PARSEC: PARALLEL SELF-CONSISTENT 3D ELECTRON-CLOUD SIMULATION IN ARBITRARY EXTERNAL FIELDS

A. Adelmann and M. A. Furman [1]

LBNL, Berkeley [1]

We present PARSEC, a 3D parallel self-consistent particle tracking program which allows electron-cloud calculations in arbitrary external fields. The Lorentz force equation is integrated with time as the independent variable. A 3D parallel Multigrid solver computes the electric field for the drive beam in the beam frame, while the space-charge field of the electrons is computed in the lab frame. The resulting total field, obtained by superposition, acts on both the beam particles and the cloud electrons. Primary and secondary emission takes place at each time step of the calculation. This sort of computation is only possible by the use of massive parallelization of the particle dynamics and the Poisson solver.

INTRODUCTION AND MOTIVATION

The electron-cloud effect (ECE) has been investigated in various storage rings for several years now. The ECE arises from the strong coupling of a two-species plasma with the surrounding vacuum chamber. Several analytical models and simulation programs have been developed to study this effect [1]. Owing to the complexity of the problem, these simulation codes typically make one or more simplifying assumptions, such as: (i) the electrons are dynamical but the beam is a prescribed function of space and time; (ii) the beam is dynamical but the electron cloud is a prescribed function of space and time; (iii) both the beam and the electrons are dynamical, but the electron-wall interaction, particularly the secondary emission process, is either absent or much simplified; (iv) the geometry of the beam and/or vacuum chamber is much simplified (eg. round beams and/or cylindrical chambers); (v) the simulation "looks" at only one specific region of the machine, typically a field-free region or one magnet of a specific kind; (vi) the forces on the particles, both from, and on, the electrons and the beam, are purely transverse. Computer codes involving these approximations, when applied in the proper context, have shed valuable information on one or more aspects of the ECE.

There are problems, however, in which any of these approximations may render the reliability of code inadequate for a quantitative understanding of the dynamics. One such example concerns problems involving very long, intense, bunches with significant variation in the longitudinal profile, which require a self-consistent, fully 3D simulation, including a full description of the storage ring lattice (or at least, a section of the lattice at least as long as the bunch). Another example might be the simulation of damping rings for future linear colliders, which make significant use of wigglers. In this article we report on progress towards the goal of a fully self-consistent and realistic simulation of the ECE which, in its final stage, will not invoke any of the above-mentioned simplifications.

THE OVERALL SIMULATION MODEL

Self-consistent formulation

Let the particle coordinates of particle k be $\vec{x}_k = (q_1, q_2, q_3)_k$, and the normalized velocity be $\vec{\beta}_k = (v_x/c, v_y/c, v_z/c)_k$ where c is the speed of light (all quantities in MKS units unless explicitly stated otherwise). We consider $\ell = 1, 2, \cdots$ magnetic elements which make up what is called the lattice L. Defining $I = \{1, 2, \cdots\}$ and $J = \{1, 2, \cdots\}$ the index sets for the beam particles and electrons, respectively as unique identifiers, we are able to distinguish beam particle ($i \in I$) and electron coordinates ($j \in J$) in a natural way (see Figure 1 as an illustration). For each particle
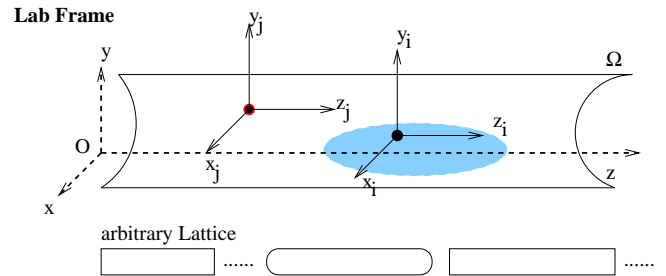

Fig. 1: (color) Geometry and Particle domains.

$k \in I \cup J$ we solve formally

$$\frac{d(m_k \gamma_k c \vec{\beta}_k)}{dt} = \vec{F}(\vec{x}_k, t) \tag{1}$$

$$\vec{F}(\vec{x}_k, t) = \frac{q_k}{\gamma_k m_k}(\vec{E}(\vec{x}_k, t) + \vec{\beta}_k \times \vec{B}(\vec{x}_k, t)) \tag{2}$$

where m_k and q_k are the mass and charge of the particle, respectively, and γ_k its usual relativistic factor. The lattice magnetic field $\vec{B}_{ext}$ in cylindrical coordinates is represented by:

$$B_\rho = \sum c_{m,n} I'_m(nk_z\rho) \sin(m\phi)\cos(nk_z z)$$
$$B_\phi = \sum c_{m,n} \frac{m}{nk_z\rho} I_m(nk_z\rho)\cos(m\phi)\cos(nk_z z) \tag{3}$$
$$B_z = -\sum c_{m,n} \frac{m}{nk_z\rho} I_m(nk_z\rho)\sin(m\phi)\cos(nk_z z).$$

where I_n is the usual modified Bessel function, which satisfies $I'_m(nk_z\rho) = \frac{1}{2}[I_{m-1}(nk_z\rho) + I_{m+1}(nk_z\rho)]$. In this formulation we will treat coasting beams only although it is straightforward to include acceleration. The potential ϕ is obtained by solving two Poisson problems with $\rho_e(\vec{x}_j)$ and $\rho_b(\vec{x}_i)$ the electron and beam charge density. Let $\vec{x}'_i = \mathcal{L}(\vec{x}_i)$ with $\mathcal{L}$ denoting the proper Lorentz transformation from the laboratory to the beam rest frame. The first Poisson problem, in which the beam charge density ρ_b is the source, reads:

$$\triangle\phi(\vec{x}'_i) = -\frac{\rho_b(\vec{x}'_i)}{\epsilon_0},\ \vec{x}'_i \in \Omega \subset \mathcal{R}^3$$
$$\phi(\vec{x}'_i) = 0,\ \vec{x}'_i \in \partial\Omega. \tag{4}$$

Upon Lorentz-transforming back to the Lab frame, this yields both an electric ($\vec{E}_b$) and magnetic field ($\vec{B}_b$). Assuming that the electrons are sufficiently non-relativistic, which is typically a good approximation, we can neglect their contribution to the magnetic field, and we can solve in the laboratory frame for the second Poisson problem, in which the electron-cloud density ρ_e is the source,

$$\triangle\phi(\vec{x}_i) = -\frac{\rho_e(\vec{x}_i)}{\epsilon_0},\ \vec{x}_i \in \Omega \subset \mathcal{R}^3$$
$$\phi(\vec{x}_i) = 0,\ \vec{x}_i \in \partial\Omega \tag{5}$$

thus the full answer is obtained by superposing the two fields:

$$\vec{F}(\vec{x}_k, t) = \frac{q_k}{\gamma m_k}(\vec{E}_e(\vec{x}_j, t) + \vec{E}_b(\vec{x}_i, t) +$$
$$\vec{\beta}_k \times (\vec{B}_{ext}(\vec{x}_k, t) + \vec{B}_b(\vec{x}_j, t))) \tag{6}$$

where $\vec{E}_e$ is the electric self-field of the electrons.

Secondary Emission Model

When an electron strikes the vacuum chamber wall, it can be absorbed or can generate one or more secondary electrons. In our computations we simulate this process by a detailed probabilistic described elsewhere [2]. This process incorporates, as inputs, the measured secondary electron yield (SEY) δ and the energy spectrum of the emitted electrons, $d\delta/dE$ for a given vacuum chamber surface material. The three main subprocesses, namely elastic reflection, rediffusion, and true secondary emission, are included. We are not concerned for the moment with the detailed processes responsible for the generation of primary electrons, chiefly the photoelectric effect, ionization of residual gas, and stray beam particles striking the vacuum chamber wall; rather, we simulate these processes with simple phenomenological probabilistic models.

Time integration

The code integrates (1) using a 4th-order Runge-Kutta method, with adaptive time step control for the electrons. We estimate the time step T_j by considering the cyclotron frequency $\omega_c = eB/m_e$, consequently

$$T_j = \frac{2\pi}{\omega_c K}. \tag{7}$$

is defined upon the factor K. Defining ϵ_{min} and ϵ_{max} the minimum and maximum error tolerated we estimate K using two Runge-Kutta steps and Richardson Extrapolation [3]. Choosing an initial step size T_j by setting $K = 1$ and let u_1 be the result of a Runge-Kutta step of length T_j. Let u_2 be the result of two subsequent Runge-Kutta steps of length $T_j/2$. We then estimate the error by $\epsilon = |u_1 - u_2|$. Richardson Extrapolation is then used to set K corresponding to a predefined error range. This procedure will guarantee the minimal work necessary to achieve a desired accuracy, considering the dynamics of the individual particle.

Particle Tracking Procedure

As a first step towards a full lattice simulation, we model a portion of the magnetic lattice by imposing periodic boundary conditions in the z coordinate, for example one half of the PSR circumference or one FODO cell of the LHC arc. Further, we assume a constant number of particles in the drive beam, and a fixed number N of time steps of size ΔT. All electrons $i \in I$ and protons $j \in J$ are advanced by $N\Delta T$ followed by a space-charge calculation. The required Poisson solver is described elsewhere in this annual review.

CONCLUSION

The presented code PARSEC is based on GenTrackE, which is written in C++ and is fully parallelized using MPI. PARSEC advances the macro particles of the drive beam and the electrons using a 4th-order Runge-Kutta method. Variable time steps for the electrons according to their dynamics are used. The arbitrarily shaped computational domain is discretized using linear finite elements, and the resulting linear system of equations is solved efficiently by the use of a massive parallel and scalable Multigrid solver.

We are finalizing the code construction and are about to start simulation of a simplified LHC FODO cell, as well as a section of the Los Alamos proton storage ring. The issue of large aspect ratios in the computational domain and the impact of the accuracy of the Poisson solver will be investigated in detail, as this is a subject which is of general importance in many space-charge dominated problems.

REFERENCES

[1] Various contributions to the Proceedings of the ECLOUD-02 Workshop [CERN Yellow Report no. CERN-2002-01], http://slap.cern.ch/collective/ecloud02/.

[2] M. A. Furman and M. T. F. Pivi, PRST-AB **5** 124404 (2002).

[3] R. Bulirsch and J. Stoer, Introduction to Numerical Analysis. New York: Springer-Verlag, 1991.