Simulation of Space Charge Dynamics in High Intensive Beams on Hybrid Systems

Nataliia Kulabukhova(✉), Serge N. Andrianov, Alexander Bogdanov, and Alexander Degtyarev

Saint-Petersburg State University, Saint Petersburg, Russia
n.kulabukhova@spbu.ru

Abstract. The method for construction of analytical expressions for electric and magnetic fields for some set of the distributions of the charge density is described. These expressions are used for symbolic computation of the corresponding electric and magnetic fields generated by the beam during the evolution in accelerators. Here we focus on the use of the matrix form for Lie algebraic methods for calculating the beam dynamics in the presence of self-field of the beam. In particular, the corresponding calculations are based on the predictor-corrector method. The suggested approach allows not only to carry out numerical experiments, but also to provide accurate analytical analysis of the impact of different effects with the use of ready-made modules in accordance with the concept of Virtual Accelerator Laboratory. To simulate the large number of particle distributed resources for computations are used. Pros and cons of using described approach on hybrid systems are discussed. In particular, the investigation of overall performance of the predictor-corrector method is made.

1 Introduction

It is not necessary to say how popular and important accelerators are presently. There are a lot of facilities all over the world created for different purposes. The number of various software packages based on different approaches is even bigger that the accelerators we have. It is better to give a brief survey of methods that are used today to calculate the dynamics of beam with space charge.

High intensive beams are interesting from both side – the theoretical and practical point of view. More particles we have, more information about the beam we will get. On the other hand, intensive beams play a great role in medicine, when needed to irradiate only diseased cells, but not the healthy ones. But it is obvious that with intensity different effects of the beam that can not be denied occur. On of this is the forces of the self field of the particles. In the works [1–4] pay attention on the impact of space charge forces especially in the case that it can lead to the so called the filamentation effect or to the Halo (e.g. see Fig. 1). And for that purposes it is important to consider the space charge forces.

The work is supported by SPbSU 0.37.155.2014 and RFBR 16-07-01113A.

O. Gervasi et al. (Eds.): ICCSA 2016, Part I, LNCS 9786, pp. 284–295, 2016.
DOI: 10.1007/978-3-319-42085-1_22

Fig. 1. (a) Filamentation, (b) Halo

The method which is commonly used [5–7] is Particle-in-Cell method (PIC). The methods popularity caused its conceptual simplicity, the relative ease at which simulations may be implemented. Often, PIC simulations are implemented from first-principles (without the need for an approximate equation of state). However, these simulations often are computationally expensive with restrictive time step and mesh spacing limitations [8].

The Fortran-based environment COSY INFINITY [9] is also well known and used. The main use of the code lies in the field of nonlinear dynamics, where it is used for the computation of perturbation expansions of Poincare maps to high orders as well as their analysis based on normal forms and other methods.

Another approach is given by Alex Dragt and his team. In [10] is said that Lie algebraic methods may be used for particle tracking around or through a lattice and for analysis of linear and nonlinear lattice properties. When used for tracking, they are both exactly symplectic and extremely fast. Tracking can be performed element by element, lump by lump, or any mixture of the two. (A lump is a set of elements combined together and treated by a single transfer map.)

In addition to single-particle tracking, Lie algebraic methods may also be used to determine how particle phase-space distribution functions evolve under transport through both linear and nonlinear elements. These methods are useful for the self-consistent treatment of space-charge effects and for the study of how moments and emittances evolve.

MARYLIE [11] is a FORTRAN program for beam transport and tracking based on a Lie algebraic formulation of charged particle trajectory calculations. This software is useful for the design and evaluation of both linear transport systems and circulating storage rings. The program is able to compute transfer maps and trace rays through single or multiple beamline elements for the full six-dimensional phase space without the use of numerical integration or traditional matrix methods. The effects of high-order aberrations are computed as an integral part of the Lie algebra approach. All non-linearities, including chromatic effects, through third order are included [12].

The number of methods and software is not restricted by these once. There are TRANSPORT, BEAMBEAM3D, IMPACT-Z, IMPACT-T, MAD [13] and

some others. But in all these methods the trajectory of one particle is calculated. And in case of intensive beams the number of particles to count is bigger then 1 billion. Though, the computer resources allow us to calculate large amount of data, the practice shows that it is better to have a parallel algorithm at the beginning than a good machine. That was our goal to make the algorithm that can be parallelized easily.

2 Lie Algebra in Accelerator Physics

The approach about which we will speak is very similar to the one on what MARYLIE is based. The evolution operator for dynamic systems is used in theoretical physics for a long time. This operator can be written in general form as Lie operator (see, for example, [10]):

$$\frac{d\mathcal{M}(\boldsymbol{U}(t), t|t_0)}{dt} = \mathcal{L}(\boldsymbol{U}(t), t) \circ \mathcal{M}(\boldsymbol{U}(t), t|t_0), \mathcal{M}(t_0|t_0) = \mathcal{I}d \;\forall\; t_0 \in [T, t_0]. \quad (1)$$

These operators define the Lie transformations $\mathcal{M}(\boldsymbol{U}(t), t|t_0)$, generated by an infinitesimal Lie operator $\mathcal{L}(\boldsymbol{U}(t), t)$ (the vector field of the dynamical system), where $\boldsymbol{U}(t)$ is a vector of control functions (for simplicity, we will omit the argument of $\boldsymbol{U}(t)$). Note that the Eq. (1) generally has the form of a nonautonomous linear operator equation. The equality 1 in the integral form has the following form

$$\mathcal{M}(t|t_0) = \mathcal{I}d + \int_{t_0}^{t} \mathcal{L}(\tau) \circ \mathcal{M}(t|\tau) d\tau. \quad (2)$$

The general solution can be written in the form of a chronological series of Volterra [14]

$$\mathcal{M}(t|t_0) = \mathcal{I}d + \sum_{k=1}^{\infty} \int_{t_0}^{t} \int_{t_0}^{\tau_1} \dots \int_{t_0}^{\tau_{k-1}} \mathcal{L}(\tau_k) \circ \mathcal{L}(\tau_{k-1}) \circ \dots \circ \mathcal{L}(\tau_1)\, d\tau_k \dots d\tau_1. \quad (3)$$

or using the Magnus presentation [15] this equality can be written as

$$\mathcal{M}(t|t_0) = \exp \mathcal{W}(t|t_0; \mathcal{L}). \quad (4)$$

Here $\mathcal{W}(t|t_0; \mathcal{L})$ – a new vector field, generated by the "old" vector field $\mathcal{L}$. In Ref. [14] analytical expressions (for step-by-step calculations) for the new operator are presented $\mathcal{W}(t|t_0; \mathcal{L})$ using "nested" series

$$\mathcal{W}(t|t_0) = \int_{t_0}^{t} \mathcal{V}(\tau)d\tau + \alpha_1 \int_{t_0}^{t} \left\{ \mathcal{L}(\tau), \int_{t_0}^{\tau} \mathcal{L}(\tau') \right\} d\tau + \\ + \alpha_1^2 \int_{t_0}^{t} \left\{ \mathcal{L}(\tau), \int_{t_0}^{\tau} \left\{ \mathcal{L}(\tau'), \int_{t_0}^{\tau'} \mathcal{L}(\tau'')d\tau'' \right\} d\tau' \right\} d\tau + \\ + \alpha_1\alpha_2 \int_{t_0}^{t} \left\{ \left\{ \mathcal{L}(\tau), \int_{t_0}^{\tau} \mathcal{L}(\tau')d\tau' \right\}, \int_{t_0}^{\tau} \mathcal{L}(\tau')d\tau' \right\} d\tau + \dots \quad (5)$$

In the work [14] the necessary conditions for the convergence of the corresponding series as well as the convergence rate are described. Thus, these relations allow us to find correct solutions of nonlinear operator equations in the form of convergent series (under some relatively simple assumptions). However, the operator form for practical solutions cannot be used in computations, so we have to choose some functional basis (in our case we use the well-known Poincare-Witt basis), which can provide appropriate solutions in the form of the following equality:

$$\mathcal{M}(t|t_0) \circ \boldsymbol{X}_0 = \sum_{k=1}^{\infty} \mathbb{M}^{[1k]}(t|t_0)\boldsymbol{X}_0^{[k]}, \quad \boldsymbol{X}_0 = \boldsymbol{X}(t_0), \quad (6)$$

where $\boldsymbol{X}$, $\boldsymbol{X}_0$ are vectors of current and initial phase coordinates of a particle, $\mathbb{M}^{[1k]}(t|t_0)$, $k \geq 1$ are matrices (two dimensional arrays) responsible for the nonlinearity of k-th order in the solution of the equation of evolution. Thus, the task of investigating the evolution of a nonlinear system is reduced to the computation of the matrices $\mathbb{M}^{[1k]}(t|t_0)$ up to the necessary order of nonlinearity with the corresponding estimates of accuracy [14]. So in the absence of effect of space-charge, we can compute corresponding matrices in the nonlinear approximation step by step up to the desired order of nonlinearity. However, taking the space charge into account, the matrices $\mathbb{M}^{[1k]}(t|t_0)$ can be computed using the method of successive approximations, if necessary [14,16].

It should be noted, that the knowledge of the matrix $\mathbb{M}^{[1k]}$ up to the required order of nonlinearity allows to calculate the dynamics of the beam as an ensemble of particles with the given accuracy. Indeed, the equality (6) allows to describe the particle beam using various forms of its description. Let's consider an ensemble of particles $\mathfrak{M}_0$ consisting of N particles at some initial time. Then the beam evolution may be described by different methods:

- with the help of a matrix phase states beam $\mathbb{M}_0^N = \left\{\boldsymbol{X}_0^1, \boldsymbol{X}_0^2, \dots, \boldsymbol{X}_0^N\right\}$,
- an envelopes matrix σ_0,
- or a particle distribution function $f(\boldsymbol{X}, 0) = f_0(\boldsymbol{X})$, $\boldsymbol{X}_0^k \in \mathfrak{M}_0$, $\forall k = \overline{1, N}$.

All these methods of particle beam description can be considered a base for forming information objects that characterize the state of the beam at the initial and current moments. About two last methods we will speak later.

On the next step we should introduce information objects that are responsible for evolution of the initial state. In our approach we use the matrices $\mathbb{M}^{[1k]}$, $\forall k \geq 1$. These matrices can be calculated according to the Lie formalism [10,14]. We should note that the group property for the evolution operator allows us to calculate the operator successively (step-by-step) for each control element of the lattice (here we refer to the control elements, such as dipoles, multipole lenses, free spaces and etc.) and for the accelerator system in the whole.

Moreover, the map that describes the impact of a control element can in turn be represented as a series of maps that reflect the impact of electromagnetic fields [10] up to nonlinearities of necessary order. These properties of the evolution operator and its consequent effect on control systems allows us to introduce information objects, each of which is responsible for mapping generated by a particular control element as a set of particular units in the defined sequence. So, for each control element (multipoles, drifts and etc.) we can calculate the necessary matrices $\mathbb{M}^{[1k]}$ and then to construct these matrices for some lattices using some concatenation procedure (see, e.g. [17]).

Naturally, beside the corresponding objects, we have a set of mathematical rules with which they act on the data objects, characterizing the state of the beam in the initial or current moments. However, these objects themselves consist of a set of virtual subagents (for example, responsible for the fringing fields or some other characteristics of control fields).

In other words, we have an additional internal subset of subagents [18] designed to study the effects of additional characteristics of the transport system on beam behaviour. It should be noted that these objects themselves do not have autonomy from the physical point of view. Introduction of such objects is justified by the fact that their use provides the necessary degree of flexibility and performance from a computational point of view. Thus, physical objects that are responsible for the impact on the ensemble of particles are in turn divided into a set of subagents, which are responsible for certain properties, but do not have an independent physical interpretation. An assembly of these sub-objects helps to secure the necessary variability and implement the optimization of control system as a whole. In other words we can change the necessary subagent without distortion of physical sense and computational sequence. Moreover, you must also monitor the state of the beam itself (values of beam envelopes, polarization, etc.), because these characteristics are very important for realization of the optimal working regime. Besides, we should carry out necessary additional computational procedures to analyze the impact of the effects of control errors on the beam characteristics. These additional computing can be also realized using additional subagents. All necessary properties of agents and subagents should be divided on physical properties (derived from physical characteristics of a primary physical object) and information properties in according to general concept of forming of information agents.

3 Self-consistent Particle Dynamics with Space Charge Forces

In this part the predictor-corrector method based on matrix formalizm will be discussed. By saying predictor-corrector we mean a multiple step method. This one can be used as an alternative to the well known Runge-Kutta method. The scheme of this method is simple. First, the extrapolation method is used to predict the value of some y_{j+1} by known previous y_j, y_{j-1}, etc. Then the obtained value is estimated and is correcting to get the better approximation of y_{j+1}. If the difference between the correcting and the predicting values exceeds a certain value, then the next iteration is running [19].

In our case the main idea of this method is to predict the distribution of the particles, which we want to get at the end, by correcting the intermediate results during the calculations. First, let's talk about the solution algorithm of self-consistent dynamic of the beam in general. Here the distribution functions will be discussed.

At the beginning we set an interval $T = [t_0, t_1]$, on which the solution is looking for, and $\Delta t = t_1 - t_0$. The transportation system is given on this interval, that means that the external fields $\mathbf{B}^{ext}$, $\mathbf{E}^{ext}$ and the function $\mathbf{F}^{ext}$ can be obtained.

Besides, the distribution function is selected from the set of initial distributions. In the simple case it can be the ideal Kapchinskij-Vladimirslij distribution. Or it can be the modifications of KV, which are given in [20]. It is a base distribution in such cases, because K-V distribution is a four dimensional distribution in phase space and has properties:

- the space charge forces are linear;
- it transforms into a K-V distribution under a linear mapping.

With other distributions this one forms something like the class of initial distributions:

- linear: $\rho_1(x, y) = \rho_0(1 - 4\varkappa^2/9)\Theta(1 - 4\varkappa^2/9))$;
- uniform: $\rho_2(x, y) = \rho_0\Theta(1 - \varkappa^2)$;
- normal: $\rho_3(x, y) = \rho_0 exp(-\alpha_3^2\varkappa^2)$, $\alpha_3 = -\frac{\pi}{2\ i\ erf(i)}$.

  Where $erf(x) = \int_0^x exp(-t^2/2)dt$ - probability integral.
- quadratic: $\rho_4(x, y) = \rho_0(1 - (4/5)^4\varkappa^4)\Theta(1 - (4/5)^4\varkappa^4)$;
- co-sinusoidal: $\rho_5(x, y) = \rho_0\cos^2(\pi\alpha_5^2\varkappa^2)\Theta(1 - \alpha_5^2\varkappa^2)$.

  Where α_5 calculates with Frenel integral.

On Fig. 2 the density function ρ_i, $i = \overline{1,5}$ as the function of scalar variable R is shown. See the [16] for more details. This list can be supplemented by other distributions depending on the task.

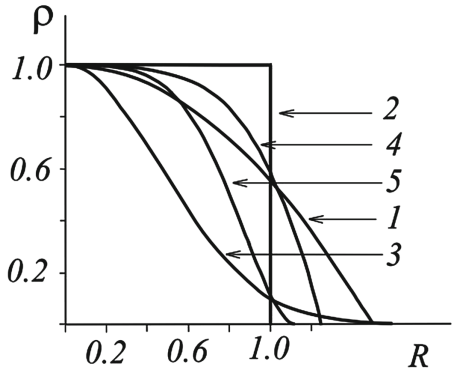


Fig. 2. Distribution of charge density: 1 - linear, 2 - uniform, 3 - normal, 4 - quadratic, 5 - co-sinusoidal.

Now we are turning back to the algorithm. After all the initial parameters are set, the evolution operator is calculated using the Eq. 3 and the technology as described above.

After that the distribution function is obtained with the help of evolution operator applying on the previous value of distribution:

$$f^1(\mathbf{X}, t) = f_0((\mathcal{M}^0)^{-1} \circ \mathbf{X}_0).$$

Substituting the obtained function to the field equations, we get $(\mathbf{B}^{self})^1$, $(\mathbf{E}^{self})^1$.

Now we are ready to calculate the function

$$(\mathbf{F}^{self})^1 = \mathbf{F}^{self}(\mathbf{B}^{self})^1, (\mathbf{E}^{self})^1, \mathbf{X}, t)).$$

Or the self-consistent Hamiltonian $(\mathcal{H}^{self})^1 = \mathcal{H}^{self}((\mathbf{B}^{self})^1, (\mathbf{E}^{self})^1, \mathbf{X}, t)$ is found.

Thereafter, the evolution operator $\mathcal{M}^1 = \mathcal{A} \circ \mathcal{M}^0$ is evolved by equation

$$\mathcal{M}(t|t_0; \mathcal{V}^{ext} + \mathcal{V}^{self}) = \mathcal{I}d + \int_{t_0}^{t} (\mathcal{V}^{ext}(\tau) + \mathcal{V}^{self}(\tau)) \circ \mathcal{M}(\tau|t_0; \mathcal{V}^{ext}(\tau) + \mathcal{V}^{self}(\tau)) d\tau. \tag{7}$$

The new average value of distribution function is found by the following equation:

$$\langle f(\mathbf{X}, t_0) \rangle^1_{\mathfrak{M}_1} = (1 - \alpha) \langle f_0((\mathcal{M}^0)^{-1} \circ \mathbf{X_0}) \rangle_{\mathfrak{M}_0} + \alpha \langle f_0((\mathcal{M}^0)^{-1} \circ \mathbf{X_0}) \rangle_{\mathfrak{M}_0}, \tag{8}$$

The final step is to verify the criteria, e.g.:

$$||\mathcal{M}^k - \mathcal{A} \circ \mathcal{M}^{k-1}|| < \epsilon, k \geq 1. \tag{9}$$

It is obvious that if the condition 9 is carried out, the solution is obtained. Otherwise, the $(\mathbf{B}^{self})^i$, $(\mathbf{E}^{self})^i$ are calculating again.

This algorithm is suitable for analytical analysis, but on practice we can not measure the distribution function. To get the result that can be proved by experiment the algorithm based on envelope dynamic was designed.

Similarly to the previous algorithm we set an interval $T = [t_0, t_1]$, $\Delta t = t_1 - t_0$, and $f(\mathbf{X}, t_0) = f_0(X)$ is the initial value of the distribution function consisting of the set of phase points when $t = t_0$, and N is the order of approximation.

First step. We calculate matrices $\mathfrak{S}_0^{ik}$, $i, k = \overline{1, N}$ by the following equation:

$$\mathfrak{S}_0^{ik} = \int_{\mathfrak{M}_0} f_0(\mathbf{X}) \mathbf{X}^{[i]} (\mathbf{X}^{[k]})^* d\mathbf{X}$$

As a form-matrix A_0 $(\mathfrak{S}_0^{11})^{-1}$ can be chosen, or $\mathfrak{S}_0^{-1}$ – if the initial set $\mathfrak{M}_0$ is an ellipsoid with the border

$$X_0^* \mathfrak{S}_0^{-1} \mathbf{X}_0 = \varepsilon.$$

Next we built approximant $\varphi_0(\varkappa_0^2) \approx f_0(\mathbf{X}_0)$, where $\varkappa_0^2 = \mathbf{X}_0^* A_0 \mathbf{X}_0$ and turn to the next step.

Second step. Here we get the block-matrices $\mathbb{P}^{1k}(B^{ext}, E^{ext}, t)$ and $\mathbb{N}_1^{1k} = \mathbb{P}^{1k}(\mathbf{B}^{ext}, \mathbf{E}^{ext}, t)$ [14] for external fields. The (ij) element of matrix $\mathbb{P}^{1k}$, for example, can be found by the following form:

$$\left\{\mathbb{P}^{1k}(t)\right\}_{ij} = \frac{1}{k_1! \dots k_n!} \left. \frac{\partial^k \mathbf{F}_i(X_j, t)}{\partial x_1^{k_1} \dots \partial x_n^{k_n}} \right|_{x_1 = \dots = x_n = 1}$$

Third step. It is necessary to find $\mathbf{E}^{self} = \mathbf{E}(\varphi_0(\varkappa_0^2))$ depending of the distribution function we have chosen (e.g. uniform, normal, etc.).

On the fourth step we calculate block-matrices $\mathbb{P}^{1k}(\mathbf{E}^{self}, t)$ with space charge effect: $\mathbb{N}_2^{1k} = \mathbb{P}^{1k}(\mathbf{E}^{ext}, t)$

Fifth step. Then comes the calculations of block-matrices $\mathbb{M}^{ik}$ where $i \le \overline{k \le N}$,

$$\mathbb{M}_1^{ik} = \mathbb{M}^{ik}(t|t_0; \{\mathbb{N}_1^{1l}\}), l = \overline{1, k},$$

$$\mathbb{M}_2^{ik} = \mathbb{M}^{ik}(t|t_0; \{\mathbb{N}_2^{1l}\}),$$

$$\mathbb{M}_0^{ik} = \mathbb{M}_1^{ik} + \mathbb{M}_2^{ik}.$$

Block-matrices $\mathbb{M}^{ik}$ are the matrix form of the evolution operator.

On the next step, after all necessary matrices have been obtained, we can substitute them into block-matrices $\mathfrak{S}_0^{ik}$

$$\mathfrak{S}_0^{ik} = \sum_{l=i}^{\infty} \sum_{j=k}^{\infty} \mathbb{M}_0^{il} \mathfrak{S}_0^{lj} \left(\mathbb{M}_0^{jk}\right)^T.$$

Step seven. Before the conditions will be checked the virtual changes of settings while beam evolution must be made:

$$\mathfrak{S}_1^{ik} = \alpha\mathfrak{S}_0^{ik} + (1-\alpha)\mathfrak{S}_0^{ik}, 0 < \alpha < 1.$$

Virtual change implies changes of settings that are necessary to built a map. Envelope matrices, functions of distributions, etc., are not changed.

Step eight. Now we can check the conditions:

$$\frac{2\|\mathfrak{S}_1^{ik} - \mathfrak{S}_0^{ik}\|}{\|\mathfrak{S}_1^{ik}\| + \|\mathfrak{S}_0^{ik}\|} < \varepsilon^{ik}. \tag{10}$$

Different equivalent rules can be used as 10. If the condition is right, the process stops. Otherwise:

$$\mathfrak{S}_0^{ik} = \mathfrak{S}_1^{ik},$$

and before turning back to the algorithm the final step is to find the approximate value $\varphi(\varkappa^2)$ for function $f(\mathbf{X}, t)$:

$$\varphi(\varkappa^2) \approx f_0(\mu_0^{-1} \circ \mathbf{X}_0) = f_0(\sum_{i=1}^{\infty} \mathbb{T}_0^{1l} X_0^{[i]}).$$

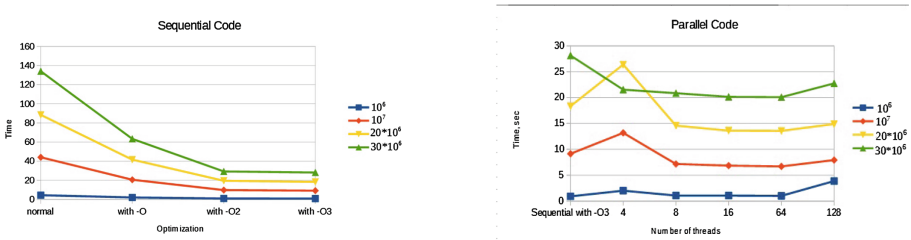
Assuming that $\varphi_0(\varkappa^2) = \varphi(\varkappa^2)$ return to the step three.

Considered algorithm can be simplified by using as approximate value the function that is constant on the ellipsoid and zero outside it. After that step three is modified to be easier and the final changes are not needed at all. That significantly accelerates the process. Moreover, choosing the approximate value from the class of polynomials allows us to use pre-computed block-matrices from special database. So this approach can significantly reduce the computations instead of numerical simulations.

4 Parallelization of Predictor-Corrector Method

Practically for every one it is obvious that the number of particles needed to compute on practice can not be provided by any known approach. The natural parallelization and distributed structures of beam physics problems allow using parallel and distributed computer systems (see works [21–23]. Analyzing the situation, it became clear that it is no matter how much resources we have, if the algorithm is not suitable for parallelization there will be not great benefit of it. As we can see from the algorithm shown above matrix formalism is a high-performance mapping approach for ODE solving. It allows to present the intermediate results and the solution of the system in the form of matrices. That makes the approach to be easy implemented in parallel code.

Due to the fact that only matrix multiplication and addition are used, we think of a GPU programming [24]. The present research is shown that there is no great benefit via parallelization of computational code for one particle by using OpenMP library (see Fig. 3 and Tables 1 and 2). In this case overhead



a. b.

Fig. 3. a – Sequential code; b – Parallel code

Table 1. Time (sec.) in sequential code for different number of particles

Optimization/Num. of particles	10^6	10^7	$20 * 10^6$	$30 * 10^6$
normal	4,3942	44.0785	88.42	133.955
-O	2.0449	20.5504	41.6161	63.3152
-O2	0.9780	9.7588	19.4914	29.2429
-O3	0.9114	9.14508	18.3444	28.1367

on data sending is significant and take the greatest part of time. On the other hand, matrix formalism allows to process a set of initial points, where parallelization is more preferable on GPUs. But using only GPUs is not justified. In our experiment we have the system that can provide the power to compute only the number of particles nearly 10^7. It's less that required, but our goal in this part of research was to test the algorithm before using in on the real system. The results has shown good parallelization of the described approach.

Table 2. Time (sec.) in parallel code for different number of particles

Num. threads/Num. of particles	10^6	10^7	$20 * 10^6$	$30 * 10^6$
Sequential with -O3	0,911436	9,14508	18,3444	28,1367
4	2,00863	13,1787	26,3788	21,5358
8	1,06923	7,16977	14,5777	20,8549
16	1,06208	6,8548	13,6071	20,15
64	1,00906	6,70448	13,5593	20,0794
128	3,86119	7,92809	14,8894	22,748

5 Conclusion

Our challenge is to provide computer simulation for developed algorithm for solving the problem of accounting space-charge forces in general and compare

this algorithm with other methods. It allows simulate both long-term evolution of a set of particles, and evaluating based on envelope description. As it was said above the method can be implemented in parallel codes on GPU+CPU hybrid Cluster. That is why the future development of the research also can be based on writing software to compare different parallel techniques for Hybrid Systems, in order to effective use of described approach to compute the required number of particles in long-term evolution of the beam.

Acknowledgments. The authors would like to express gratitude to Vladimir Korkhov for valuable help. Scientific research were performed using the equipment of the Research Park of St.Petersburg State University. The work was sponsored by the Russian Foundation for Basic Research under the projects: 16-07-01113 "Virtual supercomputer as a tool for solving complex problems" and by the Saint-Petersburg State University under the project 0.37.155.2014 "Research in the field of designing and implementing effective computational simulation for hydrophisical and hydrometeorological processes of Baltic Sea (and the open Ocean and offshores of Russia)".

References

1. Batygin, Y.K., Scheinker, A.: Suppresion of halo formation in fodo channel with nonlinear focusing. In: Proceedings of IPAC 2013. JACOW (2015)
2. Batygin, Y.K., Scheinker, A., Kurennoy, S.: Nonlinear Optics for Suppresion of Halo Formation in Space Charge Dominated Beams (2015)
3. Batygin, Y.K.: Space-charge neutralization of 750-keV proton beam in lansce injector LIN. In: Proceedings of IPAC 2015. JACOW (2015)
4. Ryne, R.D., Habib, S., Wangle, T.P.: Halos of Intense Proton Beams. IEEE (1996)
5. Paret, S., Qiang, J.: Collisional effects in particle-in-cell beam-beam simulation. In: Proceedings of IPAC 2013. JACOW (2013)
6. Wolfheimer, F., Gjonaj, E., Weiland, T.: Parallel particle-in-cell (PIC) codes. In: Proceedings of ICAP 2006. JACOW (2006)
7. Stancari, G., Redaelli, S., Moens, V.: Beam dynamics in an electron lens with the warp particle-in-cell code. In: Proceedings of IPAC 2014. JACOW (2014)
8. Bowers, K.J.: Accelerating a paticle-in-cell simulation using a hybrid counting sort. J. Comput. Phys. **173**, 393–411 (2001). Academic Press
9. Makino, K., Berz, M.: COSY INFINITY Version 9. Nuclear Instruments and Methods A558 (2005)
10. Dragt, A.J.: Lie methods for nonlinear dynamics with applications to accelerator physics. University of Maryland (2015)
11. Dragt, A.J., Ryne, R.D., et al.: MARYLIE 3.0 Users Manual: A Program for Charged Particle Beam Transport Based on Lie Algebraic Methods. University of Maryland (2003)
12. Dragt, A.J., Ryne, R.D., et al.: Numerical computation of transfer maps using lie algebraic methods. In: Proceedings of PAC 1987 (1987)
13. Ryne, R.D.: Advanced computing tools and models for accelerator physics. In: Proceedings of EPAC 2008 (2008)
14. Andrianov, S.N.: Dynamical Modeling of Control Systems for Particle Beams'. Saint Petersburg State University, SPb (2004)

15. Magnuss, W.: On the exponential solution of differential equations for a linear operator. Comm. Pure Appl. Math. **7**(4), 649–673 (1954)
16. Kulabukhova, N., Degtyatev, A., Bogdanov, A., Andrianov, S.: Simulation of space charge dynamics on HPC. In: Proceedings of IPAC 2014. JACOW (2014)
17. Healy, L.M., Dragt, A.J.: Concatenation of Lie algebraic maps. Lie Methods in Optics II. Lect. Notes in Physics, vol. 352 (1989)
18. Andrianov, S., Kulabukhova, N.: Lie algebraic methods as mathematical models for high performance computing using the multi-agent approach. In: Gervasi, O., et al. (ed.) ICCSA 2016, Part I. LNCS, vol. 9786, pp. 418–430. Springer, Heidelberg (2016)
19. Szilagui, M.: Electron and ion optics (in russian). Mir, Moscow (1990)
20. Venturini, M.: Lie methods, exact map computation, and the problem of dispertion in space charge dominated beams. Ph.D. thesis (1998)
21. Giovannozzi, M.: Space-Charge Simulation Using Parallel Algorithms
22. Bowers, K.J.: Accelerating a particle-in-cell simulation using a hybrid counting sort. J. Comput. Phys. **173**, 393–411 (2001)
23. Qiang, J., Ryne, R.D., Habib, S., Decy, V.: An object-oriented parallel particle-in-cell Code for beam dynamics simulation in linear accelerators. J. Comput. Phys. **163**, 434–451 (2000)
24. Kulabukhova, N.: GPGPU implementation of matrix formalism for beam dynamics simulation. In: Proceedings of ICAP 2012. JACOW (2012)