# Numerical simulation of motion of dust particles in an accelerator path

A.V. Piyakov[1], D.V. Rodin[1], M.A. Rodina[1], A.M. Telegin[1]

[1]*Samara National Research University, 34 Moskovskoe Shosse, 443086, Samara, Russia*

**Abstract**

The model of micron charged particles motion in an electrostatic accelerator path is presented. This article describes software that provides formation of a particle packet with given statistical characteristics and the results of modeling, obtained by the supercomputer Sergey Korolev. Performance of software implementation for the supercomputer, parallel implementation for PC and implementation for GPU is being considered. Comparison with experimental data was carried out; convergence of full-scale and numerical experiments was shown.

*Keywords:* trajectory calculation; electrostatic accelerator; supercomputer; GPU; CUDA

## 1. Introduction

Space debris remains one of the main causes of degradation of spacecraft structural elements. At the same time tendency to increase concentration of technogenic dust particles in near-earth orbit remains [1]. Considering the current trend for increasing the duration of spacecraft operation, as well as to reduce their weight and dimensions, it is necessary to have ground-based experimental facilities that allow research of new materials in conditions of interaction with high-speed dust particles.

For such research accelerators of various types have been developed, in particular, the Van de Graaf accelerator, electrostatic and electromagnetic accelerators. Electromagnetic accelerators working on the principle of a railgun are most widely used to accelerate large particles (from 1 mm and above) [2]. Electrostatic accelerators, containing drift tubes to which an accelerating voltage is applied, are used to research the degradation of materials in the flow of high-speed micron particles. The in-phase switching of the accelerating voltages with increasing particle velocity can be provided in two ways: 1) increasing the length of the drift tubes; 2) increasing the frequency of the accelerating voltage. The second method is more preferable, because it provides greater flexibility in tuning such an accelerator [3].

The main characteristics of such an experimental facility are the range of output velocities and the particle flux density at the exit from the accelerator path. The main task in developing such accelerators is a maximization of these parameters. The solution of this problem is complicated by the fact that the full-scale modeling of constructions is largely hampered by their large geometric dimensions, use of high voltages and the need to create a vacuum system for each design. The only way to design such units is through mathematical modeling. To evaluate the characteristics of particles at the exit of the accelerator path with given parameters (such as the number of drift tubes, accelerating voltage and geometric parameters of tubes.) the authors developed software that solves the assigned task.

## 2. Preparation of initial data

In the laboratory facility, the injector generates a stream of charged particles with a probabilistic distribution of velocities, as well as probabilistic radial and angular distribution of particles in a packet. Then particles come on the input of electrostatic accelerator.

For the purpose of mathematical simulation of particle trajectories in the accelerator path, it is necessary to form model packets of the particles with probabilistic characteristics matching the characteristics of real packets. In order to do that, the software module that generates random variables with required distribution law was written.

To generate a model packet of particles with a distribution similar to the real one, the Box-Muller transformation with subsequent summation of the velocity vector components and normalization for the most probable energy corresponding to 450 m/s was used.

The algorithm of the generator of a model packet of particles can be represented in the following form:

1) formation of two random values $U_1$ and $U_2$ on the interval (0; 1] with the help of a generator of uniformly distributed random variables;

2) obtaining two random variables distributed normally, in accordance with expression:

$$x = \sqrt{-2\ln U_2}\cos(2\pi U_1);$$

$$y = \sqrt{-2\ln U_2}\sin(2\pi U_1);$$

1) the repetition of points 1 and 2 makes it possible to obtain three normally distributed numbers whose geometric sum is a velocity vector in three-dimensional space:

$$V = V_x^2 + V_y^2 + V_z^2;$$

2) normalization of the total velocity vector with the help of the scale factor obtained from the most probable speed.

The distribution of velocities of charged particles at the output of the injector is shown in the Fig. 1. The distribution of the model packet of particles by velocities is shown in the same figure.

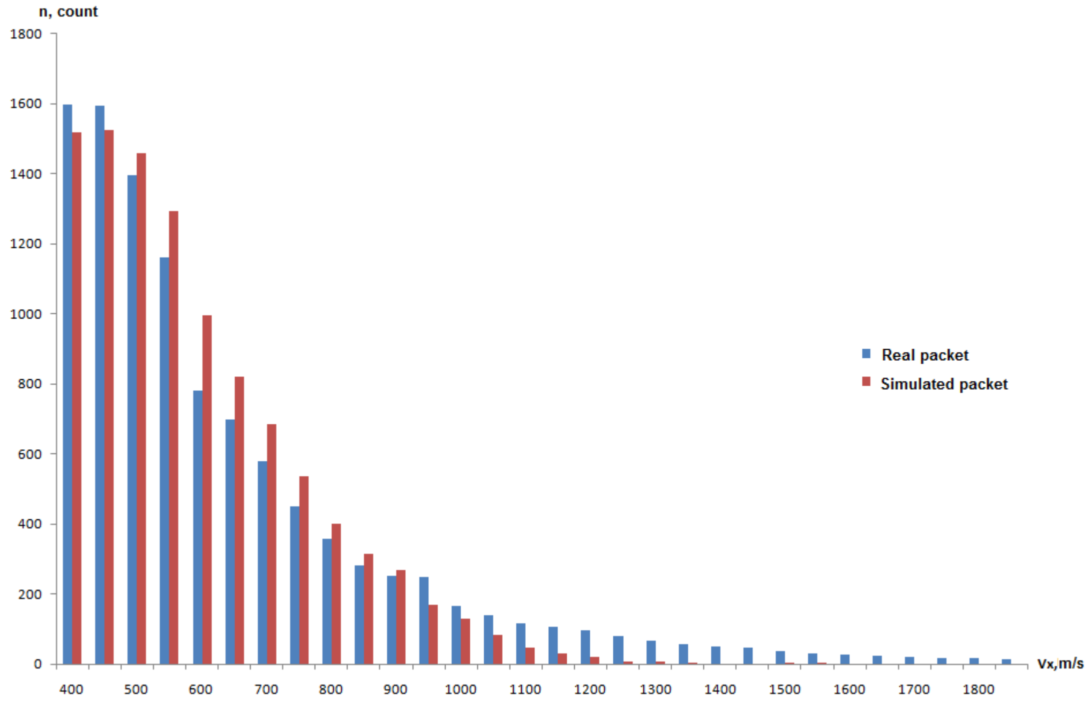

Fig. 1. Distribution of real and model particle packets on axial speeds.

The generation of the radial coordinates of the model packet of particles is also based on the Box-Muller transformation. The distribution of a model packet at radial coordinate is shown in Fig. 2.

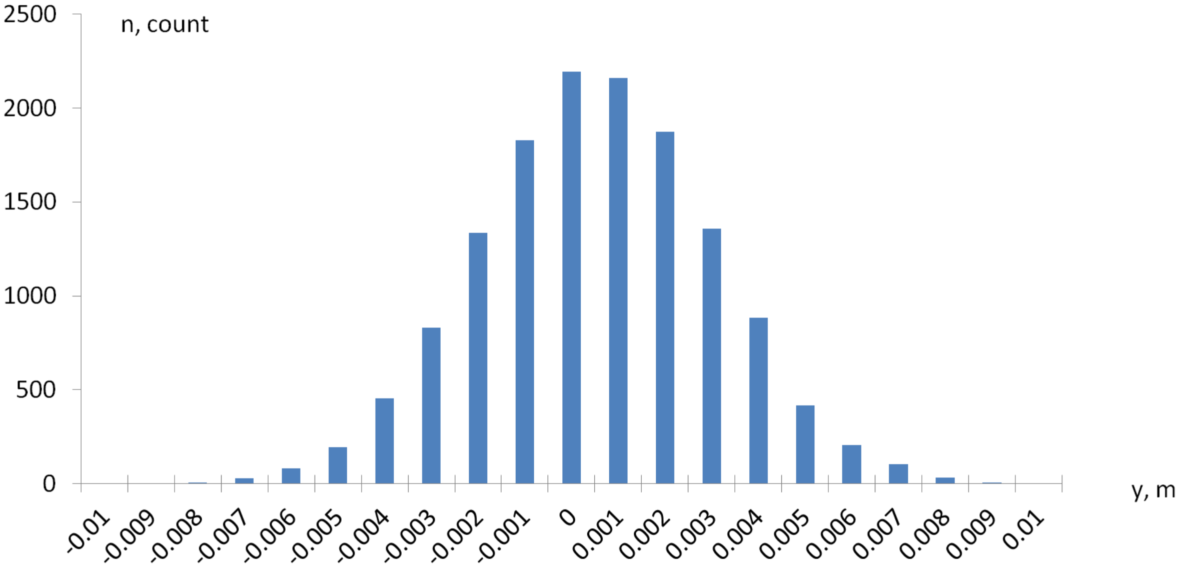

Fig. 2. The distribution of a model packet of particles at radial coordinates.

Due to the ratio of the size of the hole at the output of the injector and the distance to the charging needle, charged particles enter the input of the electrostatic accelerator at a solid angle equal to 2°. When forming the distribution of a model packet of particles at radial velocities (Fig. 3), particles with an unsuitable ratio of the radial and axial velocity components are excluded from consideration.

The acceleration received by the particle inside the path is largely determined by the ratio of mass to a charge. Therefore, the generator of the model packets also formed the probability distribution of the mass-to-charge ratio. The initial form of a distribution for a full-scale experiment was obtained indirectly, starting from a priori knowledge of the accelerating potential, as well as the velocities of the particles at the input and the output of the path. The real and model distributions are shown in Fig. 4.

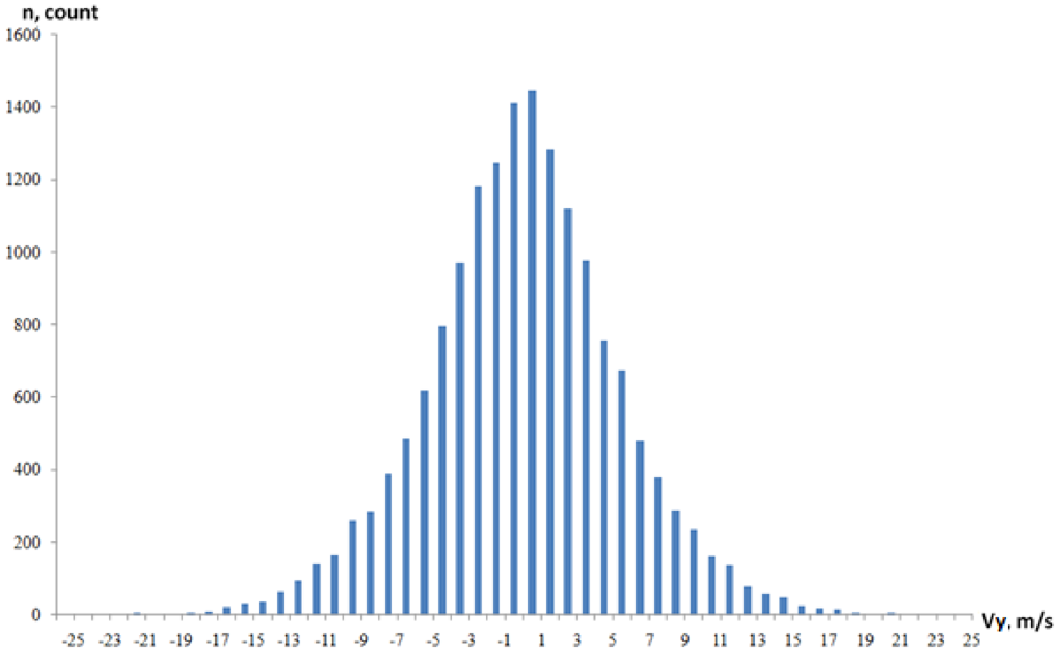

Fig. 3. The distribution of a model packet of particles on radial velocities.

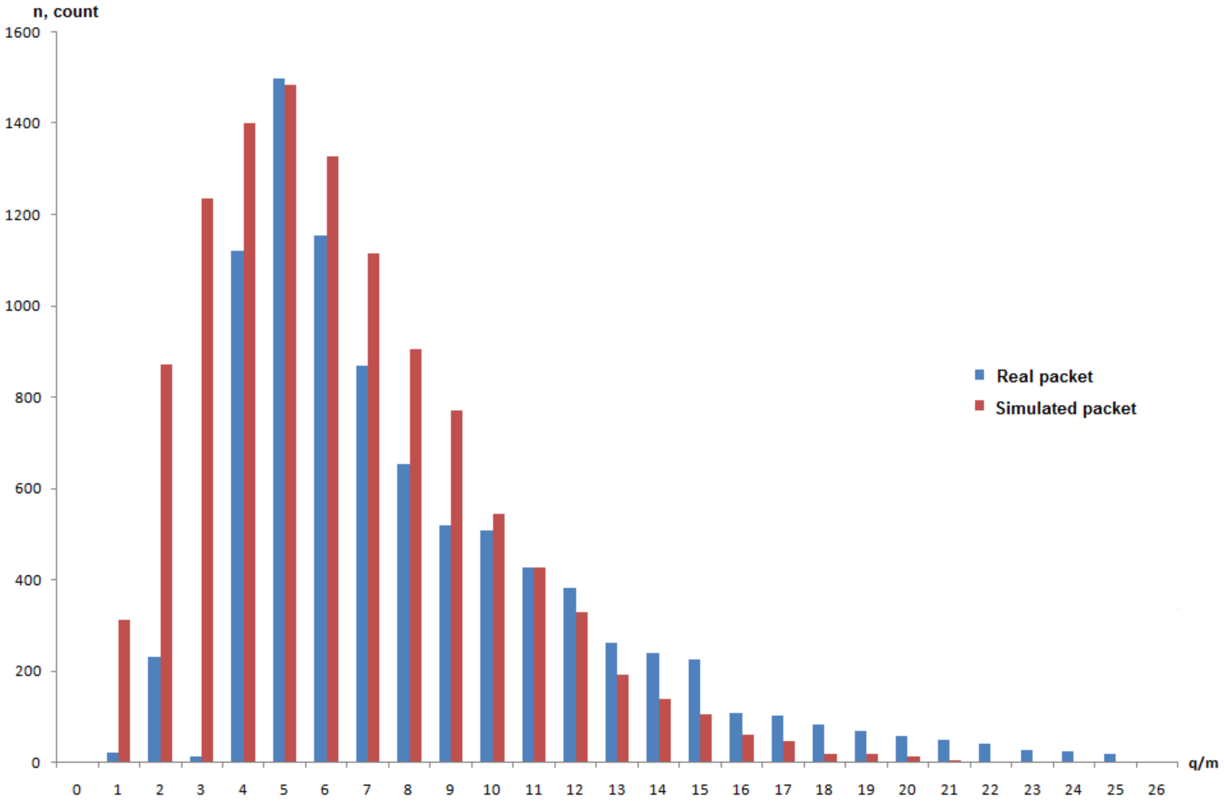

Fig. 4. Distribution of real and model particle packets on mass to particle charge in the package.

The motion of charged particles was simulated on a two-dimensional rectangular grid because of the axial-symmetry of the task. The grid nodes contain the values of electrostatic potential. Using the numerical calculation of the potential distribution for five accelerating tubes with a 10 kV potential difference between adjacent tubes, which corresponds to the real parameters of the accelerator, the field values at the nodes were obtained. The grid of the cell has a step of $9{,}775 \cdot 10^{-5}$ m which corresponds to splitting the 10 cm section into 1023 intervals or 1024 nodes.

## 3. Calculation of the trajectory of particles in a linear accelerator path using a personal computer

The program for calculating the trajectories of charged particles in the path of a linear electrostatic accelerator using a personal computer was written in the C # programming language. The package of model particles contains 16384 pieces and was formed according to the algorithms described in paragraph 2. All particles start from the coordinate $x$ = 0 m, the radial coordinate is subject to the normal distribution law and lies in the range from -0.01 to 0.01 m. The components of the particle velocity vector correspond to the parameters of the real distribution of charged particles after the injector. To preserve the state of particles, the Particle class was used, which has the necessary fields for storing two coordinates, two components of the velocity vector, a time quantum and the total time of flight.

Interpolation of the field was carried out on the assumption that the particles have only a positive coordinate at the radial axis. For this the operation of taking the module from the radial coordinate of the particle is added to the interpolator function. Interpolation occurred for a field section of 1 cm x 10 cm, respectively, the $x$ coordinate must always lie in the range 0 ÷ 0.1 m.

After a interpolation operation for particles having a negative radial coordinate, the radial field component must also be inverted, using the negative radial coordinate flag.

Particles were traced by the fourth-order Runge-Kutta method. The calculation for each of the particles was completed after the onset of one of two events: the particle exceeded the limit -0.01 m or 0.01 m along the radial coordinate, which corresponds to the precipitation of the particle on the accelerating tube, or the particle exceeded the limit of 1 m along the longitudinal coordinate, which corresponds to the passing of a particle of the entire accelerator section.

Every 0.025 m the slice of all the characteristics of particles was built. After each iteration, the value of the time quantum for each of the particles was adjusted from the condition of the minimum number of steps per one grid cell.

All the initial parameters, such as a set of particles and field values, were saved for later use on the supercomputer Sergey Korolev (SK) and for calculations using the GPU accelerator.

The results of modeling the flight of a particle packet through the accelerator path are shown in Fig. 5 – 7.

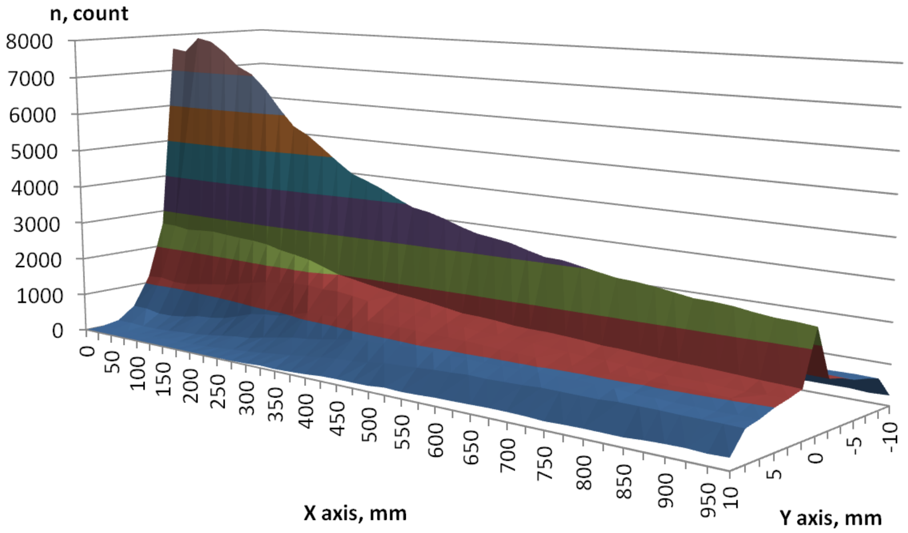

Fig. 5. The distribution of particles in the accelerator path in sections at the radial and longitudinal axes.

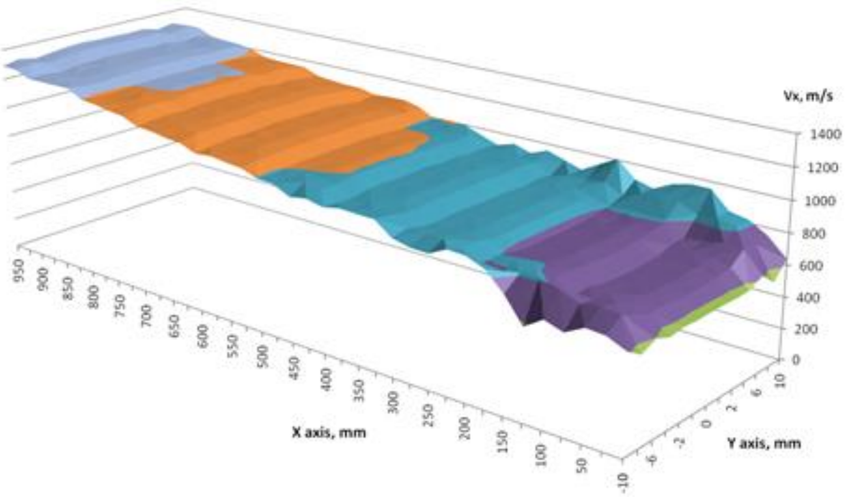

Fig. 6. Distribution of the average longitudinal velocity in the accelerator path in sections at the radial and longitudinal axes.

It can be seen that at the output from the accelerator path the average particle velocity increases more than twice, and the number of particles on the axis of the path decreases by approximately 2.5 to 3 times, however, the number of particles at the periphery increases.

The graph of distribution of mean radial velocity allows for the conclusion about the effect of focusing in the radial plane, because most of the time of flight of the accelerator path, the particle velocity along the radial coordinate has a sign opposite to the sign of their radial coordinate, i.e. particles tend to the axis of the device.

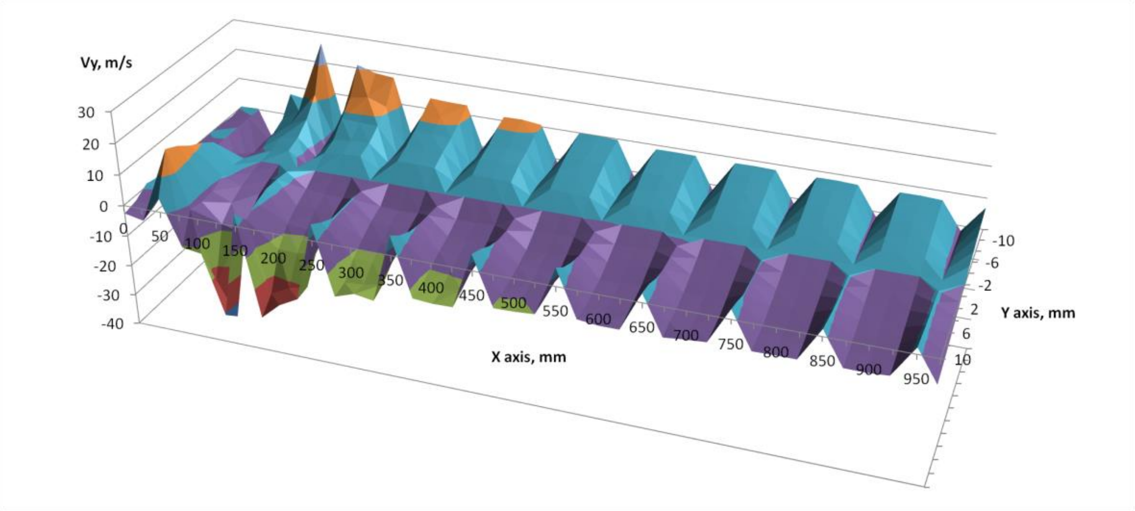

Fig. 7. Distribution of the mean radial velocity in the accelerator path in sections at the radial and longitudinal axes.

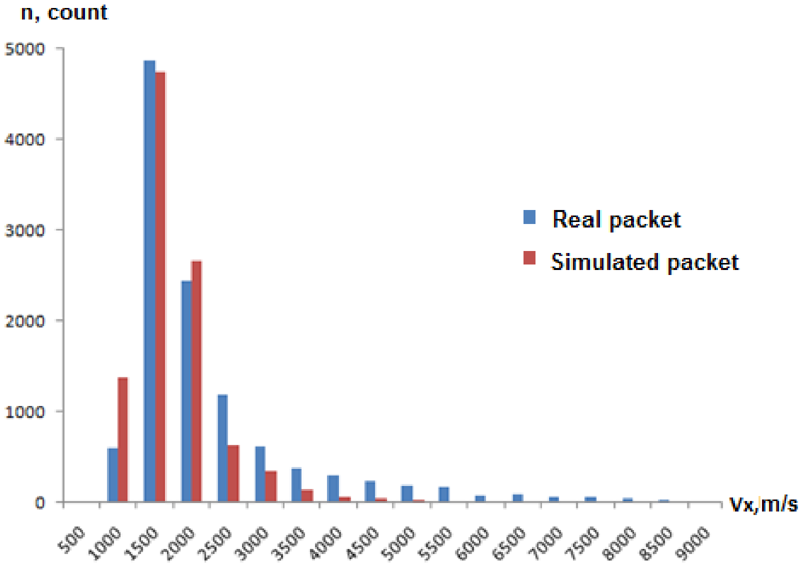

Fig. 8. Distribution of real and model particle packets on longitudinal velocities.

The main characteristic of an electrostatic accelerator is the range of speeds obtained at the output of its path. Data obtained by numerical simulation and by a real accelerator are shown in Fig. 8.

The above graphs show a good convergence of the results of numerical and full-scale experiments. In particular, the most probable particle velocity at the output from the path is approximately 1.5 ÷ 2 km/s and the number of particles reaching the output of the path for a real accelerator is 69.9 %, and for the model – 71.3 %.

## 4. Calculation of the trajectory of particles in the linear accelerator path using the supercomputer SK

To calculate the trajectories using the supercomputer SK, an implementation of the program was developed to run on 16 processors. The implementation of the program is written in the programming language C, for storing particle parameters, a structure similar to the Particle class from paragraph 3 was used.

The matrix of the electrostatic field, as well as an array with particle parameters were connected as external header files with two-dimensional arrays declared and assigned inside. The implementation of multithreading was provided by connecting the MPI library. Another difference in the implementation of the program for the supercomputer SK was that two separate methods were used to interpolate the field, since the field arrays were connected as two separate header files.

Similarly to the program implementation for a personal computer, the calculation for each of the particles was carried out by fourth-order Runge-Kutta method before the particle leaves the path. Slices of the state of the particle packet were also constructed every 0.025 m by outputting data to the output stream.

After the end of the program, the supercomputer SK software automatically collected data from each of 16 nodes into one file on the head node. The result of execution is a set of states of particles on slices that needed to be sorted by the particle index.

The results of modeling the flight of a particle packet through the accelerator path using the supercomputer SK almost completely coincide with the graphs in Fig. 5 – 7, built on the data obtained on a personal computer.

## 5. Calculation of the trajectory of particles in the linear accelerator path using the GPU accelerator

This implementation of the program for calculating the trajectories of charged particles in the path of a linear electrostatic accelerator was written in C # using CUDA libraries. The difference from the other two implementations is the calculation of particle trajectories on the GPU accelerator, which requires special data preparation.

The initial data for the calculation was loaded from the files saved by the program from paragraph 3. Further, all the variables of the double data type were stored in arrays of the same length as the number of particles. In the host memory, data arrays of vector type int2 were created for storing 64-bit variables, then the data was converted and data of the vector type was copied to the GPU memory where the calculation was performed. The values of the field at the grid nodes were stored in the texture memory, the values of the particle characteristics – in the surface memory.

Calculation of the trajectories was carried out by fourth-order Runge-Kutta method, data upload from the GPU was performed during the flight of the next 0.025 m, or when particle leaves the path. Similar to the other two implementations, distributions of the particle characteristics at the path sections were constructed.

The results of modeling the flight of a particle packet through an accelerator path using a GPU accelerator coincide with the results obtained on a personal computer and a supercomputer (Fig. 5 – 7) with small differences obtained for the average radial velocity. These differences will be explained in paragraph 6.

## 6. Comparison of calculation results for three software implementations

To compare the results of the calculations obtained by three software implementations, a trajectory of a single particle, not exceeding the limits of the accelerator path, was constructed. Then the dependencies of the relative error of the radial coordinate, the relative error of the longitudinal velocity and the relative error of the radial velocity on the longitudinal coordinate were plotted, they are shown in Fig. 9 – 11.

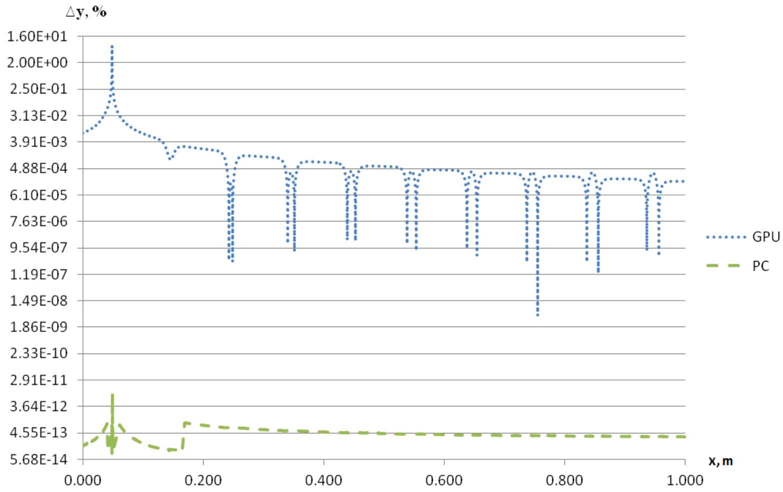

Fig. 9. Dependencies of the relative error of the radial coordinate from the longitudinal for a single particle, calculated using a GPU accelerator (GPU) and a uniprocessor personal computer (PC).

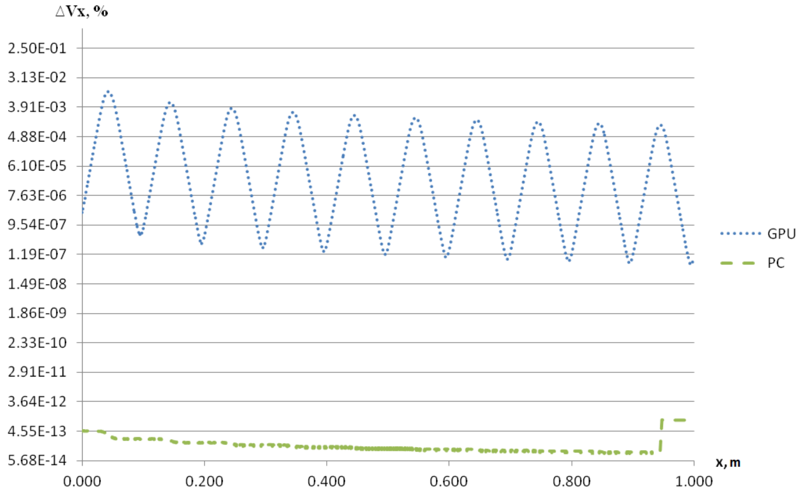

Fig. 10. Dependencies of the relative error of the longitudinal velocity at the coordinate *x* for a single particle, calculated using the GPU accelerator (GPU) and a uniprocessor personal computer (PC).

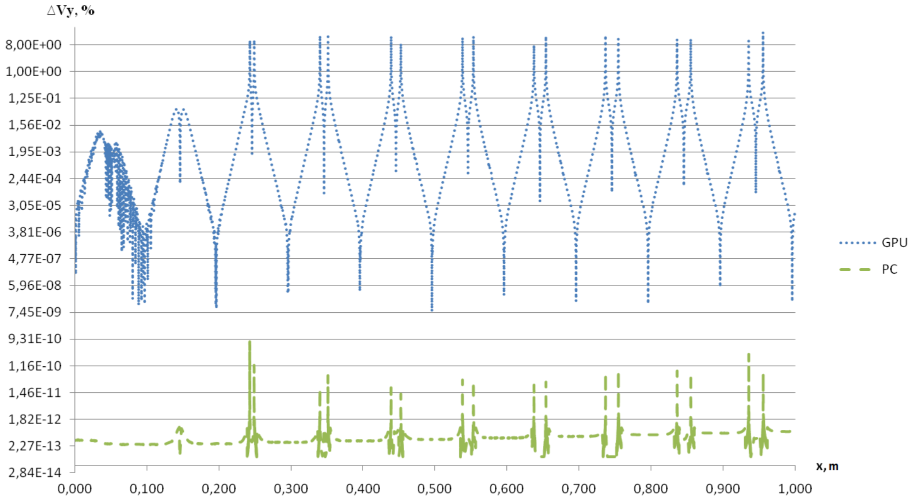

Fig. 11. Dependencies of the relative error of the radial velocity at the coordinate *x* for a single particle, calculated using the GPU accelerator (GPU) and a uniprocessor personal computer (PC).

To calculate the relative error, the data obtained from the supercomputer SK were used as reference values.

Relative errors for data obtained when calculating on a personal computer do not exceed $10^{-10}$ % and, in fact, are a rounding error. The data obtained on the GPU accelerator has a greater relative error, which is due to the aspects of the representation of double precision numbers in the GPU accelerator memory. This is especially noticeable at the transition points of the radial coordinate and radial velocity curves through zero. At these points, the relative computational error is maximal.

## 7. Conclusion

The shown accuracy for all three methods is sufficient to adequately simulate the behavior of the particle flow.

The advantages of the software implementation for a supercomputer and a personal computer include greater accuracy. However, the execution time of the programs in this case strongly depends on the number of available processor cores. So, on 16 nodes of the supercomputer SK, a package of 16384 particles was calculated in 4 min 17 s, and on a single-processor machine in single-threaded mode the calculation took 46 min 34 s.

The software implementation for the GPU accelerator (2,880 cores at 1020 MHz) has a bit lower calculation accuracy. However, this disadvantage is not significant, since the greatest relative error arises for near-zero velocities for a very small number of iterations and, in the final analysis, practically does not affect on the result. For example, for the above particle, for 250,000 iterations, the difference occurs only in the 9-significant digit. In addition, this disadvantage is largely offset by the speed of this implementation of the software: the same package on the GPU accelerator was calculated in 40.64 s.

Since the algorithm for solving this task is parallel in input parameters due to the fact that the trajectories of the particles do not depend on each other, the computational speed is scaled in proportion to the number of particles and the number of processors involved. Thus, the most optimal is the approach in which the primary analysis of alternate designs occurs using the GPU accelerator, with the final verification of the selected design carried out by the supercomputer.

## References

[1] Semkin ND, Kalaev MP, Telegin AM, Piyakov AV, Rodin DV. Multilayer film structures under the influence of micrometeoroids and space debris particles. Applied Phisics 2012; 2: 104–115.

[2] Suhachev KI. Rail electromagnetic accelerator with an external magnetic field. Bulletin of the Samara State Aerospace University 2015; 14(1): 177–189. DOI: 10.18287/1998-6629-2015-14-1-177-189.

[3] Semkin ND, Piyakov AV, Voronov KE, Bogoyavlenskij NL, Goryunov DV. Linear accelerator for simulation of micrometeorites. Devices and equipment of experiment 2007; 1: 1–8.