

THE SOLMAXP CODE

A. Chancé *, N. Chauvin, R. Duperrier, CEA/SACLAY, DSM/Irfu/SACM Gif-sur-Yvette France

Abstract

In modern sciences, use of high performance computing (HPC) became a must to go further in the modelization of complex systems. For large scale instruments like accelerators, HPC permits the virtual prototyping of very onerous parts and, thus, helps to reduce development costs. The SolMaxP code (for Solving Maxwell in Plasma) has been developed to allow complex simulations of multi species plasma coupled with electromagnetic fields, whether the electromagnetic background is or is not self-consistent with the plasma dynamics. This paper presents the main algorithm of the code and gives several examples of applications.

INTRODUCTION

The code SolMaxP is a parallelized PIC (Particle in Cell) finite difference solver of Maxwell’s equations. The code can work with a 3D domain, a 2D domain by assuming an invariance by translation along the y axis and a axisymmetric domain. In the following, we will assume that we use the general case of a 3D domain. The differential equations are solved by using a scheme of finite difference. Each particle distribution is seen as a cloud of macroparticles, which gives an evaluation of the charge and current distribution. The evolution of the cloud distributions is emulated by performing kicks on the particles at regular step. The code implements Message Passing Interface (MPI) library to run on a multiprocessor architecture. The grid is cut in longitudinal slices and uniformly distributed amongst the different nodes. Thus, each node communicates with its two neighbor nodes. Some further improvements could be a different share of the space by using other criteria (charge distribution). At each time step, the code makes a predefined sequence of operations shown on Fig. 1:

- Calculation of the charge and distribution density at each grid node.
- Update of the electric and magnetic fields at each grid node.
- Interpolation of the strength at each particle position.
- Motion integration for each particle.
- Application of collisions.

In the following, we are going to detail the different steps of this algorithm and give a few examples of applications of the code.

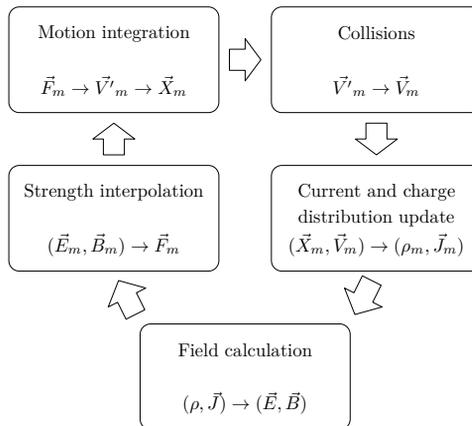


Figure 1: Principle of the SolMaxP algorithm.

ALGORITHM

Calculation of the Charge and Distribution Density

To avoid a statistical noise due to the random sampling of the macroparticles, an aliasing of each particle is done [1]. Each macroparticle is seen with a triangular distribution of which the width is the grid step. The weight of the macroparticle at each node is then the integrate of this distribution centered on the macroparticle in the node. The weights are quadratic functions of the particle position. This method significantly reduces the statistical noise and then the needed number of macroparticles, for a moderate number of extra computations.

Field Update

The code SolMaxP has two resolution modes. In the first mode, the code solves the Poisson’s equation whereas in the second one, it directly solves Maxwell’s equation in a temporal scheme.

In the first mode of SolMaxP, the code makes the quasi-static approximation. The time step is not bounded and the user can specify it freely. At each step, the code solves Poisson’s equation by using Dirichlet or Neumann conditions on the boundaries. The code solves the equation on the grid by doing the conjugate gradient algorithm.

In the second case, a Finite Difference Time Domain is implented. The code uses Yee’s scheme [2, 3] with a finite difference formulation of Maxwell’s equation.

$$\frac{\partial \mathbf{E}}{\partial t} = \frac{1}{\epsilon} (\nabla \times \mathbf{H} - \mathbf{J}) \quad \frac{\partial \mathbf{B}}{\partial t} = -\nabla \times \mathbf{E} \quad (1)$$

* antoine.chance@cea.fr

For example, the Eq. (1) becomes, after discretization:

$$\Delta t \left[\frac{B_x |_{i+1/2,j,k}^n - B_x |_{i+1/2,j,k}^{n-1}}{\Delta y} - \frac{E_z |_{i+1/2,j+1/2,k}^{n+1/2} - E_z |_{i+1/2,j-1/2,k}^{n+1/2}}{\Delta y} - \frac{E_y |_{i+1/2,j,k+1/2}^{n+1/2} - E_y |_{i+1/2,j,k-1/2}^{n+1/2}}{\Delta z} \right]$$

To respect Courant's stability bound, the code sets the time step to $\Delta t = \frac{1}{2c\sqrt{\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} + \frac{1}{\Delta z^2}}}$. The following boundary conditions are considered in the code:

- Perfect electric conductor (PEC).
- Perfectly Matched Layer (PML) for absorbing layers.
- Periodic conditions are being implemented.
- Moving window should be added next.

In this mode, the charge conservation equation is not always respected. An option can be set to make the code correct the field by solving Poisson's equation on the residual error. The four Maxwell's equations are then checked. This mode increases significantly the computation time.

Interpolation of the Strengths

The interpolation of the strengths is similar to the step updating the charge distribution. The electric and magnetic fields are interpolated at the particle position according to the near grid nodes. The code can interpolate the field in maps that the user can have initially specified.

Motion Equation Integration

The code uses Boris method [4]. The velocity vector is then updated from $t = (n - 1/2)\Delta t$ to $t = (n + 1/2)\Delta t$ as described below:

$$\mathbf{v}^- = \mathbf{v}^{n-1/2} + \frac{q\mathbf{E}}{m} \frac{\Delta t}{2} \quad (2)$$

$$\mathbf{s} = \frac{q\mathbf{B}}{m} \frac{\Delta t}{2} \quad (3)$$

$$\mathbf{v}' = \mathbf{v}^- + \mathbf{v}^- \times \mathbf{s} \quad (4)$$

$$\mathbf{v}^+ = \mathbf{v}^- + \mathbf{v}' \times \frac{\mathbf{s}}{1 + |\mathbf{s}|^2} \quad (5)$$

$$\mathbf{v}^{n+1/2} = \mathbf{v}^+ + \frac{q\mathbf{E}}{m} \frac{\Delta t}{2} \quad (6)$$

The particle position is updated by the relation:

$$\mathbf{x}^{n+1} = \mathbf{x}^n + \mathbf{v}^{n+1/2} \Delta t \quad (7)$$

Collisions

The code can manage the five following collisions:

- ionization
- recombination
- inelastic collision
- elastic collision
- dissociation

The algorithm is the following. Firstly, the code evaluates the total collision frequency ν_{tot} for the colliding particle m which is given by:

$$\nu_{\text{tot}} = \sum_i^{N_{\text{coll}}} \rho_{\text{target}} \sigma_i(E_m) v_m \quad (8)$$

where ρ_{target} is the volume density of particles in the target, N_{coll} is the total number of collision processes for the particle m , E_m is its total energy, v_m its velocity, σ_i is the cross section for the collision type i . Secondly, the code calculates the collision probability P_{coll} :

$$P_{\text{coll}} = 1 - \exp(-\nu_{\text{tot}} \Delta t) \quad (9)$$

Thirdly, the code randomly samples a number Γ in the range $[0,1]$. If $\Gamma < P_{\text{coll}}$, the collision occurs. In this case, the code calculates a new random number Γ' and compares it with the relative frequency $\frac{\nu_1}{\nu_{\text{tot}}}, \frac{\nu_1 + \nu_2}{\nu_{\text{tot}}}, \dots$ to identify the collision type which occurs.

Finally, the code modifies the velocity or charge of the particle accordingly to the collision process. Some particles can be created too. The secondary emission can also be simulated and the user can define different materials via STL model.

APPLICATIONS

High-Intensity Injectors

Several high intensity hadron linacs are under development like ESS [5] or IFMIF [6]. At low energy, the particle motion is dominated by space charge effects and matrix code are unreliable. Simulating the motion with PIC code is necessary. Moreover, the interaction of the beam with the residual gas ionizes some neutral particles, which changes the potential map in the chamber. For this reason, the PIC code must be able to accurately simulate these interactions. As an example of application, we will focus on the studies made for IFMIF/EVEDA.

In the first phase of IFMIF, called EVEDA (Engineering Validation and Engineering Design Activities) [7, 8], a 125 mA cw/9 MeV deuteron demonstrator accelerator will be built, tested and operated at Rokkasho-Mura, in Japan. This accelerator is composed by an ECR ion source, a low energy beam transport (LEBT) line with a dual solenoid lenses focusing system, a RFQ, a matching section, a superconducting Half Wave Resonator cavities section, and finally a high energy beam line equipped with a diagnostic plate and a beam dump.

The purpose of the LEBT is to transport the 140 mA/100 keV D+ beam of 0.25π mm.mrad emittance extracted from the ECR source and to match it for its injection into the RFQ. The SolMaxP simulations make possible to determine the SCC (Space Charge Compensation) transient time [7]. The simulation starts with no beam in the line but only at fixed gas pressure. Then, the beam starts to propagate in the beam line and its evolution can be followed. The time it takes for the emittance to stabilize gives

an evaluation of SCC transient time. The quasi-static approximation is used for the calculations.

The space charge potential map is calculated by SolMaxP along the IFMIF/EVEDA LEBT step by step, which enables to evaluate the SCC degree when the equilibrium is reached. The beam transport in the injector and in the first section of the IFMIF RFQ is shown on Fig. 2.

The SolMaxP code is in qualitatively good agreement with experimental results with other injectors (rise time, effect of gas nature). Quantitative experimental confrontations will be done with the beams of the IFMIF/EVEDA injector being commissioned.

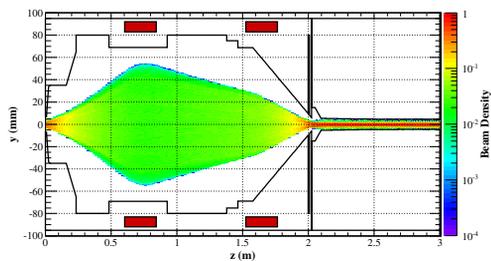


Figure 2: Beam transport simulation in the IFMIF injector and in the first RFQ cells. The red rectangles represent the solenoids.

Laser/Plasma Acceleration

During last decades, the laser plasma acceleration technology has shown a large potential for acceleration with large accelerating gradients (several tens of GV/m). Nevertheless, due to the short wavelength of the laser compared to the domain size, realistic simulations are very CPU-consuming. A special effort must be then done on the implementation of codes for the understanding and simulation of such acceleration schemes. After some modifications, the code SolMaxP was used to make a benchmark with previous calculations [9] made with the PIC code CALDER [10]. The aim was to simulate the interaction of a 30 fs $0.8 \mu\text{m}$ laser pulse of intensity $10^{19} \text{W}/\text{cm}^2$ coming with an incidence angle of 30° on a 30-nm-thick Hydrogen target of electronic density $n = 4.9 \times 10^{28} \text{m}^{-3}$. The code was run for 2 weeks on 64 processors with 128 millions of macroparticles for each species (protons, electrons). The interaction of the laser with the target is shown on Fig. 3. The plasma wave is clearly highlighted. The final energy for the protons was 9.5 MeV, which is in a good agreement with the maximum value found with CALDER: 10 MeV. First results are promising but more simulations and development must be performed to improve the code reliability on the long term.

CONCLUSION

Initially developed to simulate plasmas of ion source [11], the code SolMaxP became a more general tool to study the coupling between plasmas and electromagnetic

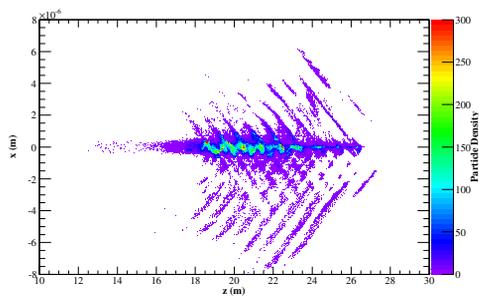


Figure 3: Density of the electrons of more than 1.5 MeV at $t=107$ fs. The target is initially at $x = 0$. The plasma wave is visible in the target. Moreover, the ejected electrons create a wave corresponding to the laser wave.

backgrounds. Its flexibility makes possible to use it for multiple applications like high-intensity injectors or laser-plasma simulations.

The code is still under development and numerous improvements can be brought but the first results look very promising for the simulation of future machines. Some future improvements could be to take into account more physical phenomena, like a precise calculation of the gas pressure along the beam line or the beam scattering on the gas.

REFERENCES

- [1] J. P. Verboncoeur, J. Comput. Phys., 174(2001), p421.
- [2] K.S. Yee, "Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media", IEEE Trans. Antennas and Propagation, Vol. 14, 1966
- [3] A. Taflov, "Computational electrodynamics: the finite-difference time-domain method", second edition.
- [4] J.P. Boris, "Relativistic plasma simulation-optimization of a hybrid code". Proc. of the 4th Conference on Numerical Simulation of Plasmas.
- [5] M. Eshraqi et al., "Conceptual design of the ESS Linac", IPAC'10.
- [6] A. Mosnier, "The IFMIF 5 MW Linacs", in Proc. of the Linear Accelerator Conference, Victoria, Canada, 2008.
- [7] N. Chauvin et al., Rev. Sci. Instrum., 83, 02B320 (2012).
- [8] N. Chauvin et al., "Final design of the ifmif-eveda low energy beam transport line", PAC'09.
- [9] E. Lefebvre et al., New J. Phys., 12(2010), 045017.
- [10] E. Lefebvre et al., Nucl. Fusion 43 (2003), p 629.
- [11] S. Nyckees et al., Rev. Sci. Instrum., 83, 02A349 (2012).