BEAM DYNAMICS MODELING BY SUB-THREE-DIMENSIONAL PARTICLE-IN-CELL CODE

L.G.Vorobiev and R.C.York

1 Introduction

In general, three-dimensional (3D) particle-in-cell (PIC) simulations of high-current charged particle beams are time consuming and have large memory demands. E.G., typically the number of macro-particles is $N_p \propto 10^6$ with 3D spatial grid dimensions of $N_{x,y,z} \propto 10^2$. These magnitudes are necessary primarily to reduce computational errors in the self-electric field calculation and to accurately describe beam line conducting boundaries.

For arbitrary beam distributions and boundaries, the charge densities and fields are evaluated on 3D grids during step-by-step integration of the trajectories [1]. Computational economy may be achieved by upgrading structural elements such as charge density redistribution block, space charge field solvers, trajectory integrators, and the calculational flow with parallel processing [2].

Another way to reduce 3D PIC computational demands is to develop a new formulation. A sub-3D PIC code, introduced in [3] provides an alternative to the general 3D PIC formulation when beam distributions and conducting boundaries satisfy certain assumptions [4,5]. This formulation allows the simulation of 3D beam dynamics in times comparable to that for two-dimensional (2D) PIC models. Though the self-consistency of the sub-3D model is approximate, preliminary studies have shown that it provides an adequate quantitative description for many beam configurations and boundary geometries [6,7].

2 General sub-3D PIC scheme

The sub-3D PIC code procedure has the same structural blocks as other conventional PIC algorithms: (i) generation of beam initial distribution, (ii) integration of macro-particle trajectories, (iii) charge density re-distribution, and (iv) self-electric field solvers. Steps (ii)-(iv) are performed repeatedly during the step-by-step calculations. Given below is an overview of the sub-3D PIC code with an emphasis on its new features.

2.1 Initial distribution for 3D beam

The 3D beam initial distribution algorithm is similar to those usually used for 3D PIC codes with the phase distribution (x, x', y, y', z, z') generated to match the desirable rms phase volume, the rms- and maximal sizes of the beam.

2.2 Integrator of macro-particles trajectories

1) A 3D trajectories integrator for the case $\Delta p/p=0$

The equations of motion of a charged particle with coordinates $\mathbf{x}=(x,y,z)$ may be derived from the Hamiltonian retaining the first or higher-order transmission characteristics of the beam line. In normalized form, the equations of motion can be written as:

$$\mathbf{x}'' = \mathbf{F}(\mathbf{x}, \mathbf{x}', s) \qquad (1)$$

where s denotes the distance along the particle trajectory, $\mathbf{F} = \mathbf{F}^{\mathbf{ext}} + \mathbf{F}^{\mathbf{sc}}$ are the external focusing and the space charge forces, and $\mathbf{x}' = d\mathbf{x}/ds$. Note, the space charge force, $\mathbf{F}^{\mathrm{sc}}$, is from the ensemble of N_p beam particles.

Integration schemes such as leapfrog retain accuracy to 2nd order. (For details see Chapter 4 of [1]). The memory requirements are modest, but small integration steps are necessary to reduce round-off error, making them computationally slow. A faster higher-order integrator for Eq. (1) is obtained by a modification of the balance method [8].

Omitting for sake of brevity the indices “*x*” in *F* and noting that $x'(s)-x'(s_k)=\int_{s_k}^{s}F(x(\tau),\tau)d\tau$, we obtain after simple algebra [2]:

$$x_{k+1}=(1+H_k)x_k-H_kx_{k-1}+J_2-H_kJ_1 \qquad (2)$$

where $x_k{=}x(s_k)$, $x_{k+1}{=}x(s_{k+1})$, $x_{k-1}{=}x(s_{k-1})$, $H_k{=}h_{k+1}/h_k$ and

$$J_1=\int_{s_{k-1}}^{s_k}\int_{s_k}^{s}[F(x(\tau),\tau)d\tau]ds=\int_{s_{k-1}}^{s_k}(s_{k-1}-s)F(x(s),s)ds$$

$$J_2=\int_{s_k}^{s_{k+1}}\int_{s_k}^{s}[F(x(\tau),\tau)d\tau]ds=\int_{s_k}^{s_{k+1}}(s_{k+1}-s)F(x(s),s)ds \qquad (3)$$

The equations are similar for the *y* and *z* coordinates. Approximations for the integrals $J_{1,2}$, expressed via $F(s_k)$, $F(s_{k-1})$ and $F(s_{k+1})$, are then substituted into (2), to achieve the integration scheme. An approximation of Eq. (2) may be written as:

$$x_{k+1}=(1+H_k)x_k-H_kx_{k-1}+\sum_{i=k-1}^{k+1}\Lambda_iF_i \qquad (4)$$

with the coefficients:

$$\Lambda_{i+1}=\frac{h_k^3}{24\mathrm{h}_k}(H_k^3+2H_k^2-1)$$

$$\Lambda_{i-1}=\frac{h_{k+1}^3}{24\mathrm{h}_k}(-H_k+1/H_k^2+2/H_k) \qquad (5)$$

$$\Lambda_i=\frac{h_kh_{k+1}}{12}(H_k^2+4H_k+4+1/H_k)$$

where $\mathrm{h}=(h_k+h_{k+1})/2$. The coefficients Λ_i are found via evaluation of the integrals $J_{1,2}$ from (3). The overall accuracy of the integration as implemented is $\psi=O(h^4)$. Fig. 1 illustrates the principle.

This approach provides a straightforward way to obtain other numerical multi-step integrators of higher-order. By increasing the number of coordinates $\boldsymbol{x}_k$ and functions $\boldsymbol{F}_k$ in the computational scheme, the accuracy can be increased. In this discussion, $\boldsymbol{x}_{k-1}$, $\boldsymbol{x}_k$, $\boldsymbol{x}_{k+1}$ and $\boldsymbol{F}_{k-1}$, $\boldsymbol{F}_k$, $\boldsymbol{F}_{k+1}$ are used. The computational algorithm (4) is implicit, because the unknown coordinates $\boldsymbol{x}_{k+1}$ are included in $\boldsymbol{F}_{k+1}$ on the right-hand side of Eq. (4). Therefore, at each step of integration the approximation $\boldsymbol{x}_{k+1}$ is found iteratively.

For cases where a fringe field description is used for external focusing, **F** will be smooth (trajectories ***x*** and space charge forces are always smooth) and the integrator (4) guarantees accuracy ψ to 4th order.

When a hard edge model is used for the external focusing, there is a discontinuity in the derivatives of **F**, but the algorithm maintains the same accuracy [2] in calculations of $J_{1,2}$. We introduce an external focusing function $F_x^{ext}(s)=G^{\pm}\varphi(x(s),y(s),z(s))$, where $\varphi(s)$ is smooth, but the focusing strength has a cutoff at the point s_k: $F_x^{ext}(s)=G^{-}\varphi(s)$ for $\mathrm{s}<\mathrm{s}_\mathrm{k}$ and $F_x^{ext}(s)=G^{+}\varphi(s)$ and $\mathrm{s}\geq\mathrm{s}_\mathrm{k}$. In Fig. 1 a trajectory is plotted for a particle that passes a hard edge focusing element with strength $G^{\pm}$. Since the function $\varphi(s)$ is smooth, the integrals $J_{1,2}$ can be evaluated with the same accuracy $\psi=O(h^4)$, and the integrator accommodates these functions by

$$x_{k+1}=(1+H_k)x_k-H_kx_{k-1}+\sum_{i=k-1}^{k+1}\hat{\Lambda}_i\varphi_i \qquad (4)'$$

with the coefficients in (5), modified to be:

$$\hat{\Lambda}_{i+1} = \frac{h_k^3}{24\mathrm{h}_k}\left[(H_k^3 + 2H_k^2)G^+ - G^-)\right]$$

$$\hat{\Lambda}_{i-1} = \frac{h_{k+1}^3}{24\mathrm{h}_k}\left[-G^+ H_k + G^- / H_k^2 + 2G^- / H_k\right] \qquad (5)'$$

$$\hat{\Lambda}_i = \frac{h_k h_{k+1}}{12}\left[(H_k^2 + 4H_k)G^+ + (4 + 1/H_k)G^-\right]$$

This integration technique was tested by 2D PIC simulations giving the same results as those from analytical models (KV-beams) and other 2D PIC codes employing leapfrog trajectory integrators. A merit of the scheme (4)-(5), (4)′-(5)′ is the possibility to have a much larger integration step in comparison with e.g. leapfrog, significantly reducing computational times. A drawback of the method (4) is the need to store arrays $\{\boldsymbol{x}_i\}$, $\{\boldsymbol{F}_i\}$, for indices $i=k-1,k,k+1$.

2) A 3D trajectories integrator for the case $\Delta p/p \neq 0$

If the functions $\boldsymbol{F}$ are smooth then $\boldsymbol{F}'$ is continuous, and the inclusion of $\Delta p/p \neq 0$ requires only minor modifications of the algorithm (4) taking into account the different advances h_k^n of each particle ($n = 1,\ldots,N_p$). Complications arise for the hard edge external focusing model or relatively short fringing fields. For beams with momentum spread, particles will pass edges of the external focusing element at different times. To make the integration valid for particles of different velocities a modification is required.

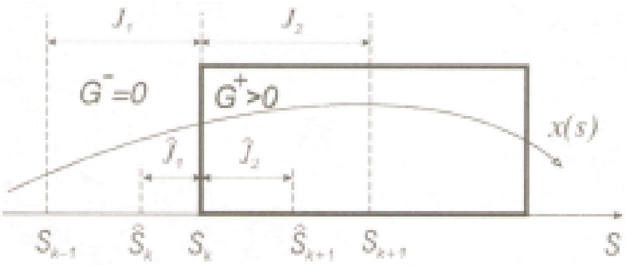

Figure 1: Particle trajectories integrator (schematic).

For example, when a reference particle with coordinate S_k arrives at the entrance edge of a focusing element, a faster particle will be inside at coordinate $\hat{S}_{k+1}$ while a slower particle will be at coordinate $\hat{S}_k$ outside the focusing element. See Fig. 1. Therefore the integrals $J_{1,2}$: $J_1 = G^- \int_{s_{k-1}}^{s_k} (s_{k-1} - s)\varphi(s)ds$, $J_2 = G^+ \int_{s_k}^{s_{k+1}} (s_{k-1} - s)\varphi(s)ds$ in (2) should be modified, taking into account the corrections: $\hat{J}_1 = (G^+ - G^-)\int_{\hat{s}_k}^{s_k} (s_{k-1} - s)\varphi(s)ds$ and $\hat{J}_2 = (G^- - G^+)\int_{\hat{s}_k}^{s_{k+1}} (s_{k-1} - s)\varphi(s)ds$. Direct calculation gives the modified coefficients Λ_i.

2.3 Slice algorithms and sub-3D field solver

A numerical technique to calculate the potential of a beam bunch by superposition of the potentials of thin charged disks (slices) was introduced in [4-6]. See Fig. 2. The discrete representation of the beam allows the computation of potentials of arbitrary bunch profiles to be fast and accurate. One of the important applications of the slice algorithm is the sub-3D Poisson Solver, which finds the solution of the 3D Poisson equation as a sequence of 2D solutions of the 2D equation.

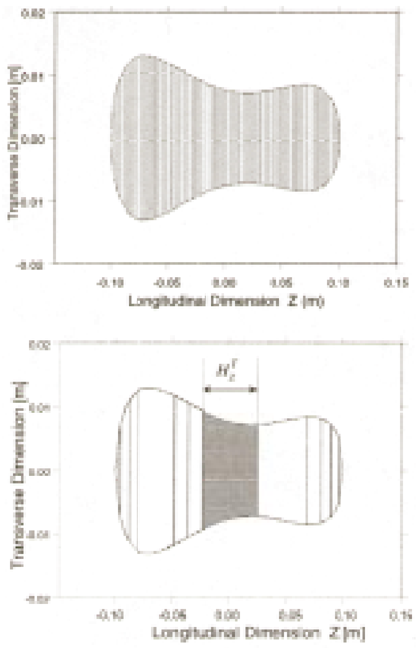

Figure 2: The illustration of the slice algorithms (left) and the sub-3D field solver (right). See [3] for details.

The space charge potential $u(\mathbf{x})$ from the charge density $\rho(\mathbf{x})$, distributed in the 3D region $\Re$, may be found from the standard 3D Poisson equation:

$$\begin{aligned} &\Delta u(\mathbf{x}) = -4\pi\rho\,(\mathbf{x}), \qquad \mathbf{x} = (x,y,z) \in \Re \\ &u(\mathbf{x})\big|_{\partial\Re} = U_0(\mathbf{x}) \end{aligned} \qquad (5)$$

For electrostatic cases, we assume $U_0 \equiv 0$ on the boundary $\partial\Re$. Splitting the Laplace operator as $\Delta_2 u = -4\pi\rho - \partial^2 u/\partial z^2$ where the notations $\Delta_2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$ or $\Delta_2 = \frac{1}{r}\frac{\partial}{\partial r}\left(\frac{\partial}{\partial r}\right) + \frac{1}{r^2}\frac{\partial^2}{\partial \varphi^2}$ are used for Cartesian and cylindrical coordinates correspondingly, after introducing the "corrected density" $\rho_{corr}(x,y,z) = \rho + \partial^2 u/\partial z^2/4\pi$, we re-write the original Poisson equation (1) as:

$$\begin{aligned} &\Delta u(\mathbf{x}) = -4\pi\rho_{corr}\,(\mathbf{x}) \\ &u(\mathbf{x}) = 0 \end{aligned} \qquad (6)$$

This reduces the original 3D problem (5) to a series of 2D problems (6). The benefit of such a representation comes a reduction in the number of 2D problems from the hundreds necessary for standard field solvers to ~5-10. The essence of this sub-3D Poisson Solver is that the driving term $\partial^2 u/\partial z^2$, is derived from: $\partial^2 u/\partial z^2 = -\partial E_z/\partial z$, with the fields $E_z = -\partial u/\partial z$ borrowed from the slice algorithm technique. (See the references [5,6] for details.) The corrected density is:

$$\rho_{corr}(x,y,z) = \rho + \frac{1}{4\pi}\frac{\partial^2 u}{\partial z^2} = \frac{\rho_{2D}}{H_z} - \frac{1}{4\pi}\frac{\partial E_z}{\partial z} \qquad (7)$$

The approximate self-consistency of such a technique is appropriate when the chamber boundaries have a simple geometry and when the beam is centered and has elliptical symmetry. The increased computational speed of the described sub-3D filed solver is due to the reduced spatial grid dimensions and fewer macro-particles (see [3]).

In the implementation of (6)-(7), the 2D charge density, ρ_{2D}, employs the standard cloud-in-cell (CIC) technique and the standard 2D Poisson Solver [1].

3 Discussion and conclusions

An initial version of the sub-3D PIC code with computational elements as described has been completed. The computational components were verified separately and compared to results from an existing 2D PIC code. To facilitate further testing and to provide more general simulation capability where required, the computer code provides both the sub-3D as well as fully 3D algorithms. The sub-3D will provide computational speed within simple boundaries, whereas the 3D will accommodate more general situations at the expense of speed. Because both the sub-3D and 3D routines share common data and execute the same procedures, the program size is not increased significantly. The approaches differ only in the method of the charge density and space charge field calculations.

The initial version of the sub-3D code uses a simplified model of the longitudinal field. Due to the relative lack of sensitivity of the longitudinal electric fields to the details of transverse charge density distribution, the use of template potentials for a specific slice charge density distribution does not cause significant errors [6,7]. As a consequence, only a moderate amount of pre-calculated data is required. In the future, a more general approach will be implemented.

This work was supported by the U.S. Department of Energy under Contract No. DE-FG02-99ER41118

References

1. R.W. Hockney and J.W. Eastwood, “Computer Simulation Using Particles”, McGraw-Hill Inc., 1981.
2. L.G. Vorobiev, Institute of Theoretical and Experimental Physics Report 22-94, Moscow, 1994 .
3. L.G. Vorobiev and R.C. York, Phys Rev. ST Accel. Beams **3**, 114201 (2000).
4. L.G. Vorobiev and R.C. York, Michigan State University Report MSUCL-1117, East Lansing, 1998.
5. L.G. Vorobiev and R.C. York, PAC’99, New York, March 1999, p. 2781.
6. L.G. Vorobiev and R.C. York, Michigan State University Report MSUCL-1191, East Lansing, 2001.
7. L.G. Vorobiev and R.C. York, “Method of Template Potentials to Find Space Charge Forces for High-Current Beam Dynamics Simulation”, these Proceedings.
8. A.A. Samarskij, “Theory of Finite-Differences Schemes”, Science Press, Moscow, 1982. In Russian.