Parallel Beam Dynamics Simulation of Linear Accelerators

Ji Qiang and Robert D. Ryne

MS 71J, Accelerator and Fusion Research Division
Lawrence Berkeley National Laboratory
1 Cyclotron Road
Berkeley, CA 94720

Abstract

In this paper we describe parallel particle-in-cell methods for the large scale simulation of beam dynamics in linear accelerators. These techniques have been implemented in the IMPACT (Integrated Map and Particle Accelerator Tracking) code. IMPACT is being used to study the behavior of intense charged particle beams and as a tool for the design of next-generation linear accelerators. As examples, we present applications of the code to the study of emittance exchange in high intensity beams and to the study of beam transport in a proposed accelerator for the development of accelerator-driven waste transmutation technologies.

1 Introduction

The emergence of high performance, large memory parallel computers has made large scale simulation an indispensible tool for the design of next-generation accelerators. This is especially true in regard to high intensity accelerators for applications such as spallation neutron sources, proposed waste transmutation systems, and accelerator-based fission and fusion energy concepts. For all these systems, the self fields within the beam (i.e. the space-charge fields) play an important role in the beam dynamics. And for all these systems, large scale simulation is essential for feasibility studies, for design optimization, and for design decisions aimed at evaluating and reducing risk and cost.

Particle simulation methods are widely used in the accelerator community [1, 2, 3, 4, 5]. However, the self-consistent simulation of high-intensity beams, especially in three-dimensions, is computationally intensive. For this reason parallel simulation codes are essential. Such codes have been developed and are now being used in the accelerator community [1, 5]. In this paper we will describe the methods used to simulate radio frequency (RF) linear accelerators. In Section 2 we will provide an overview of the mathematical model, numerical methods, and their implementation on parallel computers. Section 3 presents two applications, one from a beam physics study of longitudinal/transverse coupling in space-charge dominated beams, and the other from a study of a novel beamline design for spoke-cavity superconducting linacs.

2 Parallel Particle-In-Cell Methods

The Vlasov-Poisson model is widely used in the study of beam dynamics in RF accelerators. In this approach, the charged particles are viewed as being subject to externally applied fields and a self-field (in the mean-field approximation) associated with the beam's space charge. Collisional effects are normally negligible in linear accelerators, although they are important in circular machines. Also, the Poisson equation may be used (as opposed to the full Maxwell equations) because, in many situations, the motion in the bunch frame is nonrelativistic. The resulting quasi-static model still includes some relativistic effects because it takes account of the reduction in the space-charge force due to the self-magnetic field associated with the beam current.

The design of RF accelerators is normally performed using the arc length, rather than the time, as the independent variable. (In linear accelerators, this is normally chosen to be the z-axis of a Cartesian coordinate system.) Futhermore, one is usually interested in motion around some reference trajectory. Therefore, accelerator simulations are traditionally done using deviations from the reference trajectory as the dependent variables of the analysis. For the analysis of circular accelerators, it is essential that the numerical model preserve the symplectic (i.e. Hamiltonian) nature of the dynamics. For such cases the computer model is based on a solution of Hamilton's equations where the Hamiltonian includes the electromagnetic potentials associated with the external fields as well as the mean-field potentials associated with the space-charge fields. For linear accelerators, and for single-pass systems, the preservation of the symplectic condition is not as critical. In such cases, the computer model may integrate the Lorentz force equations, where the electric and magnetic fields in the equations have contributions from the externally applied fields and the space-charge fields. In the Lorentz force approach, the equations of motion using z as the independent variable are given by

$$x' = \frac{p_x}{p_z} \tag{1}$$

$$y' = \frac{p_y}{p_z} \tag{2}$$

$$\psi' = \frac{\frac{\omega}{c}\gamma}{p_z} - \frac{\frac{\omega}{c}}{\beta_0} \tag{3}$$

$$p_x{}' = \frac{q}{m_0 c p_z}(\frac{\gamma}{c}\mathbf{E} + \mathbf{p}\times\mathbf{B})_x \tag{4}$$

$$p_y{}' = \frac{q}{m_0 c p_z}(\frac{\gamma}{c}\mathbf{E} + \mathbf{p}\times\mathbf{B})_y \tag{5}$$

$$p_t{}' = \frac{q}{m_0 c^2}E_{z0} - \frac{q}{m_0 c^2 p_z}\mathbf{p}\cdot\mathbf{E}. \tag{6}$$

In these equations, $p_x = \gamma\beta_x$, $p_y = \gamma\beta_y$, $p_z = \gamma\beta_z$, $\gamma = 1/\sqrt{1-\beta^2}$, $\beta_i = \frac{v_i}{c}$ with $i = x, y, z$, ψ is the phase relative to the reference particle defined by $\psi = \omega(t - t_g)$, ω is a constant with units of the angular frequency, t_g is the flight time of the reference particle, p_t is the normalized energy deviation with respect to the reference particle, $p_t = \gamma_g - \gamma$, where γ_g is the γ of the reference particle, c is the speed of light, and m_0 is the rest mass of the particle. The trajectory of the reference particle on the axis of the accelerator can be determined from the following:

$$t_g{}' = \frac{1}{\beta_0 c} \tag{7}$$

$$\gamma_g{}' = \frac{q}{m_0 c^2}E_{z0}(z,t) \tag{8}$$

where E_{z0} is the on-axis external electric field (with the on-axis space-charge field assumed to be zero at the location of the reference particle), and $\beta_0 = \sqrt{1 - 1/\gamma_g^2}$.

The mean-field Coulomb potential can be obtained from the solution of Poisson's equation in the beam frame subject to different boundary conditions [6]. In our computer model we have included six types of boundary conditions. They are: open in all three dimensions, open in the transverse direction and periodic in the longitudinal direction, conducting walls in the transverse direction and open in the longitudinal direction, conducting walls in the transverse direction and periodic in the longitudinal direction. We treat two types of conducting walls, a circular or a rectangular pipe.

This model, along with a symplectic model based on Hamilton's equations and split-operator methods, has been implemented in an object-oriented 3D parallel particle-in-cell code called IMPACT (Integrated Map and Particle Accelerator Tracking Code) [1]. The following describes the parallel implementation of the model.

The IMPACT code employs the two-dimensional domain-decomposition approach of Liewer and Decyk [7]. The physical computational domain is defined as a 3-dimensional rectangular box containing all charged particles within the range $x_{min} \leq x \leq x_{max}$, $y_{min} \leq y \leq y_{max}$, and $z_{min} \leq z \leq z_{max}$. This domain is decomposed on the $y - z$ plane into a number of small rectangular blocks. These blocks are mapped to a logical two-dimensional Cartesian processor grid. Each processor contains one rectangular block domain. The particles with spatial positions within the local computational boundary are assigned to the processor containing that part of the physical domain.

The particles generated on each processor are advanced using a second order leap-frog algorithm to solve the equations of motion. If a particle moves outside the local computational domain, it is sent to the corresponding processor where it is located. A particle manager function, implemented using the message passing interface (MPI), is defined to handle the particle movement and the inter-processor communication. The particle manager first checks the y and z positions of every particle on each processor. If these positions are outside the local computational domain, the particle is copied to one of its four buffers and sent to one of its four neighboring processors. After a processor receives particles from its neighbors, it determines whether some of them need to be further sent out or not. The outgoing particles are counted and copied into four temporary arrays. The remaining particles are copied into another temporary array. This process is repeated until there is no outgoing particle found on all processors. Then, the particles in the temporary storage, along with the particles left in the original particle array, are copied into a new particle array.

After each particle moves to its local computational domain, a linear cloud-in-cell (CIC) particle-deposition scheme is done for all processors to obtain the charge density on the grid. For the particles located between the boundary grid and computational domain boundary, these particles will also contribute to the charge density on the boundary grids of neighboring processors. Hence, explicit communication is required to send the charge density on the guard grids, which is from the local particle deposition, to the boundary grids of neighboring processors to sum up the total charge density on the boundary grids. With the charge density on the grids, the Poisson equation is solved subject to different boundary conditions to obtain the potential on the grid [6]. From the potential on the grid, we calculate the electric field on the grid using a central finite difference scheme. To calculate the electric field on a boundary grid, the potential on a boundary grid of neighboring processors is required. A communication pattern similar to that employed in the charge density summation on the boundary grids is used to send the potential from the boundary grids to the guard grids of neighboring processors. After the electric field on the grids is obtained, it has to be interpolated from the grids onto the local particles to advance the particles. Since we have used the linear CIC scheme, the electric field of particles between the boundary grid and computational domain boundary will also depend on the electric field on the boundary grid of neighboring processors. A similar communication pattern is used to send the electric field from the boundary grids to the guard grids of the neighboring processors. With the electric field on grids local to each processor, the interpolation is done for all processors to obtain the space-charge force on every particle. The local particles are updated in momentum space based on the space-charge forces and external forces.

Dynamic load balancing is employed with adjustable frequency to keep the number of particles on each processor approximately equal. A density function is defined to find the local computational domain boundary so that the number of particles on each processor is roughly balanced. This number depends on the local integration of the charge density on each processor. To determine the local boundary, first, the three-dimensional charge density is summed up along the x direction on each processor to obtain a two-dimensional density function. This function is distributed locally among all processors. Then, the two-dimensional density function is summed up along the y direction to get the local one-dimensional charge density function along z. This density function is broadcast to the processors along the y direction. The local charge density function is gathered along z and broadcast to processors along the z direction to get a global z direction charge density distribution function on each processor. Using this global z direction density distribution, the local computational boundary in the z dimension can be determined assuming that each processor contains a fraction of the total number of particles about equal to $1/nproc_z$. Here, $nproc_z$ is the number of processors along the z direction in the two-dimensional Cartesian processor grid. A similar process is used to determine the local computational boundary in the y direction. Strictly speaking, the above algorithm will work correctly for a two-dimensional density distribution function which can be separated as a product of two one-dimensional functions along each direction. However, this algorithm works reasonably well for the distributions generally produced in beam dynamics simulations in linear accelerators. Fig. 1 shows the maximum and minimum number of particles per processor with and without dynamic load balance. With dynamic load balancing, the difference between the maximum number of particles and the minimum number of particles has been drastically reduced. Fig. 2 shows the excution time as a function of the processor number on the Cray T3E-900 and on the SGI Origin 2000. Good scalability of the parallel PIC code is evident.

3 Applications

An important issue for future high intensity linacs is that there may be emittance transfer betweeen the longitudinal and transverse dimensions of the beam phase space that would degrade the beam quality and possibly lead to particle loss. We have studied this issue using IMPACT. Fig. 3 shows the relative emittance growth factor ($\Delta\epsilon/\epsilon_0$) in both directions as a function of longitudinal to transverse tune ratio in a three-dimensional constant focusing channel with initial longitudinal to transverse emittance ratio 2.0 and transverse tune depression 0.5. There are three pronounced resonance stop bands near tune ratios 2:1, 2:2, and 5:2. Within each stop band, there is significant emittance transfer from the longitudinal plane to the transverse plane. Outside the stop band, the

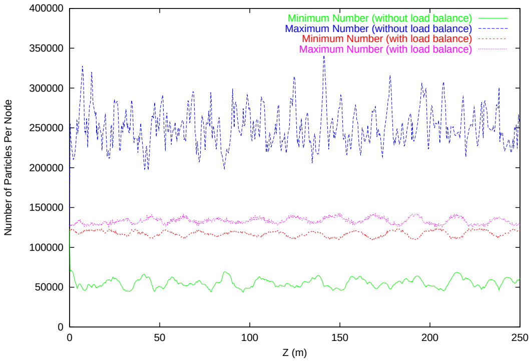

Figure 1: The maximum and minimum number of particles per processor with and without dynamic load balance.

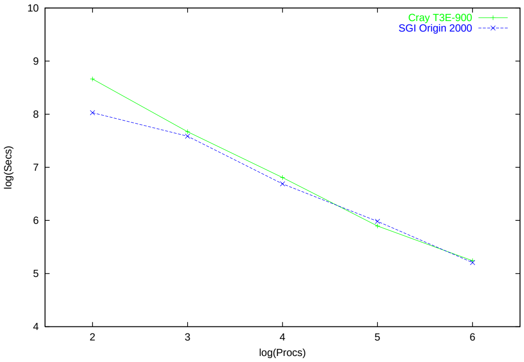

Figure 2: The time cost as a function of the number of processors on the Cray T3E and the SGI Origin 2000.

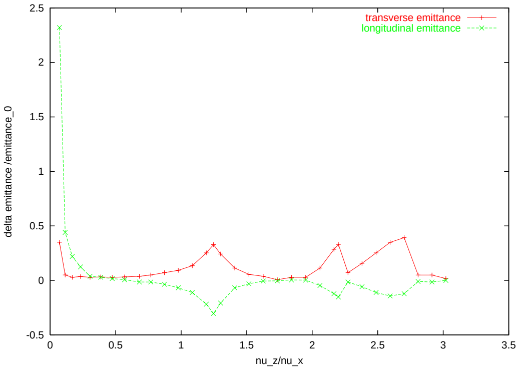

Figure 3: Emittance growth as a function of tune ratio.

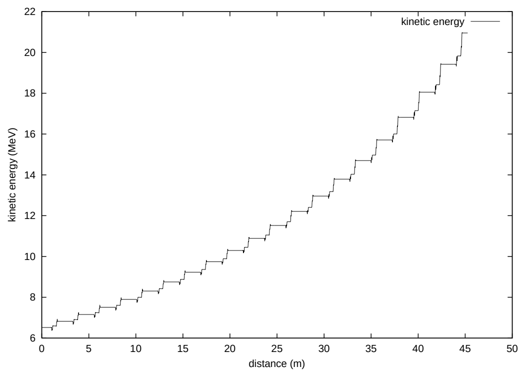

Figure 4: Kinetic energy of beam as a function of distance.

coupling between the two directions is weak, and the beam can remain non-equipartitioned for a long time. The stop bands obsevered in Fig. 3 are caused by nonlinear space-charge forces which drive collective oscillations of charge density that lead to the longitudinal/transverse coupling when the resonance condition is satisfied. These stop bands can be related to the instability of non-oscillatory modes from theoretical studies [8] and are identified as a third order mode, fourth order mode and seventh order mode, respectively.

We have also used IMPACT to study novel designs for future accelerators. As an example, we have studied longitudinal alternating phase focusing in the proposed Accelerator-Driven Test Facility. One period of the accelerator consists of a superconducting solenoid for transverse focusing and two superconducting spoke RF cavities for acceleration and longitudinal focusing. By adjusting the drive phase of the two cavities, we can achieve the same acceleration but with one cavity focusing and the other cavity defocusing. This will reduce the total zero current phase advance in one period and will potentially avoid the envelope instability occuring at large zero current phase advance. The use of longitudinal alternating phase focusing also reduces the transverse focusing required of the solenoid. Fig. 4 shows the kinetic energy as a function of distance within 20 periods of accelerator structure from the simulation. The average acceleration gradient used here ramps up from 3 MV/m to 8.32 MV/m, and the solenoid magnetic field is 2.3 Tesla. We see that after 45 meters, the kinetic energy has reached 20 MeV. Fig. 5 shows the fractional loss at the end of the simulation as a function of input current. Beyond 50 mA, the particle loss increases significantly. This sets the upper boundary for the beam intensity in this design. Numerical simulations such as this are being used to explore new concepts that will enable future superconducting linacs to operate with high fields without encountering beam instabilities.

Aknowledgements

This work used resources of the National Energy Research Scientific Computing Center, which is supported by the U.S. Department of Energy, Office of Science (USDOE/SC) under Contract No. DE-AC03-76SF00098. This work also used resources of the LANL Advanced Computing Laboratory which is supported by the USDOE/SC. Support for this research was provided by the USDOE/SC Division of High Energy and Nuclear Physics under the project, Advanced Computing

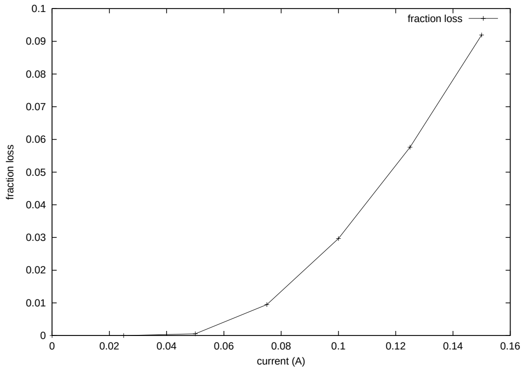

Figure 5: The fraction of particle loss as a function of beam current.

for 21st Century Accelerator Science and Technology, and by the Office of High Energy Physics under the auspices of the Los Alamos Accelerator Code Group.

References

[1] J. Qiang, R. D. Ryne, S. Habib, V. Decyk, J. Comput. Phys. **163**, 434 (2000).

[2] R. Ryne and S. Habib, in: Computational Accelerator Physics, ed. J. J. Bisognano and A. A. Mondelli, AIP Conference Proceedings 391, Woodbury, p. 377, New York, 1997.

[3] H. Takeda and J. H. Billen, "Recent Developments of the Accelerator Design Code PARMILA," Proc. XIX International Linac Conference, Chicago, IL, August 1998, pp.156-158.

[4] J. Qiang, R. D. Ryne, B. Blind, J. H. Billen, T. Bhatia, R. W. Garnett, G. Neuschaefer, H. Takeda, Nucl. Instr. and Meth. A **457**, 1 (2001).

[5] A. Friedman, D. P. Grote and I. Haber, Phys. Fluids **B 4**, 2203 (1992).

[6] J. Qiang and R. D. Ryne, Comput. Phys. Comm. **138**, 18 (2001).

[7] P. C. Liewer and V. K. Decyk, J. Comput. Phys. **85**, 302 (1989).

[8] I. Hofmann, Phys. Rev. **E 57**, 4713 (1998).