Transverse space charge effect calculation in the Synergia accelerator modeling toolkit

Konstantin Okonechnikov, James Amundson, Alexandru Macridin
Fermilab

Abstract

This paper describes a transverse space charge effect calculation algorithm, developed in the context of accelerator modeling toolkit Synergia [1]. The introduction to the space charge problem and the Synergia modeling toolkit short description are given. The developed algorithm is explained and the implementation is described in detail. As a result of this work a new space charge solver was developed and integrated into the Synergia toolkit. The solver showed correct results in comparison to existing Synergia solvers and delivered better performance in the regime where it is applicable.

1. Introduction

Synergia is a modern particle accelerator modeling toolkit with many different capabilities, such as non-linear optics, space charge, resistive wall impedance, etc. It is a hybrid application based on existing software (CHEF libraries) and newly developed packages (e.g., the Sphyraena package for solving the Poisson equation). Synergia provides a flexible lattice description interface and is optimized for parallel execution (up to 2048 processors). It also has integration with different visualization tools (e.g. Matplotlib, VisIt.)
In most accelerators, the interaction between the beam particles and the machine is dominant, though in some situations the particle-particle interaction must also also be considered. Using the split-operator technique we can split the calculations for these interactions into single-particle and particle-particle pieces [1]. The space charge effect can then be calculated independently.
In order to determine the space charge kick one has to calculate the electric field of the particles in the rest frame of the bunch. This can be achieved by solving the well-known Poisson equation:

$$\nabla^2 \cdot \phi = \frac{\rho}{\epsilon},$$

where φ is the electric potential, ρ is the charge density, and ε is the permittivity of the material.

The Synergia toolkit has a module called Sphyraena, which solves the Poisson equation. The module includes solvers with different solution approaches, boundary conditions etc. The state-of-the-art solvers are:

- 3d Open boundary conditions (Green's function and Fast Fourier Transform method)
- 2d Open boundary conditions (analytic Gaussian approximation method)
- 3d Closed cylindrical boundary conditions (mixed spectral and finite-difference method).

The goal of this work was to develop a new transverse (2d) space charge solver. There were several reasons to implement this solver:

1) Performance
   The 2d approximation is much faster than the full 3d approach. Indeed, introducing a new dimension to a 2d calculation requires more computational power and more particles to obtain an accurate solution. In many situations the full 3d approach is not necessary and the approximate results of 2d solver are sufficient.
2) Non-Gaussian bunches

Operated by Fermi Research Alliance, LLC under Contract No. DE-AC02-07CH11359 with the United States Department of Energy.

The existing 2d solver (analytical Gauss solver) in Synergia assumes that the transverse charge distribution of particles is Gaussian. This assumption is usually only an approximation. (For example, the proposed Mu2e experiment will have a highly non-Gaussian beam during the resonant extraction process. [4])

2. Transverse space charge effect

Let us briefly discuss the idea behind the transverse space charge effect calculation. More detailed discussion of this calculation can be found in other papers ([2], [3].)
In 2d space it is more convenient to solve the electric field equation directly instead of solving the Poisson equation and differentiating to obtain the electric field. The field equation is

$$\frac{\partial E}{\partial x}+\frac{\partial E}{\partial y}+\frac{\partial E}{\partial z}=4\cdot\pi\cdot\rho(x,y,z),$$

where E is the electric field and ρ is the charge density.

The main assumption which is made in the transverse space charge approximation is that the electric field has a small variation in the z direction compared to the the field in the transverse coordinates. The field equation in the transverse approximation is

$$\frac{\partial E}{\partial x}+\frac{\partial E}{\partial y}=4\cdot\pi\cdot\lambda\cdot\sigma(x,y),$$
$$\sigma=\int dz\cdot\rho(x,y,z)$$

where ρ is the 3d charge density, σ is the 2d charge density and λ is the linear charge density.

Here we took λ constant. However, the charge distribution along the z direction does not necessarily have to be exactly uniform. One can make a better approximation by taking a slowly-varying longitudinal charge distribution into account by introducing local charge density as weight factor for particular region along z axis. This work is in progress.

3. Algorithm implementation

The algorithm used in this work is very similar to one implemented in the ORBIT simulation program [3]. The main idea of the method is to represent the electric field as convolution of the charge density and Green's function:

$$E=\int\rho(x)\cdot G(x-x')\approx\sum_{i,j}\rho_i\cdot G_{i-j},$$

where ρ is the charge density and G is the solution of field equation for a single charge source.

The Fourier Transform is then applied to find the solution:

$$\hat{F}(E)=\hat{\rho}\cdot\hat{G}$$
$$E(x,y)=\hat{F}^{-1}(\hat{F}(E)),$$

where F “hat” denotes Fourier transform. Fourier Transform was calculated using Fast Fourier Transform algorithms [6].

The use of the FFT requires the Green's function to be periodic, so one has to extend the grid size at

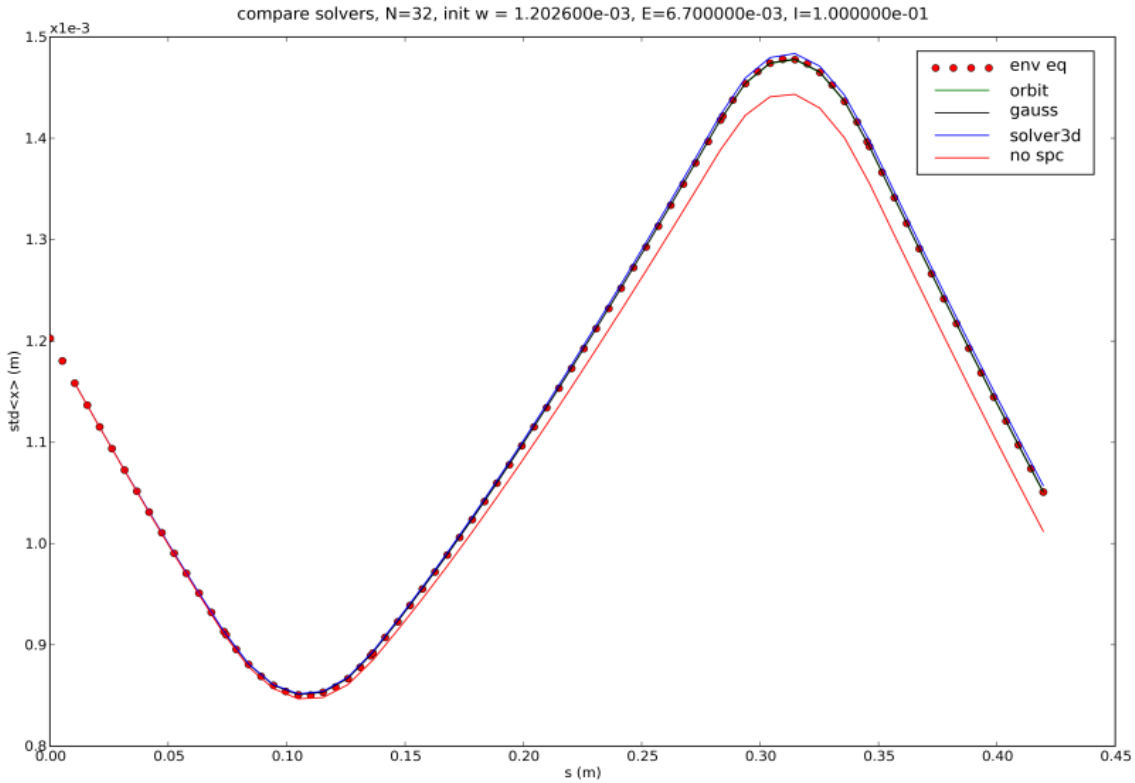
least twice in each dimension and spread the Green's function over the grid in order to use this method. A detailed explanation can be found in Ref. [5].

The algorithm includes several steps:

1) ***Define the grid*** At this step we determine the size of the grid in transverse space and allocate data structures required to store information about grid points. The location and size of the grid is determined by the parameters of the bunch, i.e. by the mean and standard deviation of the bunch coordinates.
2) ***Distribute charge on grid*** the Particle-In-Cell algorithm is used for this purpose. One important detail is that we do not take into account the particles that are outside of the grid when distributing the charge.
3) ***Calculate electric field*** This includes:
   - calculating the FFT of the charge density and Green's function
   - convolution as multiplication in the Fourier transformed space
   - calculating the inverse FFT of the convolution
4) ***Interpolate forces on the grid for the particles***
5) ***Apply the space charge kick on every particle***

4. Results

The algorithm has been implemented and integrated into the Synergia toolkit. It was tested for correctness by comparing with other result from other solvers (3d FFT solver, Gauss solver, etc.) For testing purposes a simple FODO cell was used. In the testing script the bunch was propagated through the cell for a fixed number of kicks (either 20 or 40).



Test simulation example. The standard deviation of bunch x coordinate (FODO cell propagation, 40 kicks)

The script was executed with different solvers and then the electric field and standard deviations of x and x' were compared. The tests were executed with different sets of initial parameters (number of particles, grid size, kinetic energy, current etc.) The derivative of the electric field was also analyzed. For a number of well-determined situations (as for example slight space charge effect assumption which has analytical solution called envelope equation) the results comparison showed discrepancy less than 5 percent for transverse solver and analytical solution.

The performance of the transverse space charge solver was also tested. The same testing script mentioned above was used for these purposes. As was expected, the transverse solver is much faster than the 3d solver. Some of the comparison results can be found in the table below.

Grid number*	Calculation time, Solver3d (s)	Calculation time , Transverse solver (s)
32	28.35	7.75
64	136.6	7.98
128	1319.78	8.89

Performance comparison, number of particles is 100000.

**Let N denote the grid number. Then the grid for the 3d solver will be [N,N,N], and the grid for the transverse 2d solver will be [N,N].*

5. Conclusions

The fast 2d approximation approach in solving space charge problem was investigated. The transverse space charge solver, based on this approach, was developed and integrated into the Synergia accelerator modeling toolkit. The new solver showed correct simulation results and delivered better performance (in the regime where it is applicable) in comparison to other solvers.

6. Further research

There are a number of important directions for further research. The most important one is the investigation of applicability limits of the solver. Others include the use of linear charge density factor analysis, solver parallelization and the inclusion of other boundary conditions.

7. References

1) J. Amundson, P. Spentzouris, J. Qiang and R. Ryne "Synergia: An accelerator modeling tool with 3-D space charge "
2) J. Holmes, J. Galambos, D. Jeon, D. Olsen, J. Cobb, "Dynamic Space Charge Calculations for High Intensity Beams in Rings", in Proceedings of the ICAP, 1998;
3) A. Shishlo, S. Cousineau, V. Danilov, J. Galambos, S. Henderson, J. Holmes, M. Plum, (Oak Ridge) "The ORBIT Simulation Code: Benchmarking and Applications" Oct 2006. 6pp. ICAP 2006, Chamonix Mont-Blanc, France, 2-6 Oct 2006, pp. 53-58.
4) http://mu2e.fnal.gov (Mu2e experiment official web page)
5) R. Hockney and J. Eastwood, Computer Simulation Using Particles (Adam Hilger, New York, 1988)
6) http://www.fftw.org/ (Fast Fourier Transform libraries)