ISBN: 978-3-95450-168-7 doi:10.18429/JACoW-IPAC2015-MOPMN012

SPACE CODE FOR BEAM-PLASMA INTERACTION

K. Yu*, Stony Brook University, Stony Brook, NY 11794, USA
R. Samulyak#, Stony Brook University, Stony Brook, NY 11794 and BNL, Upton, NY 11973, USA

Abstract

A parallel particle-in-cell code SPACE has been developed for the simulation of electromagnetic fields, relativistic particle beams, and plasmas. The algorithms include atomic processes in the plasma, proper boundary conditions, an efficient method for highly-relativistic beams in non-relativistic plasma, support for simulations in relativistic moving frames, and special data transfer algorithm from the moving to the laboratory frame that collects particles and fields in the lab frame without time shift due to the Lorentz transform, enabling data analysis and visualization. Plasma chemistry algorithms implement atomic physics processes such as the generation and evolution of plasma, recombination of plasma, and electron attachment on dopants in dense neutral gas. Benchmarks and experimental validation tests are also discussed. The code has been used for the simulation of processes relevant to the eRHIC program at BNL and the high pressure RF cavity (HPRF) program at Fermilab.

INTRODUCTION

Direct numerical simulation of plasma in the presence of atomic processes such as generation and recombination of charged particles is a complex multiscale problem. Plasma number density may change by orders of magnitude during relevant time scales, creating difficulties in representing secondary plasma particles by macroparticles within the Particle-in-Cell (PIC) method. Another difficulty is in the presence of different time scale, as in the case of plasmas interacting with relativistic particle beams. Evolution of atomic physics processes may be orders of magnitude longer compared to passing times of short relativistic bunches.

Novel algorithms for the simulation of plasma undergoing atomic processes and relativistic particle beams have been developed and implemented in SPACE, a parallel electromagnetic PIC code. The code has been applied to a number of problems relevant to advanced particle cooling mechanisms.

SUMMARY OF SPACE CODE

PIC Method for Maxwell's Equations

The system of Maxwell's equations in the presence of moving charges is written as

$$\frac{\partial \boldsymbol{B}}{\partial t} = -\nabla \times \boldsymbol{E} \quad (1a)$$

$$\frac{\partial \boldsymbol{E}}{\partial t} = -\frac{1}{\epsilon_0 \mu_0}\nabla \times \boldsymbol{B} - \frac{1}{\epsilon_0}\mathbf{J} \quad (1b)$$

$$\nabla \cdot \boldsymbol{E} = \frac{\rho}{\epsilon_0} \quad (1c)$$

$$\nabla \cdot \boldsymbol{B} = 0 \quad (1d)$$

The Maxwell equations are discretized using the finite difference time domain (FDTD) [1] with second order accuracy in space and time. Electric charges are represented by discrete macroparticles coupled with electromagnetic fields by the action of Lorentz forces and electric currents.

When one solves Maxwell's equations analytically, solving Eq. (1a) and Eq. (1b) is sufficient: if Eq. (1c) and Eq. (1d) are satisfied at initial time, they remain invariants of motion at later time. Due to numerical truncation errors, this property may be violated. To resolve this problem, a special rigorous charge conservation method was developed within the PIC framework in [2]. This algorithm, implemented in the SPACE code, requires solving numerically the Poisson problem (Eq. (1c)) only once for obtaining consistent initial conditions. Then only the first two Maxwell's equations (Eq. (1a) and Eq. (1b)) and the Newton-Lorentz equation are solved numerically at each time step. A schematic of processes occurring at each time step is depicted in Fig. 1. The computational sequence is of fields and particle coordinates is shown in Fig. 2. We also implemented modifications of the Boris scheme [3] proposed in [4] for dealing with rapidly accelerating particles for which the relativistic factor is not constant.

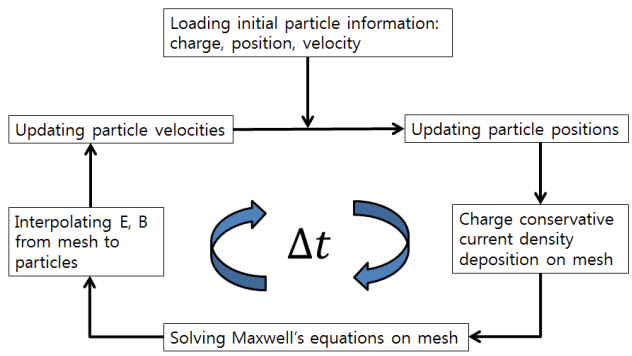

Figure 1: Schematic of one time step of SPACE code.

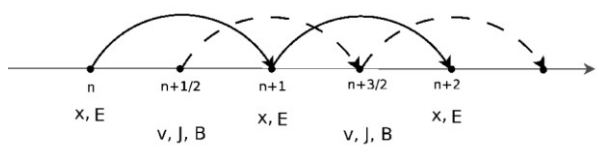

Figure 2: Computation sequence along time step

Code Structure and Properties

The code is developed in C++ utilizing the advantages of Objected-Oriented Programming. The code is

* kwangmin.yu@stonybrook.edu
# rosamu@bnl.gov

ISBN: 978-3-95450-168-7 doi:10.18429/JACoW-IPAC2015-MOPMN012

composed of three major parts. The first part, FieldSolver class, contains FDTD solvers. The second part, ParticleMover class, contains solvers for the Newton-Lorentz equation. This class also includes various physics models describing particle interactions and transformations by atomic physics processes. The code is capable of tracking numerous particle species. Finally, the third, TimeController class, controls the above classes and any miscellaneous classes such as classes performing the visualization of electromagnetic fields and particle data. The visualization is done using the visualization software called VisIt [5]. Since the main classes of the code are connected via the interface classes, the code can easily be extended by implementing additional functions and physics models. For convenience of a new problem setup, the initialization routines use XML (eXtensible Markup Language).

Parallelization Method

The electromagnetic PIC code is parallelized using a hybrid MPI / tread programming for distributed memory multicore supercomputers. FieldSolver uses domain decomposition for solving Maxwell's equations. ParticleMover uses a decomposition of particles that is independent of FieldSolver domain decomposition. Namely, particles in a parallel computing node can be distributed in the whole computational domain. As the distribution of particles is usually very non-uniform in the space, such decomposition maximizes the load balance.

The described parallel decomposition minimizes the CPU computing time but require a large amount of communications between FieldSolver and ParticleMover. We have adopted ideas from the sparse matrix storage to minimize the amount of communications and send field data to ParticleMover from only those computational cells that contain macroparticles.

Other Algorithms and Implementations

The code SPACE fully supports simulations in relativistic moving frames. But visualization of such simulations in the laboratory frame is difficult due to the Lorentz transformation property: particles collected at a fixed time in the moving frame obtain individual times after being transformed to the laboratory frame. We have designed and implemented a special data transformation algorithm from the moving frame to the lab frame that collects particles and fields data in the lab frame without time shift due to the Lorentz transformation.

Proper boundary conditions such as superconducting and periodic boundary conditions are implemented.

High plasma oscillation frequency in dense plasma imposes limitation on the time step size [6]. But plasma oscillations are often irrelevant to main physics processes of a problem if interest. Special filters are implemented to mitigate oscillations. In certain cases, we also use a larger value of the permittivity of free space (ϵ_0) in the plasma region if the main physics processes of interest are unaffected by this change. This allows us to speed up the code and extend simulations to long physical time.

ATOMIC PHYSICS ALGORITHMS

Generation of Plasma Macroparticles

Two algorithms for the dynamic generation of plasma are currently used in the code. Here we describe the first algorithm that dynamically creates plasma macroparticle pairs. The second algorithm that changes the representing number of macroparticles is described in the next subsection.

Consider an example of neutral gas ionization by a high energy particle beam. As each beam macroparticle passes through the gas, it loses energy and ionizes the medium in real time by creating electron - ion pairs. The process is described by the Bethe-Block equation [7]. Each pair of electron and ion macroparticles must be created in the same spatial location to satisfy the initial local neutrality. The mobility of ions is often very low throughout the simulation and the motion of ions can be ignored. In this case, we need to create only electron macromarticles. As described in the previous section, only equations (1a) and (1b) are solved by the FDTD scheme at each time and equations (1c) and (1d) are solved at the initial time. As each new macroparticle pair represents initially a neutral point, its addition has no effect on (1c). Plasma particles influence the system only through their motion, by changing the current in equation (1b). Therefore it is not necessary to create ion macroparticles in the assumption of fixed ions.

If electron-ion recombination is important [8], ion density can be computed on the FDTD mesh. When dopant is imposed in a simulation so that attachment of electrons to the dopant must be resolved, the negative dopant ion by electron attachment can be dealt with as another species of ion on the FDTD mesh.

Variable Representing Number

Plasma density can change by orders of magnitude via ionization and recombination processes. When a fixed representing number is used for plasma, the plasma density variation involves changing the number of macroparticles of plasma. It may cause poor accuracy in the low desnity regions represented by small number of macroparticles or waste of computing resources in high density regions containing large number of macroparticles.

This problem is effectively eliminated by using variable representing number of macroparticles. In this algorithm, a preset cloud of massless neutral plasma macroparticles (with zero representing number) is initialized, and such macroparticles are “charged” during ionization processes. The representing number of a plasma macroparticle increases when ionization occurs and it decreases when recombination occurs. This algorithm also eliminates the effect of artificial charge separation if spatially displaced ion and electron macroparticles are removed in the recombination process within the method described above.

ISBN: 978-3-95450-168-7 doi:10.18429/JACoW-IPAC2015-MOPMN012

Ionization and Recombination Algorithms

In the variable representing number algorithm, the total number of macroparticles remains constant while particle representing numbers increase or decrease according to ionization and recombination processes.

Figure 3 shows the schematic description of the ionization algorithm by a particle beam. By the movement of a beam particle (blue), its energy loss in gas is estimated and distributed to the FDTD mesh (green). After that, the number of new plasma pairs is computed on the mesh. The number of new ionization events in each mesh block is distributed to the plasma macroparticles and used for to change of their representing number. The recombination has reverse process.

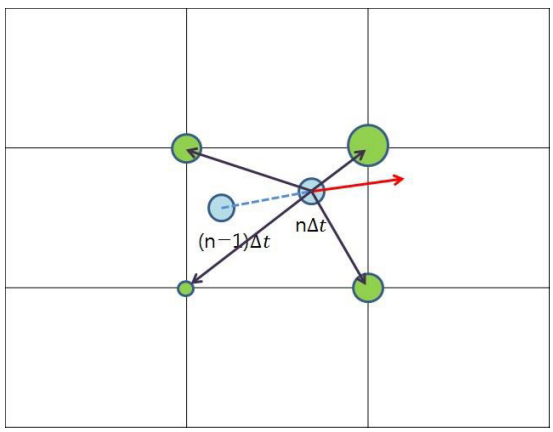

Figure 3: Schematic diagram of the ionization algorithm.

APPLICATIONS

SPACE is used for the study of plasma dynamics in a dense gas filled RF cavities [8]. A representative result is shown in Fig. 4. SPACE is also applied to study processes relevant to eRHIC. Figure 5 shows the plasma-induced reduction of the beam electric field.

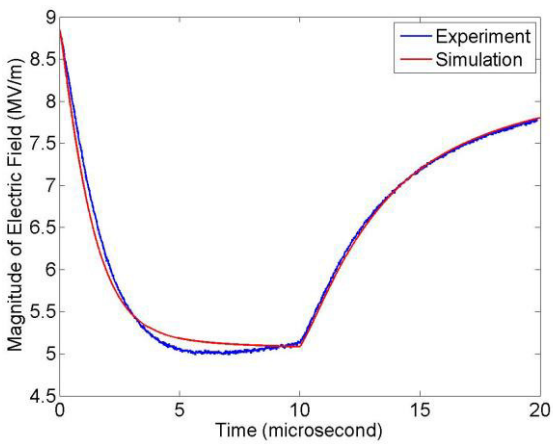

Figure 4: High-pressure RF cavity loading by injection of 2000 proton beams with 5ns spacing. The cavity contains 1470 psi hydrogen gas with 1% of dry air dopant, and is subject to 8.8 MV/m, 808 MHz external electric field. Beam is off at 10 microseconds. Courtesy of [8].

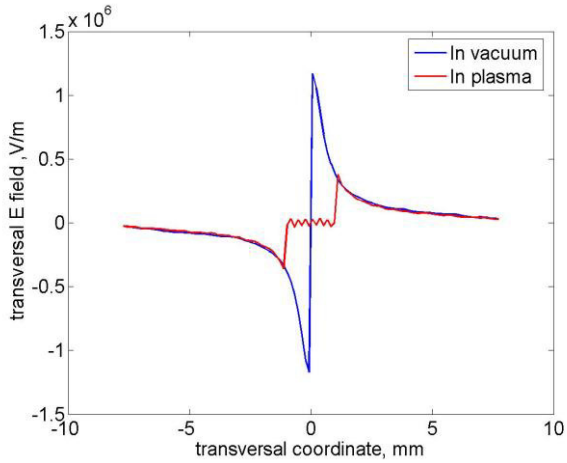

Figure 5: Comparison of electric fields of proton beam in vacuum and plasma (courtesy of [9]).

CONCLUSION

A parallel particle-in-cell code SPACE has been developed for the simulation of electromagnetic fields, relativistic particle beams, and plasmas. The atomic physics algorithms and their implementation in SPACE is validated and used for beam-plasma interaction studies.

ACKNOWLEDGEMENT

This research was partially supported by the DOE Muon Accelerator Project.

REFERENCES

[1] K. Yee, IEEE Transactions on Antennas and Propagation, 14 (3) (1966) 302-307.
[2] J. Villasenor and O. Buneman, Comput. Phys. Commun., 69 (1992) 306-316.
[3] J. Boris, Proc. the 4th Conf. on Numerical Simulation of Plasmas (Washington DC, NRL) (1971) pp 3.67.
[4] J. L. Vay, Physics of Plasmas, 15 (2008), 056701.
[5] https://wci.llnl.gov/simulation/computer-codes/visit/
[6] C. Birdsall, A. Langdon, Plasma Physics Via Computer Simulation, Series in plasma physics, Taylor & Francis, 1991.
[7] D. Neuffer, Introduction to muon cooling, Nuclear Instruments and Methods in Physics Research A, 532 (2004) 26-31.
[8] K. Yu et al., Proceedings of IPAC 2015, MOPMN013.
[9] J. Ma et al., Proceedings of IPAC 2015, MOPMN015.