

## The parallel three-dimensional PIC code for the numerical modeling of ultrarelativistic beams\*

M.A. Boronina, V.D. Korneev

**Abstract.** We present a parallel 3D algorithm for simulation of beam-beam effects in super-colliders, where colliding beams have superhigh densities and high relativistic factors. The algorithm is based on particle and domain decomposition and demonstrates good speed-up and scalability.

### 1. Introduction

In this paper, we present a parallel particle-in-cell code for the numerical simulation of ultrarelativistic charged beams in supercolliders. This code is three-dimensional and takes into account high relativistic factor values and a significantly nonlinear charge density distribution.

Today the most elaborated method of the numerical simulation of beam-beam effects for relativistic charged particles is based on the slice model. Both colliding bunches are divided into macroparticle slices, and then the two-dimensional field of transverse forces is calculated for each slice. Usually the transversal field is calculated with the Basetti–Erskine equations [1] or as a derivative of 2D potential. The Poisson equation with boundary conditions or the Green function can be used to obtain the potential [2, 3]. The particles of the counter beam receive a kick of the 2D fields, changing their motion. Standard ways of parallelizing are based on the slice decomposition, when one or a few slices are assigned to one processor. Another way is a two-dimensional domain-decomposition approach, when each processor contains one rectangular block domain. But this method is not effective when particles move far from their positions during the time step. This problem can be avoided by using a particle-field decomposition [4].

This quasi 3D approach is used for studying the problem of “strong-strong” interaction. However, the reduction to the 2D problem, where the longitudinal motion is simulated by “re-arrangement” of slices, cannot completely cover the longitudinal effects, which are of particular importance for colliding beams with superhigh densities. In addition, the model complicates the simulation of the beams crossing at a big angle. In the 3D algorithm proposed, we employ the Vlasov–Liouville equation for the distribution function of beam particles, a three-dimensional set of Maxwell’s equations and the

---

\*Supported by RFBR under Grant 11-01-00178-a “Supercomputer simulation of ultrarelativistic charged particle beam dynamics

new methods for initial and boundary conditions calculations [5], which automatically account for such difficulties. We solve these equations by using the particle-in-cell (PIC) method and the leap-frog scheme [6].

The parallel code is based on the domain decomposition along the transversal direction: every processor group receives its own part of the subdomain grid and all the particles of the subdomain. We additionally employ the particle parallelization: within the group every processor receives its own set of particles [7]. This method allows one to appreciably increase the scalability and to overcome high restrictions on the number of particles due to a high nonlinearity of the density distribution and the limited processor memory: for 6 processors and  $100 \times 100 \times 100$  grid the limit is  $2 \cdot 10^6$  particles, which means quite a small number of particles in a cell for the PIC method. With advances of the code and with the advent of its parallel supercomputer version it will be possible to apply it to the beam-beam simulations for supercritical parameters.

## 2. Computational methods

Let us consider the motion of the counter charged electron/positron beams in the rectangular domain  $[0, L_x] \times [0, L_y] \times [0, L_z]$ . The motion takes place in vacuum in self-consistent electromagnetic fields with allowance for the external focusing field. Each beam is defined by its shape, size, coordinates in space and time, the number of particles, nonlinear density distribution. We need to analyze the particle motion dependence on the given beam configuration.

The problem can be described by the Vlasov kinetic equation for the distribution function of electrons  $f_- = f_-(\mathbf{r}, \mathbf{p}, t)$  and positrons  $f_+ = f_+(\mathbf{r}, \mathbf{p}, t)$

$$\frac{\partial f_{\pm}}{\partial t} + \mathbf{v}_{\pm} \frac{\partial f_{\pm}}{\partial \mathbf{r}} + \mathbf{F}_{\pm} \frac{\partial f_{\pm}}{\partial \mathbf{p}} = 0. \quad (1)$$

The Lorentz force can be calculated from the following equation:

$$\mathbf{F}_{\pm} = e^{\pm}(\mathbf{E} + \mathbf{v}_{\pm} \times \mathbf{H}/c) \quad (2)$$

and the particle impulse  $\mathbf{p} = \gamma_{\pm} m_e \mathbf{v}_{\pm}$ , where the relativistic factor  $\gamma_{\pm} = (1 - |\mathbf{v}_{\pm}|^2/c^2)^{-1/2}$ .

Maxwells equation system connects the charge densities  $n_+$ ,  $n_-$ , the current  $\mathbf{j}$ , the electric and magnetic fields  $\mathbf{E}$  and  $\mathbf{H}$ :

$$\begin{aligned} \text{rot } \mathbf{E} &= -\frac{1}{c} \frac{\partial \mathbf{H}}{\partial t}, & \text{div } \mathbf{E} &= 4\pi(n_- e^- + n_+ e^+), \\ \text{rot } \mathbf{H} &= \frac{4\pi}{c} \mathbf{j} + \frac{1}{c} \frac{\partial \mathbf{E}}{\partial t}, & \text{div } \mathbf{H} &= 0. \end{aligned}$$

The charge density represents the distribution function moment:

$$n_+ = \int_V f_+ dV, \quad n_- = \int_V f_- dV, \quad \mathbf{j} = \int_V (f_+ \mathbf{v}e - f_- \mathbf{v}e) dV.$$

The Vlasov characteristic equations coincide with the equations of the particle motion:  $\frac{\partial \mathbf{p}_\pm}{\partial t} = \mathbf{F}_\pm$ ,  $\frac{\partial \mathbf{r}_\pm}{\partial t} = \mathbf{v}_\pm$ .

All the equations are written in dimensionless variables, the characteristic length  $L$  of the beam is 1 cm and the characteristic speed  $v$  of the particles is the speed of light. We use the particle-in-cell method with the PIC form-factor and the leap-frog scheme [8, 9]. All the components are calculated at the half-step time and space grids. In this case, all the derivatives involved in the equations are written with central differences, and this scheme provides second order with respect to time and space. For example, we use the following scheme for Maxwell's equations:

$$\frac{\mathbf{H}^{m+1/2} - \mathbf{H}^{m-1/2}}{\tau} = -\text{rot}_h \mathbf{E}^m, \quad \frac{\mathbf{E}^{m+1} - \mathbf{E}^m}{\tau} = \mathbf{j}^{m+1/2} + \text{rot}_h \mathbf{H}^{m+1/2}.$$

The second-order scheme for the impulse is the following:

$$\frac{\mathbf{p}^{m+1/2} - \mathbf{p}^{m-1/2}}{\tau} = q \left( \mathbf{E}^m + \left[ \frac{\mathbf{v}^{m+1/2} + \mathbf{v}^{m-1/2}}{2}, \mathbf{H}^m \right] \right).$$

The stability condition of the method is  $v\tau/h < 1$ , where  $h = \min\{h_x, h_y, h_z\}$ , and exactly this condition defines the times step (not the accuracy condition).

We use the Villancenor–Buneman scheme [10] in order to calculate the currents. This method exactly satisfies the Gauss law in finite differences, thus significantly reducing the approximation error and making the algorithm more robust.

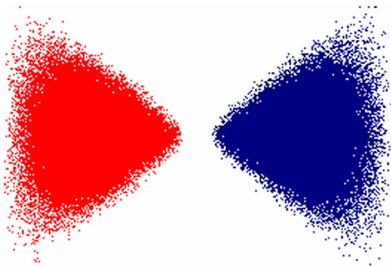
The initial conditions and boundary conditions are required for the scheme application. A new method for the boundary and initial condition calculations is based on the assumption that the model particles have the shape of needles with length  $h_z$  directed along the axis  $z$ :

$$\begin{aligned} E_x(x, y, z) &= \frac{2q(x - x_0)}{h_z((x - x_0)^2 + (y - y_0)^2)}, \\ E_y(x, y, z) &= \frac{2q(y - y_0)}{h_z((x - x_0)^2 + (y - y_0)^2)}, \\ E_z(x, y, z) &= 0, \quad \mathbf{H} = [\mathbf{v}, \mathbf{E}]. \end{aligned}$$

The advantage of the model proposed is three-dimensional Maxwell's equations, which allow calculating the beams movement regardless of the

collective motion direction. The main problem of the three-dimensional modeling is the presence of the relativistic factor, which is an unavoidable part of the problem. This leads to the high gradients and, thus, to the necessity of creation and application of the corresponding parallel algorithm [5]. Furthermore, the beam density distribution is highly nonlinear (Gaussian in each direction), and we have to use a high resolution spatial grid.

Another problem is associated with the hour-glass effect: the beam changes its shape significantly due to the focusing conditions, and the overwhelming majority of the particles are concentrated in a small intersection, while the size of the domain is very big.



**Figure 1.** Beam profiles in  $z$ - $x$  plane

In addition, the essence of the PIC method is in using an appropriate particle number in cell. The convergence condition leads to decreasing the time-step with decreasing the spatial step, and thus — to decreasing the number of particles. All these facts necessitate creating a well-balanced highly scalable 3D parallel code.

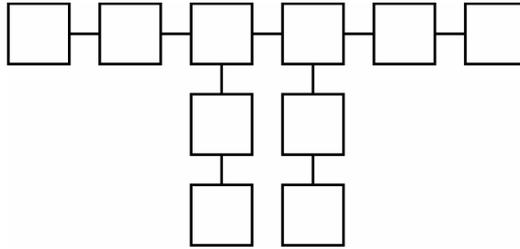
### 3. The parallel algorithm description

The parallel algorithm in question is based on the following domain decomposition. The spatial domain is uniformly divided into stripes along the axis  $y$ . The collective particle motion takes place in the longitudinal direction, thus the quantity of interprocessor communications will be smaller in comparison with the case of decomposition along the axis  $z$ . Each subdomain is assigned to a group of processors  $n = 1, \dots, \text{isize0}$ , each processor in the group ( $k = k_1, \dots, k_n$ ) having the same spatial grid data, and the particles corresponding to the physical subdomain are divided evenly among all the processors of the group. Such a way of parallelizing yields the even particle distribution within the group due to using many processors in the high-density regions of the domain.

In order to calculate the initial and boundary electric fields, each processor computes its own density 3D array and sends it to the corresponding main processor of the group. The main group processor  $n$  calculates its own field 3D arrays and sends it to all the main processors  $n = 1, \dots, N$ , after the procedure is completed, it broadcasts the obtained field data within the group. The main group processors use Maxwell's equations to obtain new fields and broadcast them within the group. When the particle leaves the corresponding subdomain, the algorithm sends its parameters to one of the neighboring processors.



**Figure 2.** Linear decomposition structure: `isize = isize0 = 10`



**Figure 3.** Decomposition structure: `isize = 10, isize0 = 6`

It is necessary to calculate the initial particle distribution twice. First, the master processor computes the number of particles in each of `isize0` groups, determines the most optimal processor distribution of given `isize` processors among these groups and the maximum number of particles in every processor. Then the master creates the initial particle parameters and sends the packs of parameters to each of `isize` processors until all the particles are distributed.

The algorithm described has been tested on some characteristic examples of a relativistic beam motion simulation. For example, the beam consists of the electrons with the same energy moving strictly along the axis  $z$  at  $\gamma = 10^4$ . The particle density is distributed according to the Gaussian law with the focusing condition with focusing dimensionless parameters  $\varepsilon_x = \varepsilon_y = 5 \cdot 10^{-7}$ ,  $\beta_x = \beta_y = 0.1$ ,  $\sigma_z = 0.1$ , where  $\varepsilon_x$  and  $\varepsilon_y$  are the radial and vertical beam emittances, respectively,  $\beta_x$ ,  $\beta_y$  are the corresponding beta-function values,  $\sigma_z$  is the size of beam along the axis  $z$ . The relativistic factor  $\gamma = 6.85 \cdot 10^3$ , the charge  $Q = 2.63 \cdot 10^8 e$ . The size of the computational domain in dimensionless units is  $L_x = L_y = 10^{-2}$ ,  $L_z = 1$ . The spatial grid is taken as  $120 \times 120 \times 120$ , the number of particles is  $10^6$ , the time step is  $10^{-5}$ , the number of time steps is  $3 \cdot 10^4$ .

All the numerical experiments are performed on the Siberian Supercomputer Center cluster (ICMMG SB RAS), 576 4-core processors Intel Xeon 5450/E5540/X5670.

In the table, the time calculation and the maximum particle number is shown. The `isize0` column is number of groups, `isize` is the number of processors, the fourth column corresponds to the maximum particle number in processor  $t = 0$  in the beginning, the fifth column is maximum particle number in processor through whole the computational steps.

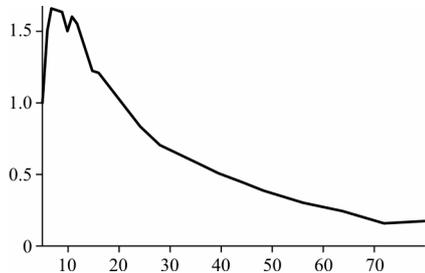
The table demonstrates the advantage of additional processors in groups in high-density regions. We can see that for `isize = isize0 = 12` the calcu-

isize0	isize	500 $\tau$ , s	$j_{\max}$ , $10^6$ , $t = 0$	$j_{\max}$ , $10^6$
6	6	98	475025	500162
6	10	39	159123	167167
10	10	93	396014	497950
10	12	49	198030	248795
12	12	84	120590	494225
6	12	37	119283	125476

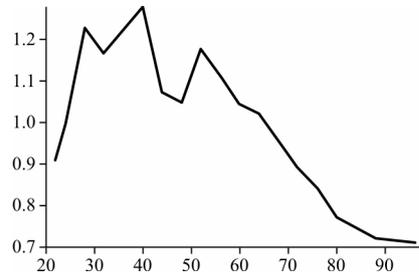
lation time is 84 s against 98 s of  $\text{isize} = \text{isize0} = 6$ , but the time needed is much less when there are 12 processors in 6 groups. We corroborate the idea of adding processors in high-density regions—it is much more effective to add 4–6 processors to the groups in the middle of calculation domain (37–39 s) rather than to distribute them along the line (93 s). This effect is connected with a significant particle number decreasing (from  $4 \cdot 10^5$  to  $1\text{--}1.5 \cdot 10^5$ ) at the initial time, and yields the number of particles decreasing from almost half all the particles down to  $1.2\text{--}1.8 \cdot 10^5$ . Although we can see that there is no big difference between 10 and 12 processors in 6 groups because of interprocessor communications within the group.

Figure 4 demonstrates the efficiency dependence on the processor number  $N$  ( $\text{isize}$ ) for  $\text{isize} = 5$ , grid  $60 \times 60 \times 60$ , the number of particles  $10^6$ , the calculation time of 1000 time steps being 300 s. The efficiency formula is  $EN_0T_N/T_{N_0}$ , where  $N$  is the number of processors,  $N_0$  is the initial number of processors. In the best way it could be monoprocessor program and  $N_0 = 1$ , but there is not sufficient memory resources to perform the numerical experiment. Thus, the efficiency is equal to 1.0 when we use  $N = \text{isize0} = 5$ .  $T_N$  and  $T_{N_0}$  are the run times on the corresponding the number of processors.

We can see from the picture that adding new processors to the high-density domain significantly increases the efficiency while the processor number is not quite big ( $\sim 20$  in 5 groups), and then interprocessor communications with spatial grids within each group takes more and more time, because



**Figure 4.** Parallelizing efficiency:  $\text{isize} = 5$ ,  $J = 10^6$ , grid  $60 \times 60 \times 60$



**Figure 5.** Parallelizing efficiency:  $\text{isize} = 20$ ,  $J = 10^7$ , grid  $120 \times 120 \times 60$

the processor number in each group arises, hence the efficiency decreases.

We can obtain the same result from higher parameters, `isize` = 20, grid  $120 \times 120 \times 60$ , the particle number  $10^7$ , the calculation time of 1000 time steps is 67 s (Figure 5). The graph dips can be explained with asymmetric processor distribution in groups, and thus some processors remain idle. From the numerical experiments, we obtain that the most efficient `isize0/isize` ratio is when `isize0` is 1/7 of the grid size along the axis  $y$  and `isize` is about 3–5 `isize0`. In the case of a big grid, the efficiency decreases due to sending grid copies within every group, however it is the only way to perform calculations with such high number of particles and small spatial steps. As the reasonable minimum  $N_y$  in processor is 4 (2 nodes + 2 auxiliary nodes), the parameters are limited only by the array size  $\sim 4N_xN_z$  of the grid subdomain in any processor.

#### 4. Conclusions

The new three-dimensional parallel algorithm for the beam-beam simulation with rather high gamma-factors ( $\gamma \sim 10^3$  and higher) has been developed. The algorithm is based on the domain and particle decomposition and allows performing numerical experiments for  $500 \times 500 \times 500$  particle grids and the particle number  $10^9$ , such parameters cannot be taken using any standard domain decomposition. The algorithm has revealed efficiency and good scalability.

#### References

- [1] Bassetti M., Erskine G. Closed Expression for the Electric Field of a Two-Dimensional Gaussian Charge. — 1980. — (CERNISR-ISR-TH/80-06).
- [2] Anderson E.B., Banks T.I., Rogers J.T. ODYSSEUS: a dynamic strong-strong beam-beam simulation for storage rings // International Computational Accelerator Physics Conference. — 1998.
- [3] Qiang Ji, Furman M.A., Ryne R.D. A parallel particle-in-cell model for beam-beam interaction in high energy ring colliders // J. Computational Physics. — 2004. — Vol. 198. — P. 278–294.
- [4] Qiang Ji, Paret S. Poisson solvers for self-consistent multi-particle simulations // ICFA Mini-Workshop on Beam-Beam Effects in Hadron Colliders, March 18–22, 2013.
- [5] Vshivkov V.A., Boronina M.A. Three-dimensional simulation of ultrarelativistic charged beams dynamics: study of initial and boundary conditions // Matem. Mod. — 2012. — Vol. 24, No. 2. — P. 67–83.
- [6] Boronina M.A., Vshivkov V.A., Levichev E.B., Nikitin S.A., Snytnikov V.N. An algorithm for the three-dimensional modeling of ultrarelativistic beams // Numerical Methods and Programming. — 2007. — Vol. 8, No. 2. — P. 203–210.

- [7] Kireev S.E. Parallel implementation of particle-in-cell method for simulation of gravitational cosmodynamics problems // *Avtometriya*. — 2006. — No. 3. — P. 32–39.
- [8] Hockney R., Eastwood J. *Computer Simulation Using Particles*. — McGraw-Hill, Int., 1980.
- [9] Langdon A.B., Lasinski B.F. *Meth. Comput. Phys.* — 1976. — Vol. 6.
- [10] Villancenor J., Buneman O. Rigorous charge conservation for local electromagnetic field solvers // *Comp. Phys. Comm.* — 1992. — Vol. 69.