https://doi.org/10.46813/2020-130-059

PiCOPIC: 2.5-D PARTICLE-IN-CELL CODE, OPTIMIZED FOR SIMULATION OF BEAM-PLASMA INTERACTIONS

O.K. Vynnyk, I.O. Anisimov

Taras Shevchenko National University of Kyiv, Kyiv, Ukraine

E-mail: alexander.vynnyk@protonmail.com

Original code optimized forsimulation of interactions between plasma and charged particles beams and bunches, based on particle in cell method described. The code is electromagnetic and fully relativistic with 2.5D axial symmetric geometry. Binary coulomb particle collisions are taken into account. The code is fully parallelized and designed for computer systems with shared memory. Ability to extend supported platforms to systems with distributed memory (like computer clusters or grids) is embedded into code architecture.

PACS: 29.17.+w;41.75.Lx

INTRODUCTION

Computer simulations play important role in modern studies of plasmas. Simulations allow to clarify many aspects of real plasma experiments, and set directions and scripts for their development. Kinetic simulations of plasmas need maximal sources [1]. On the other hand, they are the most accurate and allow to observe a lot of physical effects. Particle in cell is most widely used method of kinetic plasma simulation. Initially developed for studies of hydrodynamic calculations [2], it was adopted for plasma physics and became de-facto standard simulation method for wide range of plasma-related studies from astrophysics to controlled fusion.

One of the perspective areas of plasma physics are beam-plasma and laser-plasma interactions. Beam-plasma interactions become an object for fundamental studies as well for applied purposes. Electron beams and bunches can be used as electromagnetic waves' emitters, for excitation of wake waves and further electrons acceleration, for beam-plasma discharge ignition as a possible source of dense plasmas etc.

PiCOPIC is the successor of PDP3 [3] code, that was developed for the same purposes. This code has a lot of architecture limitations. Also, PDP3 was designed as single-thread code and can not be fully parallelized. Development of modern, fast and flexible PIC code, optimized for simulation of beam-plasma and laser-plasma interactions with perspective to extend it to cluster-based calculations was the main motivation to create PiCOPIC.

1. GENERAL PIC SCHEME

Particle in cell is fully kinetic plasma simulation approach. The mathematical model, underlying this method is the Vlasov-Maxwell system of equations [4]. PIC method performs a discretization of these equations. It replaces particles distribution function to the sum of parameters of so-called “macroparticles”. Each macroparticle represents the set of regular particles of some specie Fig. 1. Another discretization replaces continuous field’sdistribution with fixed spatial grid. Maxwell equations is solved only at the nodes of such grid. Time is also splitting into discrete steps.

Integration of the macroparticles motion equations and solving Maxwell equations is solved at every time step.

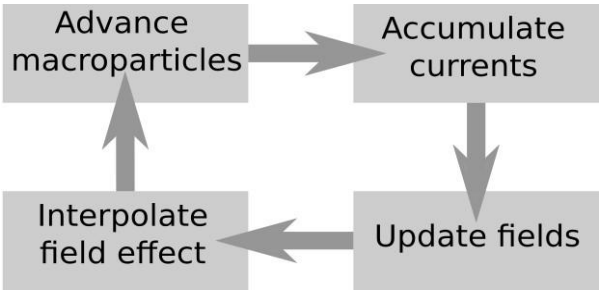

Fig. 1. General scheme of PIC method for single time step

2. CODE DESCRIPTION AND FEATURES

PiCOPIC was designed as lightweight high-performance application with simple build management procedure. So, it contains minimal amount of external dependencies and tools. One of the main objectives of this code is to make it simple for reading and updating. Accepting such requirements modular C++ approach was applied. Anyway, parts of the code, that required high performance, were implemented as low-level C/C++ code without classes. Summarizing, all performance sensitive places were implemented, with fast low-level code. Wrappers over such blocks were implemented with C++ classes and methods with low amount of system calls, designed to be understandable and extensible. Data analysis tools were written on python, using scientific libraries and jupyter notebooks.

2.1. GEOMETRY

Geometry of the simulation area is inherited from PDP3 package. It is axisymmetric with bunches' injection along the cylinder axis and the shape of macroparticles is a cylindrical ring (Fig. 2). This approach allows taking into account all three velocity dimensions, but, only 2 spatial dimensions.

Disadvantage of 2d3v simulation approach is geometrical singularity on the central axis, that causes a lot of peculiarities in calculations of particle advance, weighting etc. Another disadvantage is a unique weight of each macroparticle. This feature requires to be taken into account during binary collisions, temperature and density calculations etc.

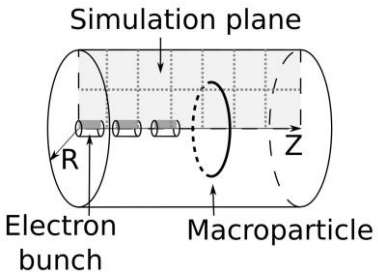

Fig. 2. Geometry

2.2. CORE ARCHITECTURE

In opposite to its predecessor, PiCOPIC code designed to store data and logic completely separate. Each specie of data (e.g. current and field grids, macroparticle parameters) stored in separate data structure, that processing by OOP methods of corresponding class, designed to reduce system calls. Such design allows to simply data manipulation, addressing it by pointers.

Each major part of the PIC loop could be implemented with several alternate algorithms, that can be altered at configuration phase. All these alternatives are implemented as child classes of common parent abstract class, or separate methods of the same class.

Required algorithm can be altered by setting compiler directives. Building alters the algorithm, following these directives. General scheme of the architecture is described on Fig. 3.

PiCOPIC supports several alternative algorithms for macroparticles advance: Boris [5], Vay [6] or Higuera-Cary [7] algorithms. Charge conservation is implemented with Villasenor-Buneman [8] and ZigZag [9] algorithms. Maxwellian solver implemented only with Yee [10], but technically multiple alternatives could be added to existing one.

Simulation model data stored as the set of member variables of configuration class. Object of this class initialized once, during application start. Initial configuration of the simulation model and conditions set uses this object.

Configuration of the package can be done in configuration phase, before build, where specific algorithms and other internal parameters can be set, and via configuration file, where simulation model parameters can be set.

Output data can be written in two alternate formats: plain text and HDF5 [11]. HDF5 is default format and recommended for usage. Plain text is supported as simple format for debug and development purposes. Output is configuring, using objects, named “probes”. Each probe describes the subarea of the simulation area with one of grid parameters (e.g. current, electric field, temperature), which should be written as output data.

Output data analysis implemented as several python libraries, wrapped around python scientific libraries and libraries for HDF5. They used in various jupyter note books [12], that can be simply customized for current purposes.

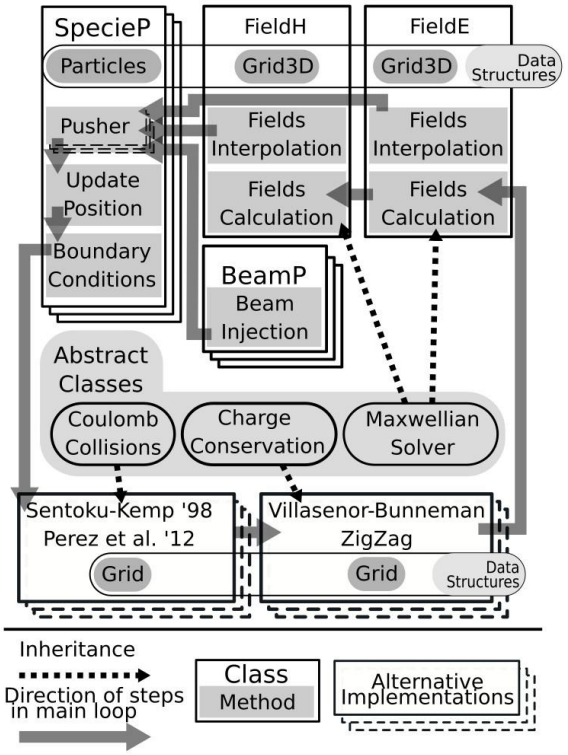

Fig. 3. Temperature weighting scheme

2.3. TEMPERATURE CALCULATION

Temperature is a macroparameter of gas or plasma and has sense only when the condition $n_k / N << 1$ is satisfied, where n_k is amount of particles with velocities $v - v + dv$ and N is the total amount of particles. This condition is possible only for large amount of N. In the most cases of PIC simulation this condition is not satisfied for single grid cell, where mean amount of macroparticles is frequently less then 10^2. In this case temperature could be interpreted only as mean energy.

Calculation of mean energy map is implemented with two alternative algorithms: direct mean energy calculation and mean energy weighting. Filtering of direct velocity of macroparticles also implemented for both algorithms.

Finding mean energy per PIC grid cell is quite simple procedure, that can be described by expression:

$$\begin{aligned} P_{i,j} &= \sum_{k}^{N_{i,j}} W_k m_s v_k, \\ n_{i,j} &= \sum_{k}^{N_{i,j}} W_k, \\ E_{i,j} &= \frac{P_{i,j}^2}{2mn_{i,j}^2}, \end{aligned} \qquad (1)$$

where $P_{i,j}$ is a total momentum of particles in [i, j] cell; $n_{i,j}$ is a number of particles and $N_{i,,j}$ is a number of

macroparticles in [i, j] cell; m_s is a mass of particle specie; W_k is a weight of k's macroparticle; v_k is a velocity of k's macroparticle.

Directed velocity filtering is also simple. The main idea is to subtract the directed velocity of particles from its total velocity. The expression for this procedure is:

$$\begin{aligned} \vec{P}_{i,j} &= \sum_k^{N_{i,j}} W_k m_s \vec{v}_k, \\ T_{i,j} &= \frac{P_{i,j}^2 - \vec{P}_{i,j}^2}{2mn_{i,j}^2}, \end{aligned} \tag{2}$$

where $T_{i,j}$ is a mean energy of chaotic motion in [i, j] cell. Calculation becomes invalid for high gradients of the directed velocity.

Main idea of weighting-based temperature calculation is to weight particles to the cell nodes, instead of its direct counting. Weighting procedure depends on the form factor of macroparticle. In the case of PiCOPIC, this procedure is described on Fig. 4 and expressed as

$$\begin{aligned} \xi_{k,[i,j]} &= \frac{V_{k,prtl,[i,j]}}{V_{cell,[i,j]}}, \\ \Xi_{i,j} &= \sum_k^{N_{i,j}} \xi_{k,[i,j]} W_k, \end{aligned} \tag{3}$$

$V_{k,pltl,[i,j]}$ is a part of k's particle volume, weighted to node [i, j]; $V_{cell,[i,j]}$ is a volume of cell-like structure with the center at the position of node [i, j]; $\Xi_{i,j}$ is a total particle weight for [i, j] node. Say, physically $\Xi_{i,j}$ is a value, proportional to weighted density of particles (e.g. electrons or ions).

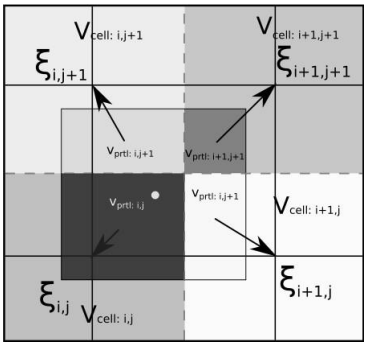

Fig. 4. Temperature weighting scheme

Further procedure is similar to eqs. (2), (3), except momentums that are also weighting to grid nodes:

$$\begin{aligned} P_{i,j} &= \sum_k^{N_{i,j}} \xi_{k,[i,j]} W_k \gamma_k m_s v_k, \\ \vec{P}_{i,j} &= \sum_k^{N_{i,j}} \xi_{k,[i,j]} W_k \gamma_k m_s \vec{v}_k, \\ T_{i,j} &= \frac{P_{i,j}^2 - \vec{P}_{i,j}^2}{2m\Xi_{i,j}^2}. \end{aligned} \tag{4}$$

There are options to apply bilinear and bicubic interpolation to existing temperature maps. Testing and comparison of different algorithms are shown on Fig. 5.

PiCOPIC's code also takes into account relativistic case of such calculation.

2.4. BOUNDARY CONDITIONS

Boundary conditions for particles were implemented as reflection for particles of background plasma and absorption for beam particles. Perfectly matched layer [13] can be applied to the boundaries of field grids. Such approach can effectively absorb plasma waves and oscillations, preventing reflection of such waves from the outer boundaries of the simulation area.

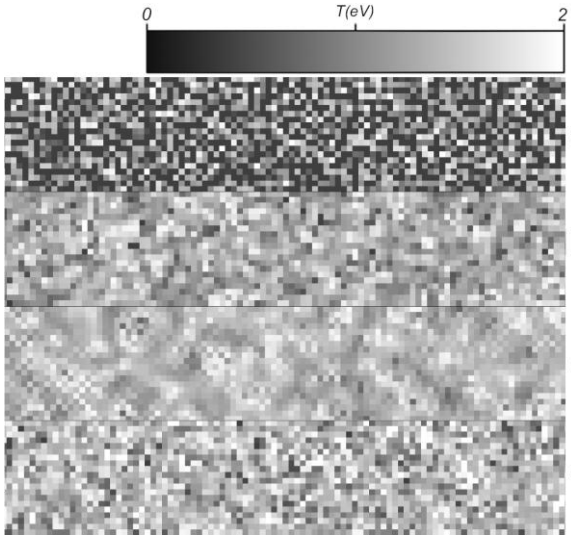

Fig. 5. Comparison of temperature maps, generated with different algorithms: a – direct counting; b – direct counting with bilinear interpolation; c – direct counting with bicubic interpolation; d – weighting

2.5. BINARY COULOMB COLLISIONS

Various processes in plasmas are related to plasma heating. There are set of conditions, where taking into account of coulomb binary collisions between charged particles is required. PiCOPIC uses modern direct simulation Monte-Carlo algorithm for coulomb collision calculation [14] as main and its predecessor [15] as an alternative algorithm.

2.6. PARALLELIZATION

Package was designed to be fully parallel and run on systems with shared memory as well as distributed one. It's hard to parallelize PIC loop. Each of its steps depends on each other. It requires frequent synchronization of parallel threads. And some of the loop steps cannot be parallelized et al. (e.g. charge conservation).

Traditional way to parallelize such code is to split simulation area to domains (see Fig. 2). Each domain has a real boundary or overlay area on all its edges (depends on its relative location). In the case of real boundary macroparticle reached the domain's edge, interacts with it, according to the boundary condition

logic. When particle reaches overlay area, it moves to the domain, next to the current one.

Grid values weight to overlay areas in the common way, but after each weighting overlays should be synchronized as shown on Fig. 6. This procedure was implemented as the separate PIC loop step. Synchronization procedure is also paralleled in the staggered order.

Each macroparticle is a low-level data structure, placed in the free memory. It is linked to the particle specie object with C++ pointer. Such design allows to move the particle from one domain to another in very performance efficient way: just by its repointing. Disadvantage of the pointer change method is the requirement to work in the shared memory.

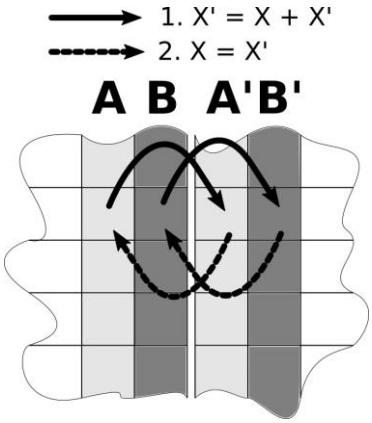

Fig. 6. Overlaying

2.7. TEST SIMULATIONS

To review performance and correct working of the package, several test simulations with parameters close to previous beam-plasma interaction simulations [16, 17] were performed. The similar result was obtained, but with higher accuracy, as the simulations were setup with higher amount of macroparticles and larger number of grid nodes.

Electrons velocity distribution in the subarea far from PML regions was calculated to check, if thermal velocity distribution works correctly. Obtained distribution was close to maxwellian (Fig. 7).

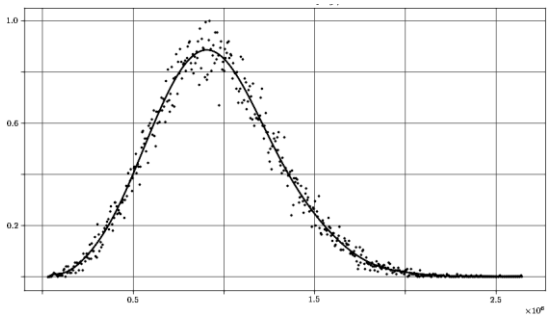

Fig. 7. Normalized electrons velocity distribution for the subarea, that is far from PML regions

Dependency of electric field from the time was plotted and compared to the same plot, for simulation with the same parameters, processed on PDP3. This test was directed to check if the package works correctly at the large simulation periods (10^4...10^5 steps), both simulations (using PDP3 and PiCOPIC) were performed for case of resonant multibunch electron beam injection to warm plasma. Fig. 8 shows, that E_z component changing is quite similar to PDP3's result. The differences can be explained by higher accuracy of simulation with PiCOPIC.

Evolution of the electric field and temperature is quite similar to [17]. Fig. 9 shows E_r images for the region, that is close to the beam injection point for different moments of time. One can see the field perturbations, similar to ones described in [17].

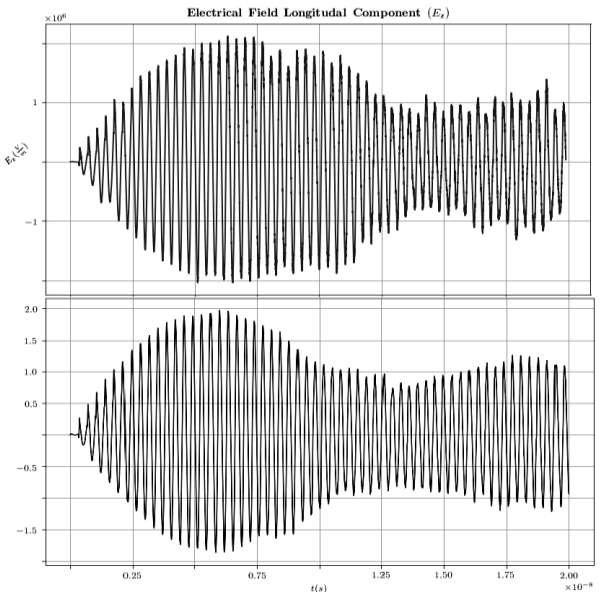

Fig. 8. Dependency Ez from time for the same cases, simulated with PDP3 (top) and PiCOPIC

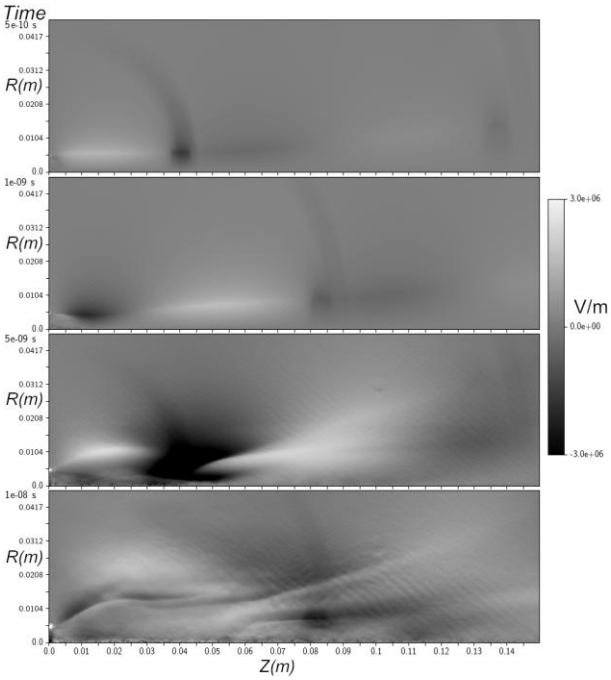

Fig. 9. E_r spatial map for the different time moments. Field perturbations are on late time moments (bottom)

CONCLUSIONS

The code for kinetic plasma simulation, based on particle in cell method described. The code is written on

C++ programming language. Performance-critical algorithms covered by low-level C-style code. The code optimized for simulations of beam-plasma interactions and uses 2.5D axial symmetric geometry. Binary coulomb collisions implemented with modern DSMC algorithm. Data outputs to binary format HDF5, designed for scientific purposes. It provides fast access to various data subsets, including cases of large size of the data and allows working with large data sets efficiently. The code is designed as fast, fully parallel with working on computer systems with shared memory and architectural ability to execute it on computational clusters with relatively minor code update.

REFERENCES

1. C.K. Birdsall, A.B. Langdon. *Plasma physics via computer simulation.* Taylor and Francis, New York. 2005.
2. M.W. Evans, F.H. Harlow. *The particle-in-cell method for hydrodynamic calculations.* 1957.
3. Y.M. Tolochkevych, T.E. Litoshenko, I.O. Anisimov. 2.5d relativistic electromagnetic PIC code for simulation of the beam interaction with plasma in axial-symmetric geometry *// Journal of Physics: Conference.* 2014, v. 511, p. 012001.
4. A. Vlasov. The vibrational properties of an electron gas // *Soviet Physics Uspekhi*. 1968, v. 10(6), p. 721-733.
5. J.P. Boris. *Relativistic plasma simulation optimization of a hybrid code*. 1970.
6. J.-L. Vay. Simulation of beams or plasmas crossing at relativistic velocity // *Physics of Plasmas.* 2008, v. 15(5), p. 056701.
7. V. Higuera, J.R. Cary. Structure-preserving second-order integration of relativistic charged particle trajectories in electromagnetic fields // *Physics of Plasmas.* 2017, v. 24(5), p. 052104.
8. J. Villasenor, O. Buneman. Rigorous charge conservation for local electromagnetic field solvers // *Computer Physics Communications*. 1992, v. 69(2), p. 306-316.
9. T. Umeda, Y. Omura, et al. A new charge conservation method in electromagnetic particle-in-cell simulations // *Computer Physics Communications.* 2003, v. 156(1), p. 73-85.
10. K. Yee. Numerical solution of initial boundary value problems involving maxwell's equations in isotropic media // *IEEE Transactions on Antennas and Propagation*. 1966, v. 14(3), p. 302-307.
11. M. Folk, G. Heber, et al. An overview of the hdf5 technology suite and its applications. in Proceedings of the EDBT/ICDT 2011 // *Workshop on Array Databases*. AD '11. Association for Computing Machinery, New York, NY, USA. 2011, p. 36-47.
12. J. Perkel. Why jupyter is data scientists' computational notebook of choice // *Nature.* 2018, v. 563, p. 145-146.
13. J.-P. Berenger. A perfectly matched layer for the absorption of electromagnetic waves // *Journal of Computational Physics.* 1994, v. 114(2), p. 185-200.
14. F. Perez, L. Gremillet, et al. Improved modeling of relativistic collisions and collisional ionization in particle-in-cell codes // *Physics of Plasmas*. 2012, v. 19, p. 083104.
15. Y. Sentoku, A. Kemp. Numerical methods for particle simulations at extreme densities and temperatures: Weighted particles, relativistic collisions and reduced currents // *Journal of Computational Physics.* 2008, v. 227, p. 6846-6861.
16. O.K. Vynnyk, I.O. Anisimov. Wake wave excited by the sequence of relativistic electron bunches: Initial stage // *Problems of Atomic Science and Technology*. *Series «Plasma Physics» (118).* 2018, № 6, p. 160-163.
17. O.K. Vynnyk, I.O. Anisimov. Evolution of the wake wave excided by the sequence of the relativistic electron bunches // *Problems of Atomic Science and Technology*. *Series «Plasma Physics» (122).* 2019, № 4, p. 55-58.

Article receievd 15.10.2020

PiCOPIC: 2.5-МЕРНЫЙ КОД, ОПТИМИЗИРОВАННЫЙ ДЛЯ МОДЕЛИРОВАНИЯ ПЛАЗМЕННО-ПУЧКОВОГО ВЗАИМОДЕЙСТВИЯ

А.К. Винник, И.А. Анисимов

Описан оригинальный код для моделирования взаимодействия электронных сгустков и пучков с плазмой, основанный на методе крупных частиц. Код электромагнитный, полностью релятивистский, выполнен в 2,5-мерной аксиально-симметрической геометрии. Учитываются парные кулоновские столкновения заряженных частиц. Дизайн кода полностью параллельный для запуска на компьютерных системах с общей памятью. В архитектуру заложена возможность расширения поддерживаемых платформ на системы с распределенной памятью (вычислительные кластеры или гриды).

PiCOPIC: 2,5-ВИМІРНИЙ КОД, ОПТИМІЗОВАНИЙ ДЛЯ МОДЕЛЮВАННЯ ПЛАЗМОВО-ПУЧКОВОЇ ВЗАЄМОДІЇ

О.К. Винник, І.О. Анісімов

Описаний оригінальний код для моделювання взаємодії електронних згустків із плазмою, заснований на методі крупних частинок. Код електромагнітний, повністю релятивістський, виконаний у 2,5-вимірній, аксіально-симетричній геометрії. Враховуються парні кулонівські зіткнення заряджених частинок. Дизайн коду повністю паралельний для запуску на комп'ютерних системах зі спільною пам'яттю. В архітектуру коду закладена можливість розширення підтримуваних платформ на системи з розподіленою пам'яттю (обчислювальні кластери або гріди).