# Ef: Software for Nonrelativistic Beam Simulation by Particle-in-Cell Algorithm

*A. Yu.* Boytsov[1,⋆] and *A.A.* Bulychev[2]

[1] *Veksler And Baldin Laboratory of High Energy Physics, Joint Institute for Nuclear Research, 141980 Dubna, Moscow region, Russia*
[2] *Independent Researcher, Moscow, Russia*

**Abstract.** Understanding of particle dynamics is crucial in construction of electron guns, ion sources and other types of nonrelativistic beam devices. Apart from external guiding and focusing systems, a prominent role in evolution of such low-energy beams is played by particle-particle interaction. Numerical simulations taking into account these effects are typically accomplished by a well-known particle-in-cell method. In practice, for convenient work a simulation program should not only implement this method, but also support parallelization, provide integration with CAD systems and allow access to details of the simulation algorithm. To address the formulated requirements, development of a new open source code – *Ef* – has been started. It's current features and main functionality are presented. Comparison with several analytical models demonstrates good agreement between the numerical results and the theory. Further development plans are discussed.

## 1 Introduction

Nonrelativistic electron and ion beams are widely used in nuclear and particle physics, nanotechnology, microelectronics, medicine and other areas of science and technology. Numerical simulations of charged particles dynamics is essential step in order to achieve required device parameters. Nonrelativistic beams have a range of specific properties that simulation software should take into account.

First, at low energies particle-particle interaction (space charge) plays prominent role and has to be considered alongside with external fields. Particle-in-cell algorithm is typically used for this purpose (see the next section for more details). Besides, nonrelativistic kinematics can be utilized to simplify calculations without too much loss in accuracy, whereas relativistic regime can be provided as an option. Due to large number of particles (tens of millions and more) and long time durations to simulate (milliseconds, while characteristic time scale is nanoseconds) the problems demand extreme amounts of computational resources. Parallelization is the only means to provide them, and, therefore, it is required to accomplish the modelling in reasonable amount of time.

Another important aspect is device geometry, which plays major role at low beam energies. Integration with CAD systems is highly desirable to simplify geometry specification. A useful feature would be an option to automate search for optimal device configuration and it's various parameters.

[⋆] e-mail: boytsov@jinr.ru

For practical means, such properties as intensity, emittance, current fluctuations, particle energy spectrum, phase profiles and particle trajectories are of most interest. An ideal program should provide tools to display and analyze them easily.

To comply with principle of reproducible research [1], the source code of the program should be open and available for examination.

Apart from these general requirements, there are application-specific concerns. In particular, authors' motivation stems from modelling of electron beam ion sources (EBIS). Such sources are commonly used to produce high-density beams of highly charged ions. Their principle of operation employs an electrostatic trap to hold the ions while ionizing them to desired charge state within an electron beam. In EBIS different spatial and time scales are involved: first, slow ions and fast electrons have incomparable time evolution scales, second, transversal size of the beam (less than 1 mm radius) is much less than longitudinal trap length (50-100 cm). Besides, various instabilities may form that can destroy steady state of the electron beam or heat confided ions, leading to ion losses and lack of ionization efficiency. Such properties require fine control over details of the simulation algorithm so that it is possible to modify and experiment with it.

A representative list of existing software implementing particle-in-cell method can be found at [2, 3]. Warp, PIConGPU and Smilei [4–6] can be mentioned specially: all of them are professionally developed and overall of very high quality. Unfortunately, while source code of these programs is open, it seems that massive changes in the architecture would be required to incorporate certain necessary features. Apart from that, there are several freely distributed (SNOP [7]) and commercial packages (CST-Studio, Opera3d, SIMION, EGUN-IGUN, etc). However, they do not provide access to the source code which makes it impossible to customize them.

To address the aforementioned requirements, development of a new program – named *Ef* – specialized for low-energy beams modelling has been started [8]. While writing new code is a laborious task, it is an optimal solution in the long-term perspective. Currently the program focuses on simulation of electron beam ion sources, but it also should be useful for modelling of other low-energy charged particle beam and (possibly) plasma devices.

## 2 Implementation Details

The basis of the created program is a so-called "particle-in-cell" method. Initially suggested in the 1950s for hydrodynamics simulations and popularized in the 1960s for plasma research, it is well described in the literature [9, 10]. The method aims to model dynamics of a large number of charged particles in external electromagnetic fields taking into account particle-particle interaction. First, the equations of motion of individual particles are discretized, so that time evolution can be viewed as a sequence of time steps. At each time step, forces acting on particles consist of two components: the external fields and particle self-interaction. The external fields are considered preset. As for the particle-particle interaction, the idea is to approximately compute electromagnetic fields generated by the particles instead of calculating binary Coulomb forces between each pair of them. Such collective fields are obtained by numerically solving Maxwell's equations, which requires to perform spatial discretization of the simulation volume, approximate contribution of each particle to charge density and current at each node, and solve for potentials on the mesh using finite difference or finite element method. After the combined action of the external and particle-generated fields on each particle is computed, integration of the dynamics equations is performed.

The current version of the program [8] implements nonrelativistic dynamics with Boris time integration scheme [11], which is notable for it's long term energy conservation [12].

For interparticle interaction electrostatic particle-in-cell method is used. It takes into account only electric fields generated by particles, so that on each time step the Poisson equation $\Delta\phi = -\rho/\epsilon_0$ is solved. The Laplace operator $\Delta$ is discretized using finite difference scheme. To compute charge density $\rho$, macroparticle's charge is distributed between the nodes of enclosing cell in reverse proportion to distance to the node (for each particle, only enclosing cell nodes are used). Such algorithm is surprisingly efficient for large array of applications, and more advanced versions can be implemented in the future.

To run the simulation, it is necessary to prepare a configuration file describing the simulation domain. Currently the program supports specification of several sources of particles with various properties, such as charge, mass, mean momentum, etc. It is possible to choose between particle-in-cell method and noninteracting particles for computations of space charge. External electric and magnetic fields can be specified, either uniform or by simple analytical expression. It is possible to define inner regions, that absorb particles on collisions and act as conductors under fixed potential. During the computation, the program outputs files with simulation results, containing all the information inside the simulation domain (HDF5 library and file format are used [13]). Such files can be used as a starting point for new computation, which is convenient to resume interrupted execution.

The program is implemented in C++. It's source code is free and open and available for modification and redistribution under MIT license.

## 3 Examples

To test correctness of the code, we have performed comparisons of the numerical results with several analytical models.

First tests have been devoted to evolution of a single particle in free space and uniform electric and magnetic fields. The particle-in-cell algorithm is not used in these computations, and only time integration scheme is tested. Numerical results demonstrate good agreement with the theory [8].

The simplest example with particle-particle interaction is propagation of an electron beam in free space. For a beam with rectangular cross section (ribbon beam) moving along the Z-axis, with "height" (dimension along the Y-axis) much larger than "width" (dimension along the X-axis), under a number of simplifying assumptions it is possible to derive a trajectory of an electron located on the edge of the beam [14] (which can be interpreted as it's envelope):

$$x(t) = x_0 + \frac{qI_0}{4\epsilon_0 m v} t^2, \qquad z(t) = vt.$$

In these relations, $I_0$ is linear current density per unit of "height" and $v$ is particles speed. To check them numerically it is possible to perform a simulation of a beam propagating in free space and plot projections of particles positions on the X-Z plane. The result are shown on the left panel of figure 1.

In practice, it is common to use magnetic field to prevent beam divergence. In the case of ribbon beam propagating in uniform magnetic field, trajectory of the electron on the edge of the beam is given by the following equation [14]:

$$x(t) = x_0 + \frac{I_0 m}{2\epsilon_0 q v B^2}\left(1 - \cos(\omega_g t)\right), \qquad z(t) = vt, \qquad \omega_g = \frac{qB}{m}.$$

Numerical results can be compared with analytical predictions using the same procedure as in the field-free case. The results are shown on the right panel of figure 1. In both cases numerical results are in good agreement with the theory.

Another example we would like to consider is propagation of a cylindrical electron beam inside a conducting tube. By solving Poisson equation

$$\begin{aligned} &\Delta\phi_1 = -\rho/\epsilon_0, \quad 0 \le r \le r_{beam} \\ &\Delta\phi_2 = 0, \quad r_{beam} \le r \le r_{wall} \end{aligned}$$

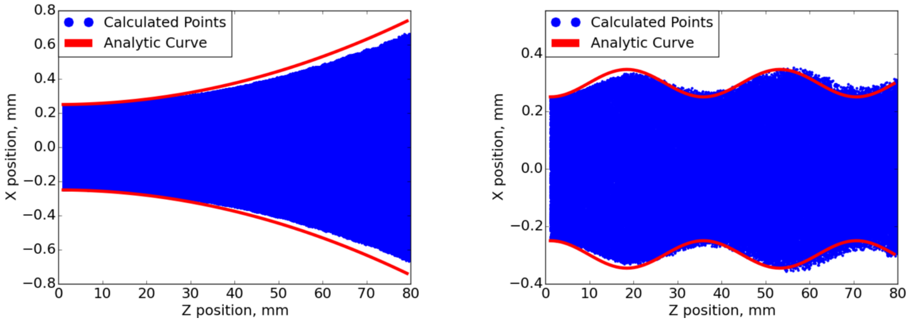

**Figure 1.** Numerical simulations and analytical expressions for envelope of an electron beam propagating in free space (left) and uniform magnetic field (right). Total simulation time is 5 [ns], time step is $5 \cdot 10^{-12}$ [s]. Beam current is 0.008 [A], particles' speed corresponds to 1 [keV] energy. 500 new macroparticles are added each time step. For the right figure, magnetic field is $B = 0.02$ [T].

with boundary conditions $\phi_1(r_{beam}) = \phi_2(r_{beam})$, $\phi_1'(r_{beam}) = \phi_2'(r_{beam})$, $\phi_2(r_{wall}) = U_0$ it is possible to obtain potential distribution inside the tube. Numerical and analytical results are compared on the figure 2. It can be seen, that agreement is decent.

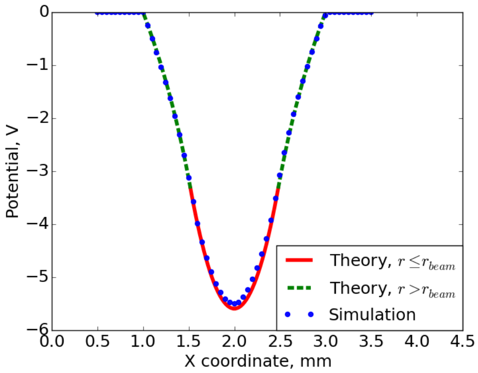

**Figure 2.** Potential well inside a conducting tube with an electron beam. Beam current $I = 0.01$ [A], beam radius $r_{beam} = 0.5$ [mm], inner radius of the tube $r_{wall} = 1$ [mm], and potential on the tube $U_0 = 0$ [V].

## 4 Conclusion

Simulation of nonrelativistic low-energy beams has a number of features that distinguish them from other areas of charged particles modelling. To take them into account, development of new specialized code has been started. The current version of the program has functionality sufficient to perform basic simulation tasks and numerical results are in good agreement with predictions of several simple analytical models.

Speaking of development plans, our current long-term goal is simulation of an electron beam ion source in reflective operation mode [15]. This requires various improvements to physical and mathematical models, such as support for particle-particle magnetic interaction, collisions and ionization, multigrid Poisson solver, advanced macroparticle shapes, etc. From a technical side, top priority is GPU acceleration. Besides, current experimental features: graphical user interface, import of 3d models and precomputed external fields - need to be improved.

Authors would like to thank Dr. E.E. Donets for help and discussion. One of the authors (A.Yu.) would like to acknowledge support from JINR AYSS Grant. Computations were held on the basis of the heterogeneous computing cluster HybriLIT (LIT, JINR).

## References

[1] R.D. Peng, Science **334**, 1226 (2011)

[2] *Accelerator physics codes*, Wikipedia, `https://en.wikipedia.org/wiki/Accelerator_physics_codes`

[3] *Electromagnetic particle-in-cell computational applications*, Wikipedia, `https://en.wikipedia.org/wiki/Particle-in-cell#Electromagnetic_particle-in-cell_computational_applications`

[4] A. Friedman, R.H. Cohen, D.P. Grote, S.M. Lund, W.M. Sharp, J.L. Vay, I. Haber, R.A. Kishek, IEEE Transactions on Plasma Science **42**, 1321 (2014)

[5] M. Bussmann, H. Burau, T.E. Cowan, A. Debus, A. Huebl, G. Juckeland, T. Kluge, W.E. Nagel, R. Pausch, F. Schmitt et al., *Radiative Signatures of the Relativistic Kelvin-Helmholtz Instability*, in *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis* (ACM, New York, NY, USA, 2013), pp. 5:1–5:12

[6] J. Derouillat, A. Beck, F. Pérez, T. Vinci, M. Chiaramello, A. Grassi, M. Flé, G. Bouchard, I. Plotnikov, N. Aunai et al., Computer Physics Communications (2017)

[7] V.L. Smirnov, S.B. Vorozhtsov, *SNOP-Beam dynamics analysis code for compact cyclotrons*, in *Proceedings of the XXI Russian Accelerator Conference* (2012), pp. 325–327

[8] *Ef: Software for nonrelativistic beam simulation by particle-in-cell algorithm*, `https://github.com/epicf/ef/wiki`

[9] R.W. Hockney, J.W. Eastwood, *Computer Simulation Using Particles* (CRC Press, 1988)

[10] Y.N. Grigoryev, V.A. Vshivkov, M.P. Fedoruk, *Numerical "Particle-in-Cell" Methods: Theory and Applications* (VSP, 2002)

[11] J. Boris, *Relativistic Plasma Simulation—Optimization of a Hybrid Code*, in *Proceedings of the Fourth Conference on Numerical Simulation of Plasmas* (Naval Research Laboratory, Washington DC, 1970), pp. 3–67

[12] H. Qin, S. Zhang, J. Xiao, J. Liu, Y. Sun, W.M. Tang, Physics of Plasmas **20**, 084503 (2013)

[13] The HDF Group, *Hierarchical Data Format, version 5* (1997-2017), http://www.hdfgroup.org/HDF5/

[14] I.V. Alyamovsky, *Electron Beams and Electron Guns* (Soviet Radio, 1966)

[15] E.D. Donets, D.E. Donets, E.E. Donets, V.V. Salnikov, V.B. Shutov, Y.A. Tumanova, V.P. Vadeev, M. Bjorkhage, A. Lundquist, L. Liljeby et al., Review of Scientific Instruments **73**, 679 (2002)