left(\mathrm{e}Z_i n_i-\mathrm{e}n_{init}\exp\left(\frac{\mathrm{e}}{\mathrm{k_B}T_e}(\phi-\phi_{init})\right)\right) \tag{23}$$

where: e – elementary charge, ε_0 – electrical permittivity, n_i – ion number density, $n_{init}=\frac{\rho_{init}}{e}$ – initial electron number density, ϕ_{init} – initial potential, $\mathrm{k_B}$ – Boltzmann constant, T_e – electron temperature in Kelvins.

To compute the right-hand side (RHS) [21] of (23) the actual charge density $\rho^{ion}_{i,j,k}$ and potential values from previous iteration (old values) $\phi^{old}_{i,j,k}$ must be used.

$$RHS=-\frac{1}{\varepsilon_0}\left(\rho^{ion}_{i,j,k}-\mathrm{e}n_{init}\exp\left(\frac{1}{T_{e(\mathrm{eV})}}\left(\phi^{old}_{i,j,k}-\phi_{init}\right)\right)\right) \tag{24}$$

Now RHS is put to (12):

$$\frac{\phi_{i-1,j,k}-2\phi_{i,j,k}+\phi_{i+1,j,k}}{(di)^2}+\frac{\phi_{i,j-1,k}-2\phi_{i,j,k}+\phi_{i,j+1,k}}{(dj)^2}+\frac{\phi_{i,j,k-1}-2\phi_{i,j,k}+\phi_{i,j,k+1}}{(dk)^2}=RHS \tag{25}$$

And after transformations the iterative formula for potential is:

$$\phi_{i,j,k}=\left(\frac{\phi_{i-1,j,k}+\phi_{i+1,j,k}}{(di)^2}+\frac{\phi_{i,j-1,k}+\phi_{i,j+1,k}}{(dj)^2}+\frac{\phi_{i,j,k-1}+\phi_{i,j,k+1}}{(dk)^2}+RHS\right)/\left(\frac{2}{(di)^2}+\frac{2}{(dj)^2}+\frac{2}{(dk)^2}\right) \tag{26}$$

To calculate the solution in the whole domain, n_u equations have to be solved until required convergence of solution is achieved.

2.5. Newton-Raphson method of solving a nonlinear Poisson equation

Second method of solving a nonlinear set of equations, namely Newton-Raphson [22] is more complex, but yields a significant acceleration of computations. It relies on estimating the solution of function $F(\vec{\phi})=0$, which represents the system of nonlinear equations, and iteratively updating it by the negative product of inversed Jacobian J and function F, according to the relation:

$$\vec{\phi}^{n+1}=\vec{\phi}^{n}-J^{-1}F\left(\vec{\phi}^{n}\right) \tag{27}$$

First, in this iterative method, the algorithm is [21]: (a) computing this part of vector $\vec{B}$, which is dependent on potential ϕ.

$$\vec{B}_{\phi}=\frac{\mathrm{e}n_{init}}{\varepsilon_0}\exp\left(\frac{\vec{\phi}-\vec{\phi}_{init}}{\mathrm{k_B}T_e}\right) \tag{28}$$

The next step (b) is computing function $F\left(\vec{\phi}^{n}\right)$ by subtracting values from vector $\vec{B}=\vec{B}_{init}+\vec{B}_{\phi}$ from product of matrix M (see previous subsection) and vector of potentials $\vec{\phi}$:

$$F\left(\vec{\phi}^{n}\right)=M\cdot\vec{\phi}^{n}-\vec{B}_{init}-\vec{B}_{\phi} \tag{29}$$

Then, (c) Jacobi matrix diagonal elements need to be calculated:

$$P_i=\frac{\mathrm{e}n_{init}}{\varepsilon_0\mathrm{k_B}T_{e,init}}\exp\left(\frac{\phi_i-\phi_{i,init}}{\mathrm{k_B}T_e}\right) \tag{30}$$

Thus, (d) Jacobi matrix is estimated by following formula:

$$J=M-diag(P) \tag{31}$$

The following linear system of equations (e) need to be solved for $\vec{y}$ using a separate method (e.g Preconditioned Conjugate Gradient – PCG. For the algorithm, see next subsection):

$$J\left(\vec{\phi}^{n}\right)\vec{y}=F\left(\vec{\phi}^{n}\right) \tag{32}$$

The last step (f) is to update the potentials:

$$\vec{\phi}^{n+1}=\vec{\phi}^{n}-\vec{y} \tag{33}$$

If $\vec{y}$ is sufficiently small, then computations stop, otherwise next iterations continue until the results converge.

2.6. Preconditioned Conjugate Gradient (PCG) method

PCG [23] is only one of many methods of iterative solution estimation of linear equation in form of: $M\vec{x} = \vec{b}$ for x. The square coefficient matrix M must be symmetric and positive definite, and should also be large and sparse. The column vector $\vec{b}$ must be of length equal to the rank of M. Preconditioner is used to accelerate convergence of the conjugate gradient method. The preconditioner matrix M_P has to be symmetric positive-definite and constant. The method is much faster than Gauss elimination method of computing $\vec{x} = M^{-1}\vec{b}$. See table 1 for PCG algorithm.

Table 1. PCG algorithm [23].

```
Initialization
  ○ $k = 0$
  ○ $\vec{x}_0 = 0$
  ○ $\vec{r}_0 = \vec{b} - M\vec{x}_0$
  ○ $\vec{z}_0 = M_P^{-1}\vec{r}_0$
  ○ $\vec{p}_0 = \vec{z}_0$

  Loop
    ○ $\alpha_k = \frac{\vec{r}_k^T\vec{z}_k}{\vec{p}_k^T M\vec{p}_k}$
    ○ $\vec{x}_{k+1} = \vec{x}_k + \alpha_k\vec{p}_k$
    ○ $\vec{r}_{k+1} = \vec{r}_k - \alpha_k M\vec{p}_k$

    ○ If $||\vec{r}_{k+1}|| < \epsilon_{tol}$, then exit loop.
    ○ $\vec{z}_{k+1} = M_P^{-1}\vec{r}_{k+1}$
    ○ $\beta_k = \frac{\vec{z}_{k+1}^T\vec{r}_{k+1}}{\vec{z}_k^T\vec{r}_k}$
    ○ $\vec{p}_{k+1} = \vec{z}_{k+1} + \beta_k\vec{p}_k$
    ○ $k = k + 1$

  End loop
```

2.7. Electric field

When the potential ϕ is known, the electric field $\boldsymbol{E}$ can be evaluated using the following relation:

$$\boldsymbol{E} = -\frac{\Delta\phi}{\Delta\boldsymbol{r}} \tag{34}$$

Rewriting (34) into a numerical form, expressions for $\boldsymbol{E}$ components internal nodes and boundary nodes can be obtained using central difference scheme (35) and forward/backward difference scheme (36)-(37) respectively [17]:

$$E^x_{i,j,k} = -\frac{\phi_{i+1,j,k} - \phi_{i-1,j,k}}{2di} \quad E^y_{i,j,k} = -\frac{\phi_{i,j+1,k} - \phi_{i,j-1,k}}{2dj} \quad E^z_{i,j,k} = -\frac{\phi_{i,j,k+1} - \phi_{i,j,k-1}}{2dk} \tag{35}$$

$$E^x_{0,j,k} = -\frac{\phi_{1,j,k} - \phi_{0,j,k}}{di} \quad E^y_{i,0,k} = -\frac{\phi_{i,1,k} - \phi_{i,0,k}}{dj} \quad E^z_{i,j,0} = -\frac{\phi_{i,j,1} - \phi_{i,j,0}}{dk} \tag{36}$$

$$E^x_{ni-1,j,k} = -\frac{\phi_{ni-1,j,k} - \phi_{ni-2,j,k}}{di} \quad E^y_{i,nj-1,k} = -\frac{\phi_{i,nj-1,k} - \phi_{i,nj-2,k}}{dj} \quad E^z_{i,j,nk-1} = -\frac{\phi_{i,j,nk-1} - \phi_{i,j,nk-2}}{dk} \tag{37}$$

Knowing ***E*** the electrostatic force ***F*** can be easily obtained by multiplying ***E*** by electric net charge *q* of the particular node. The next subsection describes the operation of updating velocity and moving particles to new positions.

2.8. Pushing particles

Pushing particles consist of velocity updating and moving particles to new locations. This operation requires integration of acceleration vector expressed for the electrostatic case, in n-th time step using the following formula:

$$\boldsymbol{a}^n = \frac{q}{m}\boldsymbol{E}^n \tag{38}$$

where: *q, m* – charge and mass of considered particle, E^n – weighted average of electric field at particle location.

The (38) is suitable only for the case without a magnetic field, while the electromagnetic Lorentz's force requires more sophisticated integration method, e.g. Bunemann-Boris algorithm [17]. But let's consider the electrostatic case and transform (38) to obtain new velocity:

$$\frac{\Delta \boldsymbol{v}^n}{\Delta t} = \boldsymbol{a}^n \quad \rightarrow \quad \frac{\boldsymbol{v}^{n+1} - \boldsymbol{v}^n}{\Delta t} = \boldsymbol{a}^n \quad \rightarrow \quad \boldsymbol{v}^{n+1} = \boldsymbol{v}^n + \boldsymbol{a}^n \Delta t \tag{39}$$

After the velocity update, one can integrate the new position by:

$$\boldsymbol{r}^{n+1} = \boldsymbol{r}^n + 0.5 * \left(\boldsymbol{v}^{n+1} + \boldsymbol{v}^n\right)\Delta t \tag{40}$$

Results from this method are suitable for electrostatic PIC method. At the first stage, in simulations the time steps are assumed to be small enough to ensure proper convergence of the results, but more accurate tests of the stability of applied numerical methods are intended to be performed in the future works.

2.9. Direct Simulation Monte Carlo collisions

In order to simulate fluid flows, even if the considered gases are very rarefied, collisions need to be introduced into the code. As it is well known, interactions between particles inside a fluid govern its behavior. Such properties such as pressure and viscosity are the effects of collisions between particles inside flows. Each change of particle direction leads to energy and momentum exchange, such as flowing around obstacle or collisions with other particles connected with their chaotic thermal motions.

Besides of elastic hard-sphere, another type of collision which is often encountered in heliophysics needs to be considered, namely charge exchange (CEX collision) between ions and neutrals [24]. The main feature of CEX is that the electron jumps from a neutral atom to an ion. In the mechanical sense there is negligible momentum and energy exchange, only electrical charge. This means that in the case of fast SW protons and slow LISM neutrals, due to charge exchange slow ions and fast neutrals could be born, namely energetic neutral atoms (ENA). The following formula shows the process of electron exchange between particles.

$$p^* + H \rightarrow H^* + p \tag{41}$$

where: p^*, p – fast and slow proton, H^*, H – fast and slow neutral hydrogen atom.

CEX collision of fast neutrals and slow ions can also occur, as well as electron impact and photoionization, but in heliospheric processes, the first one has the most profound influence on the heliospheric structure.

To apply collisions into a numerical algorithm, which simulates rarefied gas flows, for which the Knudsen number is less than one ($Kn > 1$), a probabilistic Monte Carlo method can be utilized. This method, first proposed by Prof. Graeme Bird is called Direct Simulation Monte Carlo (DSMC) and uses macromolecules impingements to solve Boltzmann equation. In the DSMC method fluid dynamics is simulated by large number of real molecules or macromolecules, which move through computational domain, coupled by particle-particle interactions performed in the probabilistic way. The interactions in simulations usually are collisions but in ES-PIC algorithm we adapt this method to simulate also CEX collision. The DSMC algorithm is shown in Table 2.

Table 2. DSMC algorithm implemented in the electrostatic PIC program (for more see [25]).

```
Initialization
    o  Get time interval $dt$ and maximum volume velocity $Q_{max}\left[\frac{m^3}{s}\right]$ from the main program as well as list of particles
    o  Sort particles to the cells (create particle containers corresponding to each cell)
    o  Set maximum temporary volume velocity to 0: $Q_{max,tmp} = 0$
    o  Compute cell volume: $dV = di \cdot dj \cdot dk$
    o  Set number of particles in macroparticle: $N_m$
    o  Set collision counter to 0

    o  Loop over all cells
       1.  Count number of particles in cell: $N_p$
       2.  If number of particles $N_p$ in cell in less than 2, than skip collision in this cell
       3.  Compute number of groups: $N_g = 0.5 \cdot \frac{N_p^2 \cdot N_m \cdot Q_{max} \cdot dt}{dV}$
       4.  Convert float number of groups to integer number
       5.  Iterate over all numbers of groups
       6.  Randomly select two different particles in group and take its positions ($\vec{p}_1, \vec{p}_2$) and velocities ($\vec{v}_1, \vec{v}_2$)
       7.  Compute relative velocity between selected particles: $\vec{c}_r = \vec{v}_1 - \vec{v}_2$
       8.  Compute relative velocity magnitude: $c_r = |\vec{c}_r|$
       9.  Evaluate collision cross section: $\sigma_{colls}\ [m^2]$ (at this stage of works it is assumed to be constant)
       10. Compute volume velocity $Q = \sigma_{colls} \cdot c_r$
       11. If $Q > Q_{max,tmp}$ update maximum temporary volume velocity: $Q_{max,tmp} = Q_{c_r}$
       12. Evaluate probability of collision: $P = \frac{Q}{Q_{max}}$
       13. Compare evaluated value with a randomly selected probability $P_{rnd}$
       14. Perform collision between particles if $P > P_{rnd}$ (see table 3 and table 4 below)
    o  End loop

    o  Return maximum temporary volume velocity $Q_{max,tmp}$ to the main program
```

Table 3. Elastic collision algorithm [25].

```
o Take velocities of two colliding particles $\vec{v}_1,\ \vec{v}_2$
o Take mass of two colliding particles $m_1, m_2$
o Compute center of mass velocity of two particles: $\vec{v}_{cm} = \frac{m_1\vec{v}_1+m_2\vec{v}_2}{m_1+m_2}$
o Compute relative velocity between colliding particles: $\vec{c}_r = \vec{v}_2 - \vec{v}_1$
o Compute relative velocity magnitude: $c_r = |\vec{c}_r|$
o Pick two random angles $\chi, \varepsilon$ according to Bird's method:
      $\cos(\chi) = 2 \cdot n_{random} - 1$
      $\sin(\chi) = \sqrt{1-\cos^2(\chi)}$
      $\varepsilon = 2\pi \cdot n_{random}$        where random number is defined as: $n_{random} \in [0 \div 1]$
o Perform rotation of relative velocity $\vec{c}_r$ about angles $\chi, \varepsilon$:
      $\vec{c}_r = c_r \cdot \begin{pmatrix} \cos(\chi) \\ \sin(\chi)\cos(\varepsilon) \\ \sin(\chi)\sin(\varepsilon) \end{pmatrix}$
o Compute velocities of particles $\vec{v}_1, \vec{v}_2$ after collision:
      $\vec{v}_1 = \vec{v}_{cm} + \frac{m_2}{m_1+m_2}\vec{c}_r$
      $\vec{v}_2 = \vec{v}_{cm} - \frac{m_1}{m_1+m_2}\vec{c}_r$
o Assign new velocities to colliding particles
```

Table 4. Charge exchange collision algorithm.

```
o Get ion velocity magnitude: $v_{ion} = |\vec{v}_{ion}|$
o Pick two random angles $\chi, \varepsilon$ according to Bird's method:
      $\cos(\chi) = 2 \cdot n_{random} - 1$
      $\sin(\chi) = \sqrt{1-\cos^2(\chi)}$
      $\varepsilon = 2\pi \cdot n_{random}$        where random number is defined as: $n_{random} \in [0 \div 1]$
o Perform rotation of ion velocity $\vec{v}_{ion}$ about angles $\chi, \varepsilon$:
      $\vec{v}_{ion} = v_{ion} \cdot \begin{pmatrix} \cos(\chi) \\ \sin(\chi)\cos(\varepsilon) \\ \sin(\chi)\sin(\varepsilon) \end{pmatrix}$
o Create new ion particle with position $\vec{p}_{ion}$ and velocity $\vec{v}_{ion}$ of the old neutral particle ($\vec{p}^{old}_{neutral}, \vec{v}^{old}_{neutral}$):
      $\vec{p}_{ion} = \vec{p}^{old}_{neutral}$
      $\vec{v}_{ion} = \vec{v}^{old}_{neutral}$
o Create new neutral particle with position $\vec{p}_{neutral}$ and velocity $\vec{v}_{neutral}$ of the old ion particle ($\vec{p}^{old}_{ion}, \vec{v}^{old}_{ion}$):
      $\vec{p}_{neutral} = \vec{p}^{old}_{ion}$
      $\vec{v}_{neutral} = \vec{v}^{old}_{ion}$
o Delete old particles
```

The 14th step in DSMC algorithm is responsible for performing collision. It depends on the assumed model, if the collision is simple hard sphere elastic, which exchanges momentum and energy (see Table 3) or charge exchange (see Table 4). The results of different kinds of collision models are exchanged energy and momentum between particular part of the fluid, in the case of mechanical collision, or between ionized and neutral species in the case of CEX. In electrostatic PIC model, these two kinds of collision models are used. To simulate the interaction between two different counter-flowing magnetized plasmas, first with one component simulating only ions and the second one as two components simulating mixture of ions and neutrals, mechanical hard sphere collision is applied between ionized components, while both interact with neutrals by charge exchange collisions. This simplified approach, can give us some characteristic features of two supersonically counter-flowing plasmas interaction such as supersonic two shock shape of the flow with separating layer between these two mediums. This method is some kind of bypassing the problem of lack of the magnetic field. Although we want to consider this interaction to be collisionless, we use the collisions to separate these two streams, which are separated in real world in magnetic manner. Collisions only pertain to particles belonging to different species. In the same species particles don't collide with each other, so the flow is still in some approximation collisionless. The tables 3. and 4. describe elastic hard sphere and charge exchange collision algorithms.

2.10. Example simulation

In this subsection two simulations are presented to check the Poisson solver and DSMC method implementations. Simulated models aren't strictly the models of SW-LISM interaction. The only intention of these examples is the confirmation of correctness of the methods implementation: collisions occurrence and electrostatic drift of charged particles. Parameters of the simulations are selected in such way to make computations as inexpensive as possible. More, not only qualitative but also quantitative tests, which apply more to the heliospheric processes in physical manner, are expected to be performed in the future, but they aren't in the scope of this work.

In order to check the qualitative correctness of Poisson solver a simulation of ion flow around a sphere was performed. The domain (Fig. 8) was modelled as a 20x20x40 cm cube. At its center a source sphere with the radius of 4 cm was set. The whole domain was discretized by 20x20x40 cells. One sources was set: a uniform flow of particles from xy-plane into the domain with z-velocity 7000 m/s. It was assumed that particles are single ionized oxygen with number density of 10^{11} m^{-3}. Deby'e length is 0.0105132 m, the simulation lasted 0.00004 s (200 time steps each of $2*10^{-7}$ s) and the results are shown in Fig. 9 to Fig. 13. The influence of electric field is evident. Ionized particles accelerate in the direction of decreasing potential, what was expected, resulting in attraction and absorption by the sphere.

In order to check the qualitative correctness of DSMC method implementation a simulation of counter-flowing neutral fluid was performed. The domain (Fig. 14) was modelled as a 20x20x20 cm cube. At its center a source sphere with the radius of 4 cm was set. The whole domain was discretized by 40x40x40 cells. Two sources were set: 1) uniform flow of particles from xy-plane into the domain with z-velocity 1000 m/s, 2) spherically uniform flow of particles from the central sphere with a velocity normal to its surface of 1000 m/s. It was assumed that particles of both sources are helium with number density of 10^{17} m^{-3}. DSMC collisions were enabled with interaction cross section of 10^{-14} m^{2} of macroparticles consisting of $2*10^{10}$ particles. Deby'e length is 0.000033 m, the simulation lasted 0.05 s (1000 time steps each of $5*10^{-5}$ s) and the results are shown in Fig. 15 and Fig. 16. It is evident that two counter-flowing streams collide with each other creating a structure of increased density, which confirms the DSMC method works enabling compressibility of the flow.

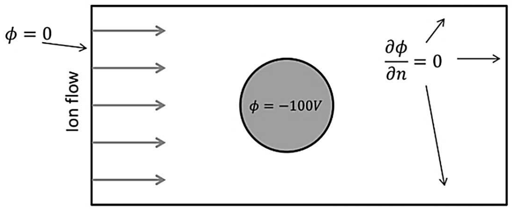

Fig. 8. Geometry of simulated domain.

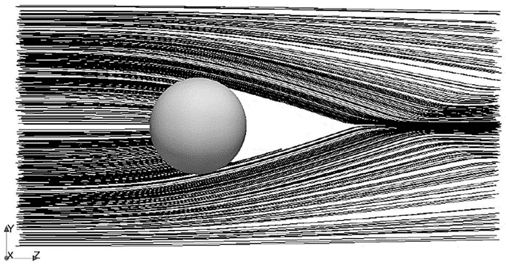

Fig. 9. Streamlines of velocity magnitude.

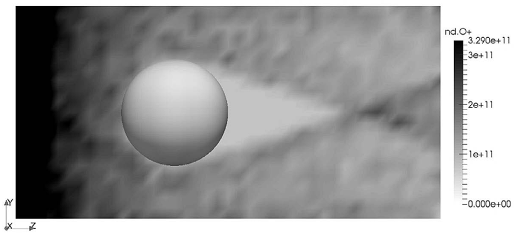

Fig. 10. Number density in [m^{-3}] after 0.00004 s simulation time.

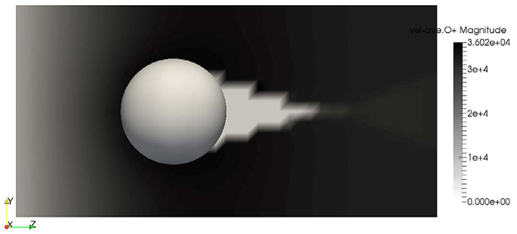

Fig. 11. Velocity magnitude in [m/s] after 0.00004 s simulation time.

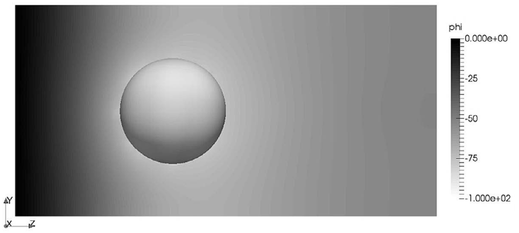

Fig. 12. Electric potential in [V] after 0.00004 s simulation time.

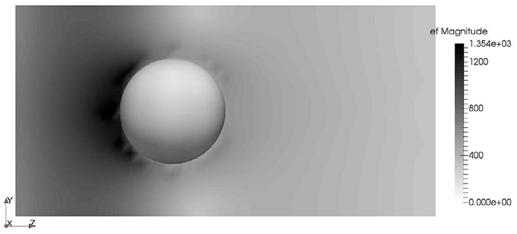

Fig. 13. Electric field magnitude in [V/m] after 0.00004 s simulation time.

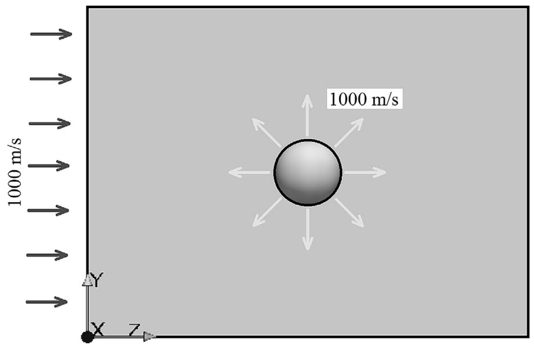

Fig. 14. Geometry of simulated domain.

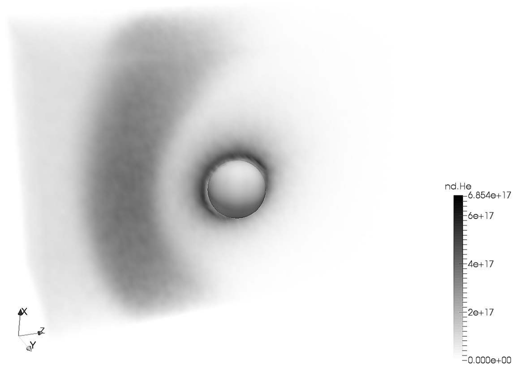

Fig. 15. Number density in [m^{-3}] after 0.05 s simulation time.

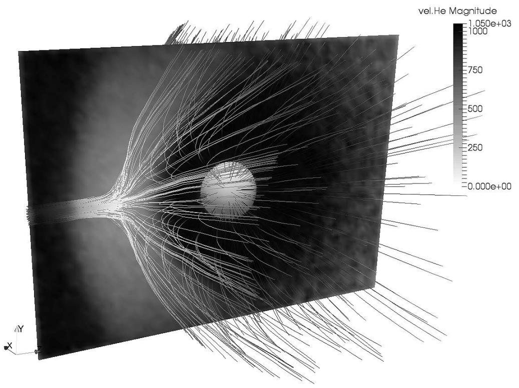

Fig. 16. Velocity magnitude in [m/s] after 0.05 s simulation time with visualization of streamlines from source at the center and little source region of the streamlines at xy-plane.

3. CONCLUSIONS

The first step to develop a fully kinetic Particle-in-Cell (PIC) model to describe the interaction between SW and the LISM was made. A simple electrostatic approach was applied, which reduced Maxwell's set of equations for the problem to only one equation of Gauss law for electricity. From the numerical point of view, to find a solution of the electrostatic problem, the Poisson equation needs to be solved for electric potential. Both a simple Gauss-Seidel and a much more sophisticated Newton-Raphson with Preconditioned Conjugate Gradient schemes were used to find the electric field values on the computational grid, which served to compute the forces acting on particles moving through the domain. A simple algorithm was used to integrate equations of motion in order to determine particle velocities and positions. The Direct Simulation Monte Carlo method was used to simulate collisions between particles of two counter-flowing streams of ions, as well as charge exchange collisions between ionized fluids and neutral component of one fluid. Two qualitative tests were performed to check correctness of implementation for Poisson solver and for Direct Simulation Monte Carlo method. More tests to validate not only applied methods of computation but also the physical models, are foreseen in the further works. Results obtained in the scope of this work are a promising step towards a fully electromagnetic PIC model of SW-LISM interaction. The feasibility of the proposed approach as a tool for the investigation of the heliosphere will be discussed in the future paper.

ACKNOWLEDGMENTS

The work was carried out as part of a research project funded by the Institute of Aviation. WK also thanks prof. R. Ratkiewicz for all the support and reviewers for their remarks.

BIBLIOGRAPHY

[1] Davis, L., 1955, “Interplanetary Magnetic Fields and Cosmic Rays”, Physical Review, Volume 100, Issue 5, pp. 1440-1444.

[2] Parker, E., 1963, “Interplanetary dynamical processes”, Interscience Publishers, New York, NY.

[3] Axford, W. I., 1972, “The Interaction of the Solar Wind With the Interstellar Medium”, Solar Wind. Edited by Charles P. Sonett, Paul J. Coleman, and John M. Wilcox, Washington, Scientific and Technical Information Office, NASA, p.609.

[4] Baranov, V. B., Malama, Yu. G., 1993, “Model of the solar wind interaction with the local interstellar medium – Numerical solution of self-consistent problem”, Journal of Geophysical Research, Volume 98, Issue A9, pp. 15157-15163.

[5] Pauls, H. L., Zank, G. P., Williams, L. L., 1995, “Interaction of the solar wind with the local interstellar medium”, Journal of Geophysical Research, Volume 100, Issue A11, pp. 21595-21604.

[6] Zank, G. P., Pauls, H. L., Williams, L. L., Hall, D. T., 1996, “Interaction of the solar wind with the local interstellar medium: A multifluid approach”, Journal of Geophysical Research, Volume 101, Issue A10, pp. 21639-21656.

[7] Washimi, H., Tanaka, 1996, “3-D Magnetic Field and Current System in the Heliosphere”, Space Science Reviews, Volume 78, Issue 1-2, pp. 85-94.

[8] Ratkiewicz, R., Barnes, A., Molvik, G. A., Spreiter, J. R., Stahara, S. S., Vinokur, M., Venkateswaran, S., 1998, “Effect of varying strength and orientation of local interstellar magnetic field on configuration of exterior heliosphere: 3D MHD simulations”, Astronomy and Astrophysics, Volume 335, pp. 363-369.

[9] Pogorelov, N.V., Matsuda, T., 1998, “Influence of the interstellar magnetic field direction on the shape of the global heliopause”, Journal of Geophysical Research, Volume 10, pp. 237.

[10] Opher, M., Stone, E. C., Liewer, P. C., 2006, “The Effects of a Local Interstellar Magnetic Field on Voyager 1 and 2 Observations”, Astrophysical Journal, Volume 640, Issue 1, pp. L71-L74..

[11] Heerikhuisen, J., Pogorelov, N.V., 2010 “Kinetic Modeling of Interstellar Hydrogen in the Heliosphere”, Numerical Modeling of Space Plasma Flows, Astronum-2009, Astronomical Society of the Pacific, San Francisco, pp. 227.

[12] M. Strumik, A. Czechowski, S. Grzedzielski, W. M. Macek, and R. Ratkiewicz, 2013, “Small-Scale Local Phenomena Related to the Magnetic Reconnection and Turbulence in the Proximity of the Heliopause”, Astrophysical Journal Letters, Volume 773, L23 pp. 1-5.

[13] Kunz, M. W., Schekochihihin, A. A., Chen, C. H. K., Abel, I. , Cowley, S. C., 2015, “Inertial-range kinetic turbulence in pressure-anisotropic astrophysical plasmas”, J. Plasma Phys., Volume 81, Issue 5, pp. 1-61.

[14] Howes, G. G., Cowley, S. C., Dorland, W., Hammett, G. W., Quataert, E., Schekochihin, A. A., 2006, “Astrophysical Gyrokinetics: Basic Equations and Linear Theory”, The Astrophysical Journal, Volume 651, Issue 1, pp. 590-614.

[15] Valentini, F., Trávníček, P., Califano, F., Hellinger, P., Mangeney, A., 2007, “A hybrid-Vlasov model based on the current advance method for the simulation of collisionless magnetized plasma”, Journal of Computational Physics, Vol. 225, Issue 1, pp. 753.

[16] Ren, C., 2011, "Introduction to Particle-in-cell Methods in Plasma Simulations", The 2011 HEDP Summer School, University of Rochester, from http://hedpschool.lle.rochester.edu/2011SummerSchool/lectures/Ren.pdf.

[17] Birdsall, C. K. and Langdon A. B., 2005, "Plasma Physics via Computer Simulation", Taylor & Francis Group, New York, NY.

[18] Brieda, L., 2010, "The Electrostatic Particle In Cell (ES-PIC) Method", from https://www.particleincell.com/2010/es-pic-method/.

[19] Griffiths, D. J., 1999, "Introduction to Electrodynamics", 3th Edition, Prentice Hall, Upper Saddle River, New Jersey, NJ.

[20] Kahan, W., 1958, "Gauss-Seidel Methods of Solving Large Systems of Linear Equations", Ph.D. thesis, University of Toronto.

[21] Brieda, L., 2016, "Fundamentals of the Particle In Cell Method 2016", lecture materials, Particle In Cell Consulting LLC, California, CA.

[22] Deuflhard, P., 2004, "Newton Methods for Nonlinear Problems. Affine Invariance and Adaptive Algorithms", Volume 35, Springer, Berlin.

[23] Barrett, R., Berry, M., Chan, T. F., Demmel, J., Donato, J., Dongarra, J., Eijkhout, V., Pozo, R., Romine. C., van der Vorst, H., 1994, "Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods", SIAM, Philadelphia.

[24] Blum, P.W., Fahr, H.J., 1970, "Interaction between Interstellar Hydrogen and the Solar Wind", Astronomy & Astrophysics Volume 4, pp. 280-290.

[25] Bird, G. A., 1994, "Molecular Gas Dynamics and the Direct Simulation of Gas Flows", Clarendon Press, Oxford.

ALGORYTM NUMERYCZNY PARTICLE-IN-CELL DLA PROBLEMU ELEKTROSTATYCZNEGO

Streszczenie

Istniejące globalne modele oddziaływania wiatru słonecznego (SW) z lokalną materią międzygwiazdową (LISM) opisują heliosferę, która powstaje w wyniku interakcji tych dwóch ośrodków. Istnieje silna motywacja do opracowania modelu kinetycznego wykorzystującego metodę Particle-in-Cell (PIC) w celu opisu zjawisk, które zachodzą w heliosferze. Jednakże jest to długoterminowy cel naukowy. W artykule przedstawiono elektrostatyczny model numeryczny PIC, opracowany w Instytucie Lotnictwa w Warszawie, który obejmuje kolizje mechaniczne i rezonansową wymianę ładunków pomiędzy cząstkami w sposób probabilistyczny metodą Direct Simulation Monte Carlo. Jest to pierwszy krok w opracowywaniu symulacji heliosfery zawierającej efekty kinetyczne w plazmach bezzderzeniowych. W tym artykule koncentrujemy się tylko na prezentowaniu prac, które zostały wykonane z wykorzystaniem algorytmu numerycznego PIC.

Słowa kluczowe: plazma, heliosfera, Particle-in-Cell, Monte-Carlo.