

Aspects of Parallel Simulation of High Intensity Beams in Hadron Rings

Alfredo U. Luccio § and Nicholas L. D'Imperio

Brookhaven National Laboratory. C-AD Department. Upton, NY 11973, USA †

Abstract. A PIC code to simulate high intensity beams in hadron circular accelerators, *BNL-Orbit*, has been implemented on parallel computers already in 1999. The issues of full 6-dimensional tracking and the relevant approximations are discussed, as well as solvers for the space charge problem in the presence of walls.

1. Introduction

The code *Orbit* [1] has been designed for PIC tracking of a particle beam in a high intensity circular hadron accelerator. In the code, space charge forces are continuously calculated and applied to the individual macroparticles of the herd as transverse momentum kicks and as longitudinal energy kicks.

The Brookhaven edition of the code *Orbit*, or *BNL-Orbit* was made fully MPI [2] parallel, typically running on a Unix Linux platform. The parallelization of *Orbit*, with space charge calculation done in three dimensions, is structured around the concept of longitudinally partitioning the beam into segments, the number of which can be varied. The parallelization had to possess the capability to efficiently handle almost any beam configuration ranging from coasting beams of uniform longitudinal density with lengths equalling the circumference of the machine to bunched beams of non-uniform longitudinal density with lengths a small fraction thereof.

2. Split Operator. Long Bunches

In *Orbit* the propagation of the beam is controlled by a Split Operator technique. At each stage the herd is transformed through maps calculated for a bare lattice, followed by the application of space charge kicks. Maps are provided by an optical program as MAD [3] and are arranged sequentially along the circumference of the machine. Space charge kicks are applied at certain locations in the lattice: "SC nodes".

In PIC simulation the independent variable can be either time t , or space s . While time can be a natural choice, because the space charge interaction must be calculated

§ To whom correspondence should be addressed (luccio@bnl.gov)

† Work performed under the auspices of the U.S. Department of Energy

with the position of all the macros at the same time, in a cyclic accelerator, where particle traverse the same position in the lattice many times, the position is a convenient clock. *Orbit* uses s as the independent variable.

To solve the space charge problem in the presence of accelerator chamber walls, we use a pre-calculated impedance budget, or direct calculation of the electromagnetic field from charges and currents in the beam and image charges and currents induced on the wall. Impedances are useful to represent lump properties of the entire wall structure, direct calculation is done to address in detail local wall properties.

Given beam and wall image charge and current distribution, and assuming steady state current flow, the electromagnetic problem can be solved via two partial elliptic differential equations, Poisson Law and Ampere Law (in the Coulomb's gauge)

$$\nabla^2\Phi(P) = -\frac{\rho(Q)}{\epsilon_0}, \quad \nabla^2\vec{A}(P) = -\frac{\vec{j}(Q)}{\mu_0}, \quad (1)$$

with ρ the beam charge distribution, and \vec{j} the current distribution at a source point Q . The solution would yield the scalar electric potential Φ and the magnetic vector potential \vec{A} in each field point P , from which space charge kicks are calculated.

To find ρ , one should bin the macroparticles on a suitable grid according to their position coordinates (x, y, z) , and to find \vec{j} one should bin the herd according to the momentum coordinates $(p_x, p_y, \Delta p/p)$.

For long bunches, as it is common in circular accelerators like synchrotrons, *Orbit* makes at the present the customary approximative assumption that the flow of beam current is parallel to the walls, thus representing the partial compensation between space charge repulsion and space current attraction by only solving the Poisson equation and multiplying the kicks by a factor $1/\gamma^2$. Then, transverse momentum kicks and longitudinal energy kicks assume the form

$$\frac{\delta p_{\perp}}{p} = \wp \frac{\partial \phi}{\partial r} L_T, \quad \frac{\delta \Delta E}{E} = \beta^2 \wp \frac{\partial \phi}{\partial z} L_s, \quad (2)$$

where L_T and L_s are the "lengths" of a kick, and \wp the *perveance*

$$\wp = \frac{4\pi\lambda q h r_0}{\Delta x \beta^2 \gamma^3 m_0}, \quad (3)$$

with λ the longitudinal current density, h the harmonic number and Δx the (Cartesian) grid mesh size.

In principle, equation (1) should be solved in 3 Dimensions. For a numerical treatment the herd is binned to a grid of points and the equation is solved on that grid by finite difference methods. The transverse grid is terminated at the wall boundary, and the longitudinal grid covers the whole length of the beam bunch. For long bunches in sychrotrons it is impractical to make the longitudinal grid step as small as the transverse, but in practice it is not necessary to do so, because the longitudinal space charge distribution varies only smoothly along the beam, and the longitudinal motion of particles within the beam is much slower than the transverse motion. The longitudinal grid will then sensibly consist of beam segments, long enough that the average density in each segment, the transverse aspect ratio of the segment, and the wall configuration

around the segment can be considered constant. In *Orbit* we then write for the beam space charge

$$\rho(x, y, z) = \rho_u(x, y) \rho_z(z). \quad (4)$$

with $\rho_z(z)$ a constant within a segment.

This approximation simplifies the problem, since we can now only solve the transverse Poisson problem simultaneously in each segment by parallel computation.

3. Problem Decomposition and Load Balancing

In two dimensions the space charge calculations in *Orbit* take place at given points called nodes, situated around the circumference of the ring. The herd of macro particles, or macros, which compose the beam, arrive at each node independent of time and are transversely represented as a flat disk. The particles are then binned onto a 2-d mesh from which the potential is calculated using a sparse LU solver. The directional derivative of the potential is the respective component of the force which is then applied to each particle as a kick proportional to the length of the space charge element. Proceeding in this manner the entire ring is traversed for a given number of turns.

The parallelization for 2-d divides the herd as evenly as possible over the number of processors. The processors then separately track their particles around the ring. At a space charge node, the processors bin their particles onto a local mesh and then communicate the local meshes to a global mesh. The calculations then proceed on the global mesh as before. This scheme scales linearly.

In three dimensions a different approach must be used. The simulation can no longer be independent of time, because to correctly represent the longitudinal interactions between adjacent beam segments, all segments must be populated by macros all considered at the same time. Therefore, at each space charge node the beam, still represented as a flat disk up to this point, is expanded longitudinally to bring each particle to their appropriate position at a given time.

The beam can then be divided into longitudinal segments whose boundaries are delineated by the space charge elements in the ring. The processes each take a number of these segments and do all calculations independently. The problem is still decomposed by subdividing the herd, however, the subdivision of the herd is dependent on the longitudinal locations of the particles.

In a typical run for a ring with K space charge elements, N total macros, and P processes, N/P macros would initially be injected into the ring by each process. Each macro has no constraint regarding its longitudinal position upon injection and therefore may be found in any of the K elements. When the first space charge element is encountered, the processes synchronize and expand their respective herds longitudinally. The ring is spatially decomposed along its length and so each process is assigned K/P space charge elements with one of the processes taking the remainder. The macros are then exchanged among the processes based on their longitudinal positions. After the

exchange each process contains all the macros in the global herd that belong to its K/P slices. The communication involved is large only for the first space charge element as the synchrotron motion is relatively slow and particles will infrequently migrate between processes. The processes then do a 2-d transverse space charge calculation for each of their K/P slices before collapsing the beam and continuing to track. The process repeats itself at the next space charge element though with less particle exchange.

This idea works well for a uniform beam. If the beam is not longitudinally uniform the simulation will not be efficiently load balanced, as one process may have many more macros than another. Therefore, several factors must be considered when decomposing the problem over the process domain.

The computational burdens which have the greatest effect on the performance of the code are dependent on two variables. The number of space charge elements over which the Poisson equation must be solved and the number of macros in the herd. Therefore, rather than simply dividing the the number of elements evenly among the processes it is more efficient to consider the number of elements assigned as a function of the number of macros contained within them. *BNL-Orbit* therefore dynamically calculates an optimal decomposition scheme to balance the load at each space charge element, as shown in figure 1.

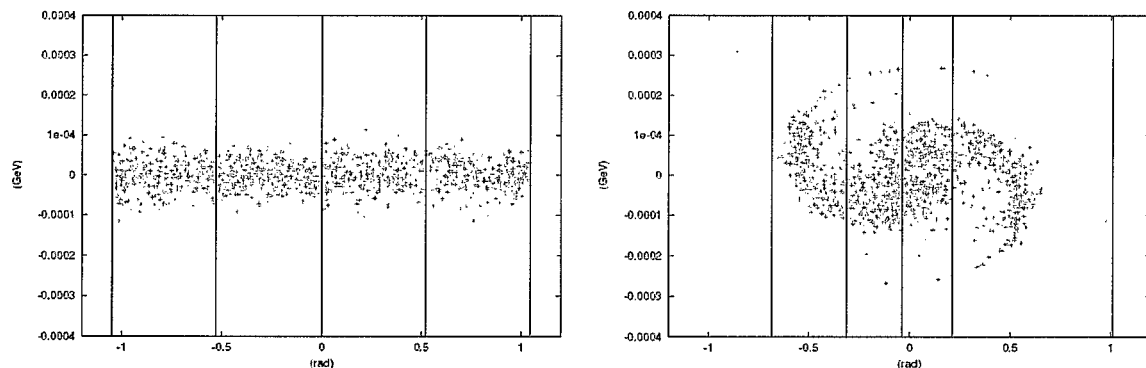


Figure 1. Process decomposition superimposed on the longitudinal phase space of a herd at (a) injection and (b) at a later time. Example for the SIS [4]

4. Details of Longitudinal Beam segmentation

The longitudinal beam segmentation is established using as a guide the shape of the beam envelope represented by the square root of the twiss functions β . The length of a segment is a fraction of a β -wavelength, as shown schematically in figure 2. The local accelerator chamber profile is associated to each segment. When the herd, that can be imagined as a flat disk, reaches a SC node at a longitudinal position s_{SC} , all the macros are there at different times. To calculate space charge kicks, the beam bunch is then reconstructed bringing each macro to the position s at the (center of the) segment where it was or will be at a common time, using the transfer maps between s and s_{SC} . After

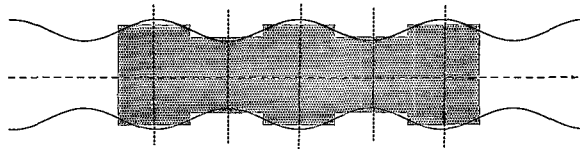


Figure 2. Slicing a beam. The wavy lines represent the envelope of the beam (*beta-wave*). Dashed vertical lines represent planes where the Poisson equation is solved.

kicks are calculated and applied to the individual macros, the beam is again flattened and transferred to the next node. Figures 3 show a gaussian beam bunch thus expanded in a FODO channel. Transfer maps for the expansion are for the *bare lattice*, within the

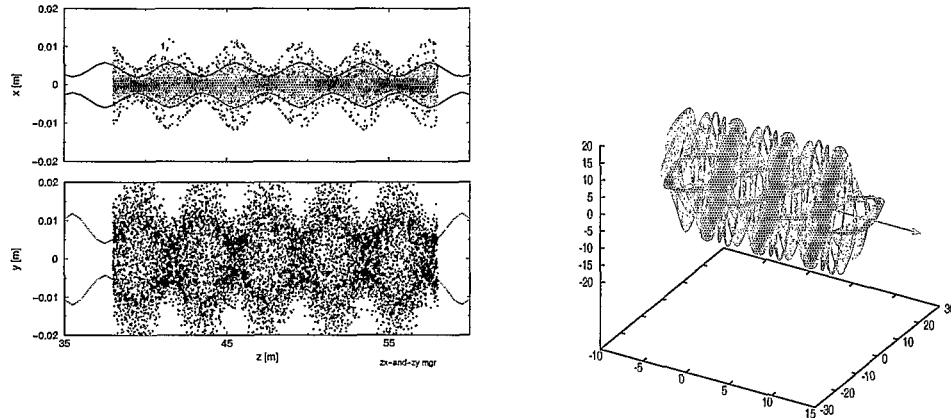


Figure 3. Expanded Gaussian beam and its 3D rendition. In the lhs figure the beam envelope is also shown

very approximation that leads to the concept of split operators.

5. 2D Poisson Solvers

The differential formulation of Poisson Law is in equation (1), the integral expression for the potential in a field point P is

$$\Phi(P) = \frac{1}{4\pi\epsilon_0\gamma^2} \int \frac{\rho(Q)}{r} dQ. \quad (5)$$

calculated by integration over the distribution at the source point Q with a Green function $G(r) = 1/r$, $r = |P - Q|$.

In an integral formulation the image charge distribution on the walls is part of the *input* of the problem and must be calculated in advance, conversely, in a differential formulation the image is part of the *solution*.

Equation (5) can be solved by direct integration (Brute Force) or the integral can be reduced to a convolution between the FFT transforms of $\rho(Q)$ and $G(r)$.

$$\Phi(r) = Const \times FFT^{-1} \left(\tilde{G}(\omega) * \tilde{\rho}_{\perp}(\omega) \right).$$

Both BF and FFT are implemented in *BNL-Orbit*.

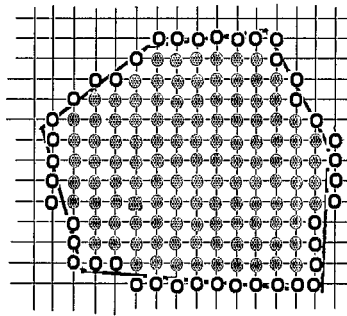


Figure 4. Solving with perfectly conducting walls

The differential Poisson Equation, including boundary condition on the walls (Dirichlet condition for the function or Neumann condition for its derivative at the walls) can be numerically dealt with in various ways. At the present *BNL-Orbit* is limited to perfectly conducting walls, where it is $\Phi(P_{walls}) = 0$. The field outside is zero and, by Gauss's Theorem, the total wall (image) charge is equal to the beam charge.

Let us discretize equation (1) on a $M \times N$ Cartesian grid with equal spacing in the transverse coordinates x, y . The equation and its solution (implicit sum on subscript and superscript indices) can be written as

$$-4\pi\rho_{ij} = \mathcal{L}_{ij}^{kl}\Phi_{kl}, \quad \Phi(P) = -\frac{1}{4\pi}\mathcal{L}^{-1}\rho(Q). \quad (6)$$

Use the second order expression for the second partial derivative (in x)

$$\frac{\partial^2\Phi}{\partial x^2} = \frac{1}{h^2}(\Phi_{i-1,j} - 2\Phi_{i,j} + \Phi_{i+1,j}), \quad (7)$$

and write the Laplacian matrix ∇^2 in discrete form over a Cartesian grid that extends to the wall

$$\mathcal{L}_{ij}^{kl} = -4\delta_i^k\delta_j^l + \delta_{i+1}^k\delta_j^l + \delta_{i-1}^k\delta_j^l + \delta_i^k\delta_{j+1}^l + \delta_i^k\delta_{j-1}^l. \quad (8)$$

The set (6) is a system of linear equations. Figure 4 schematically suggests how to achieve a solution for perfectly conducting walls. Walls are mapped to n empty dots, and the interior to m full dots. The system of equations is exactly determined, with $n + m$ known quantities, i.e. $\Phi = 0$ at the n empty dots and ρ at the m full dots, and $m + n$ unknowns, i.e. m values of Φ to be calculated at the full dots and ρ_{image} at the n empty dots.

The solution is found in *BNL-Orbit* by using one of two alternate methods, depending on the problem: (i) by LU decomposition and (ii) by iteration.

(i) The Laplacian is a symmetric band sparse matrix. To solve the system of equations of the form $Ax = b$ we use the LU decomposition for sparse matrices. By replacing the coefficient matrix A by its LU factorization we have $Ax = (LU)x = L(Ux) = Ly$, where $y = Ux$; This reduces one equation, $Ax = b$, to two equations, $Ly = b$ and $Ux = y$. Using forward substitution, one calculates y , after which x is calculated using backward substitution. The LU factorization of a symmetric band sparse matrix is itself a symmetric band sparse matrix and so is done only once at

the beginning of the program and stored in memory. This is very efficient for small to moderate grids where storage of the sparse LU factorization is manageable [5].

(ii) Instead of LU Decomposition, the discretized Poisson's can be solved by iteration. From equations (6) and (8) obtain

$$\Phi_{i,j} = \frac{1}{4} (\Phi_{i-1,j} + \Phi_{i,j+1} + \Phi_{i+1,j} + \Phi_{i,j-1} - \rho_{i,j}), \quad (9)$$

and solve this by iteration, starting with a guess. At iteration k it is

$$\Phi_{i,j}^{k+1} = \frac{1}{4} (\Phi_{i-1,j}^k + \Phi_{i,j+1}^k + \Phi_{i+1,j}^k + \Phi_{i,j-1}^k - \rho_{i,j}).$$

Since the beam density generally evolves slowly from one space charge node to the next, iterative techniques benefit through more rapid convergence. Several techniques were applied to the problem including: Successive Over Relaxation (SOR), SOR with Chebychev acceleration, and Conjugate Gradient (CG). Preconditioned CG was also considered but memory considerations precluded its use. As expected, CG showed the most rapid convergence, however, the basic algorithm requires more operations than either SOR technique and was therefore less efficient in this case. It was also found that basic SOR was most efficient for small grids ($N < 128$) while SOR with Chebychev acceleration was most efficient for large grids ($N > 128$). The number of iterations required for convergence to 8 significant digits was approximately $4N$ for both SOR methods using optimal relaxation parameters.

6. Approximations for the Longitudinal Dimension

In first approximation, as discussed before, we solve the 3D problem by segmenting the beam longitudinally and solving the 2D equation (1) simultaneously in each segment. In this approach, the longitudinal space charge forces will be calculated simply by taking the differences of the potential at a given radius between adjacent slices. Still, within this model we can do something better that takes into account details of the longitudinal distribution. A perturbative approach is the following

In 3D, using the decomposition of equation (4), Poisson becomes

$$\Phi_z \nabla_{\perp}^2 \Phi_u + \Phi_u \frac{\partial^2 \Phi_z}{\partial z^2} = -\frac{1}{\epsilon_0} \rho_z \rho_u \quad (10)$$

To 0-th order both $\rho_z(z)$ and $\Phi_z(z)$ are piece-wise constant

$$\begin{cases} \rho_z(z) \approx \rho_{\parallel} \\ \Phi_z(z) \approx \Phi_{\parallel} \end{cases}, \quad \frac{\partial^2 \Phi_z}{\partial z^2} \approx 0.$$

Equating z functions and (x, y) functions on both sides, this yields

$$\Phi_{\parallel} = \rho_{\parallel}, \quad \nabla_{\perp}^2 \Phi_u = -\frac{1}{\epsilon_0} \rho_{\perp}(x, y).$$

That says that an approximate solution to the Poisson equation is obtained by solving for Φ_u in the transverse space, using the transverse charge density, and then multiply the result by a constant longitudinal Φ_{\parallel} , or

$$\Phi^{(0)}(x, y, z) = \Phi_{\parallel} \Phi_u(x, y).$$

For a better solution, let's use a perturbative method. With ϕ a small longitudinal potential, write

$$\Phi_z^{(1)}(z) = \Phi_{\parallel} + \phi(z).$$

and insert this into equation (10)

$$\left(\Phi_{\parallel} + \phi(z)\right) \nabla_{\perp}^2 \Phi_u + \Phi_u \frac{\partial^2 \phi(z)}{\partial z^2} = -\frac{1}{\epsilon_0} \rho_z \rho_u,$$

to find, after cancelling out the lowest order terms

$$\frac{\partial^2 \phi(z)}{\partial z^2} + \omega^2 \phi(z) = 0, \quad \text{with } \omega^2 = -\frac{\rho_u}{\epsilon_0 \Phi_u}. \quad (11)$$

The complete solution is

$$\Phi_z^{(1)}(z) = \Phi_{\parallel} + \frac{1}{\omega} \frac{\partial \Phi_z}{\partial z}(0) \cos(\omega z), \quad \text{with } \Phi_z(z=0) = \Phi_{\parallel}, \quad (12)$$

Note that (i) the frequency ω in equation (11) is a (weak) function of (x, y) . e.g., for a Gaussian shaped beam, the transverse charge density and transverse potential have similar shape. (ii) The derivative of the longitudinal potential in the center of the segment is \approx proportional to the longitudinal variation of current in the beam at that location -remember that at the lowest order the longitudinal potential is equal to the longitudinal charge density- This has nothing to do with the transverse size of the beam but with its longitudinal phase space profile. (iii) The previous observation is consistent with the impedance models, where the longitudinal space charge kick on the particles is proportional to the charge per unit length in the beam.

Acknowledgments

The Authors acknowledge the continuous support of the Center for Data Intensive Computing at Brookhaven and thank Prof I. Hoffman and Dr. G. Franchetti for their hospitality at the GSI-Darmstadt Laboratory and for important discussions.

References

- [1] J.D.GALAMBOS, J.A.HOLMES, D.K.OLSEN A.LUCCIO and J.BEEBE-WANG: *Orbit User's Manual Vers. 1.10*. Technical Report SNS/ORNL/AP 011, Rev.1, 1999.
- [2] W.GROPP, E.LUSK, A.SKJELLUM: *Using MPI*. MIT Press, 1998.
- [3] H.GROTE and F.CH.ISELIN: *The MAD program, Vers.8.19*. Technical Report CERN/SL/90-13, Geneva, Switzerland, 1996.
- [4] B.FRANCZAK: *SIS Parameter List*. Technical Report GSI-SIS-TN/87-13, sept 1987.
- [5] W.HAGER: *Applied Numerical Linear Algebra*. Prentice Hall, 1988.