3D SPACE CHARGE CALCULATIONS FOR BUNCHES IN THE TRACKING CODE ASTRA*

G. Pöplau†, U. van Rienen, Rostock University, Germany
K. Flöttmann, DESY, Hamburg, Germany

Abstract

Precise and fast 3D space charge calculations for bunches of charged particles are of growing importance in recent accelerator designs. One of the possible approaches is the particle-mesh method computing the potential of the bunch in the rest frame by means of Poisson's equation. Fast methods for solving Poisson's equation are the direct solution applying Fast Fourier Transformation (FFT) and a finite difference discretization combined with a multigrid method for solving the resulting linear system of equations. Both approaches have been implemented in the tracking code Astra. In this paper the properties of these two algorithms are discussed. Numerical examples will demonstrate the advantages and disadvantages of each method, respectively.

INTRODUCTION

The program package Astra (A space charge tracking algorithm) has been successfully used in the design of linac and rf photoinjector systems. The Astra suite originally developed by K. Flöttmann tracks macro particles through user defined external fields including the space charge field of the particle cloud [1].

The first version of Astra allowed the calculation of space charge fields of bunches with azimutal symmetry only. A further development was the implementation of a FFT based Poisson solver for full 3D space charge calculations with free space boundary conditions [2]. Recently a new set of 3D Poisson solvers has been implemented in Astra by G. Pöplau. These Poisson solvers are iterative algorithms, among them the state-of-the-art multigrid Poisson solver. The first version of the multigrid solver especially developed for space charge calculations on adaptive discretizations was introduced in [5]. Further developments can be found for instance in [6, 7].

In this paper the basic concepts of both the FFT and the iterative Poisson solvers are described. Advantages and disadvantages are discussed. Further the similarities and differences of the two approaches are demonstrated with numerical test examples.

3D SPACE CHARGE PARTICLE MESH ALGORITHMS

The particle-mesh (PM) method is a widely used algorithm to calculate space charge fields described for instance in [3]. It is assumed that the bunch is modelled by means of a distribution of macro particles. Generally, a rectangular box, in the following denoted as Ω, is constructed around the bunch. Further a Cartesian grid is defined inside the box and the values of the space charge density ρ are assigned at the grid points by a volume weighted distribution of the charge of the macro particles. Next, the potential φ is calculated by means of Poisson's equation given by

$$-\Delta\varphi = \frac{\rho}{\varepsilon_0} \qquad \text{in } \Omega \subset \mathbb{R}^3,$$

where ε_0 denotes the dielectric constant. The application of a Poisson solver provides the potential at the mesh points. Two different approaches are implemented as Poisson solvers in Astra. One is the wide spread FFT Poisson solver. As second approach iterative Poisson solvers have been implemented recently.

FFT Poisson Solver

The FFT Poisson solver is based on the construction of the solution of Poisson's equation by means of the discrete convolution

$$\varphi_{i,j,k} = \sum_{i',j',k'} G_{i-i',j-j',k-k'} \cdot \rho_{i',j',k'}, \qquad (1)$$

where $\varphi_{i,j,k}$ and $\rho_{i',j',k'}$ refer to the discrete values of φ and ρ, respectively. The Green's function is denoted by G. Applying the discrete Fourier transformation (DFT) yields

$$\hat{\varphi}_{l,m,n} = \hat{G}_{l,m,n}\hat{\rho}_{l,m,n} \qquad (2)$$

due to the convolution theorem. Here, the circumflex denotes the DFT and (l,m,n) the harmonic wave numbers. Finally, the potential at the grid points is obtained by an inverse DFT. Hence the Fourier approach can be considered as direct Poisson solver.

It is well-known that the DFT can be efficiently calculated by Fast Fourier Transformation (FFT) algorithms assuming that the number of discretization steps N is a power of 2. The numerical effort of the Fourier approach in the tree dimensional case is $O(M \log N)$ with the same number of steps in each coordinate direction and $M = N^3$ the total number of grid points.

Further, the DFT provides periodic functions and periodic boundary conditions are the natural boundary conditions for the FFT Poisson solver. In order to realize free space boundary conditions (also referred to as open boundary conditions) the Green's function is constructed such

* Work supported by DESY, Hamburg
† gisela.poeplau@uni-rostock.de

that the potential has the required decay of r^{-1} (see [3] for the detailed description).

While the FFT Poisson solver performs stable and very efficiently for many applications, it is restricted to equidistant meshes, that do not allow an adaptive discretization of more complicated particle distributions. Another drawback is that the FFT Poisson solver can be performed on a rectangular box only.

It has to be mentioned that only the Green's function is dicretized by the FFT Poisson solver but not the Laplacian (see equation (1)). That means in particular that the Fourier method would provide a very good approximation of φ if the particle distribution and with it ρ is smooth. In the next section the example of the discontinuous spherical particle distribution demonstrates the disadvantage of this behavior.

Iterative Poisson Solvers

Opposite to the FFT approach the Laplacian is now discretized. Second order finite differences (FD) are a common discretization technique. It provides a linear system of equations

$$Au = f, \qquad (3)$$

where u denotes the vector of the unknown values of the potential and f the vector of the given space charge density at the grid points. Since the matrix A is sparse, iterative solvers can be applied efficiently. In Astra, three different iterative solvers are implemented: multigrid (MG) and multigrid pre-conditioned conjugate gradients, a preconditioned conjugate gradient method (PCG) with Jacobi pre-conditioner, and (mainly for comparison reasons) the successive over relaxation (SOR).

The implementation of PCG and SOR is very simple but both algorithms suffer from the drawback that the number of iterations grows with $O(N^2)$. Nevertheless the PCG algorithm is still quite efficient on non-equidistant meshes [7]. Multigrid has optimal performance, i. e. the number if iteration steps to obtain a certain accuracy is independent of N, but the implementation is quite complicated.

Compared to the FFT Poisson solver the FD approach allows more flexibility. Thus non-equidistant adaptive discretizations are possible. Further a greater variety of boundary conditions can be realized. In Astra the following boundary conditions are implemented: free space boundary conditions, perfect electric conducting wall (Dirichlet boundary) on a rectangular box, perfect electric conducting beam pipe of elliptical shape (see [4] for more details).

Taking again $M = N^3$ as the total number of grid points the numerical effort for the multigrid method is $40M$ per iteration (with multigrid performed as V-cycle with two pre- and 2 post-smoothing steps, see [6] for explanation). With three iterations to achieve an accuracy of 10^{-2} (usually sufficient for space charge calculations) this sums up to a numerical effort of $120M$ arithmetical operations. Consequently the FFT solver can be faster than multigrid for small N ($N = 16, 32$) assuming an equidistant mesh and free space boundary conditions. Nevertheless the execution times of the tracking example in the next section show that multigrid can compete with FFT even on an equidistant mesh. Here, the solution of the previous time step is employed as initial guess for the iteration.

NUMERICAL TESTS

Spherical Bunch

In this subsection the previously discussed differences between the FFT and the iterative Poisson solvers are demonstrated with the example of a spherical particle distribution. The potential is calculated only once, i. e. no tracking considered. Assuming a uniform particle distribution the numerical results can be compared with the well-known analytical solution of a charged sphere.

The considered bunch contains 30,000 macro particles representing electrons, has a radius of 1 mm and a charge of -1 nC. The calculations were performed on an equidistant mesh with $N = 32$. Poisson's equation was solved with free space boundary conditions.

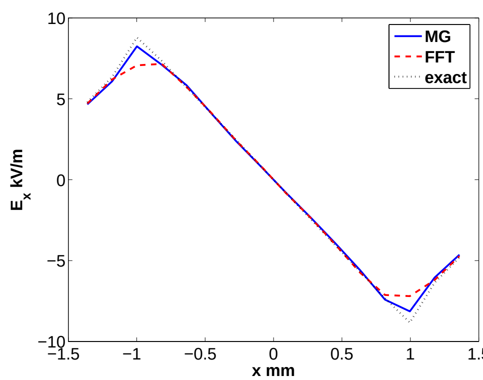

Figure 1: Transverse electric field: E_x along the x-axis.

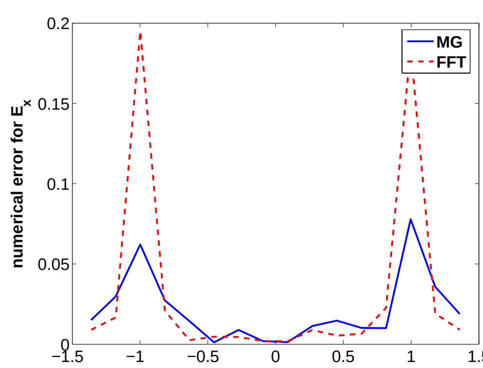

Figure 2: Error of the transverse electric field (E_x) along the x-axis.

Figure 1 shows the numerical results for the x-component of the electric field along the x-axis (i. e. $y = 0$, $z = 0$). There is good agreement of the values for both FFT and multigrid solver with the exact solution for $|x| < 1$. At the edges of the bunch the results of the multigrid solution coincide much better with the exact values. Based on an approximation by trigonometric polynomials the FFT Poisson solver smoothes the values at the hard edges of the bunch.

Tracking Example

As small tracking test a bunch of 10,000 macro particles representing electrons were chosen. The particle distribution is Gaussian with $\sigma_x = \sigma_y = 0.75$ mm and $\sigma_z = 1.0$ mm. Further the bunch has a total charge of -1 nC and an average energy of 2 MeV. It is tracked over a distance of 3 m. Additionally a quadrupole (length 0.2 m, gradient 0.1 T/m) is placed at $z = 1.2$ m. Figure 3 and 4 show

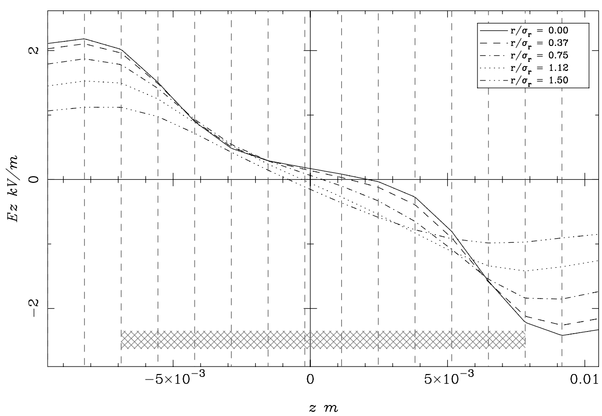

Figure 3: FFT Poisson solver: Longitudinal electric field component at the position $z = 3$ m.

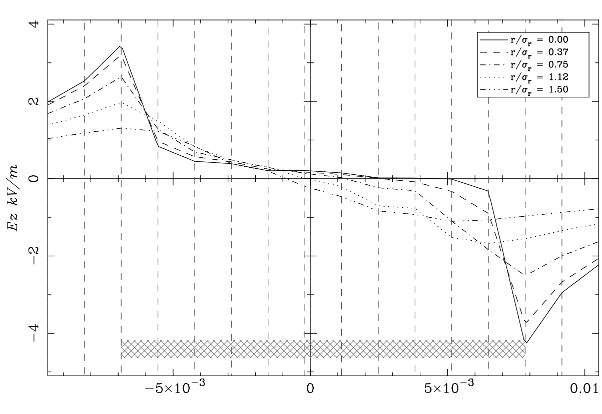

Figure 4: Multigrid Poisson solver: Longitudinal electric field component at the position $z = 3$ m.

the longitudinal component of the electric field at the position $z = 3$ m for the FFT Poisson solver and the multigrid Poisson solver, respectively. Both plots were taken with the *fieldplot* routine of Astra. While the values coincide quite good around the centre of the bunch they differ even more close to the edges.

In Table 1 the tracking times for the different Poisson solvers are collected. The number of steps N is here equal for each coordinate direction. While the FFT Poisson solver requires a power of 2 for N the iterative solvers can employ the advantage to choose N arbitrarily. The non-equidistant mesh is constructed as follows: around the bunch the mesh is equidistant and the same as for the FFT method (see the plots of the Figures 3 and 4), further the mesh is expanded with double step size.

Table 1: Tracking times for different Poisson solvers.

Poisson solver	**N = 28**	**N = 32**
FFT	–	237 s
MG, equidistant	202 s	229 s
MG, non-equidistant	219 s	271 s
PCG, equidistant	183 s	241 s
PCG, non-equidistant	188 s	247 s

CONCLUSIONS

Recently new iterative Poisson solvers have been implemented in the tracking code Astra for 3D space charge calculations. Numerical comparisons to the established FFT Poisson solver show good agreement of the results. The greater flexibility of iterative solvers is discussed and demonstrated with a small tracking example. Further development is required to employ the advantages of the iterative solvers, such as the adaptive discretization of complicated particle distributions.

REFERENCES

[1] K. Flöttmann, ”Astra”, DESY, Hamburg, `www.desy.de/~mpyflo`, 2000.

[2] K. Flöttmann, S.M. Lidia and P. Piot, ”Recent Improvements to the ASTRA Particle Tracking Code”, in Proceedings of PAC’03, Oregon, May 2003, pp. 3500–3502.

[3] R.W. Hockney and J.W. Eastwood, ”Computer Simulation Using Particles”, Institut of Physics Publishing, Bristol, 1992.

[4] A. Marovik̋, G. Pöplau, U. van Rienen and K. Flöttmann, ”Simulation of 3D Space Charge Fields of Bunches in a Beam Pipe”, Proceedings of EPAC’06, Edinburgh, June 2006.

[5] G. Pöplau, U. van Rienen, J. Staats and T. Weiland, ”Fast Algorithms for the Tracking of Electron Beams”, in Proceedings of EPAC’00, Vienna, June 2000, pp. 1387–1389.

[6] G. Pöplau, U. van Rienen, M.J. de Loos and S.B. van der Geer, ”A Fast 3D Multigrid Based Space-Charge Routine in the GPT Code”, Proceedings of EPAC’02, Paris, pp. 1658–1668.

[7] G. Pöplau, U. van Rienen, M.J. de Loos and S.B. van der Geer, ”Multigrid Algorithms for the Fast Calculation of Space-Charge Effects in Accelerator Design”, IEEE Transactions on Magnetics **40**, no. 2, 2004, pp. 117–120.