AN EFFICIENT 3D SPACE CHARGE ROUTINE WITH SELF-ADAPTIVE DISCRETIZATION *

G. Pöplau†, U. van Rienen, Rostock University, Rostock, Germany

Abstract

Precise and fast 3D space charge calculations for bunches of charged particles are still of growing importance in recent accelerator designs. A widespread approach is the particle-mesh method computing the potential of a bunch in the rest frame by means of Poisson's equation. Whereas an adaptive discretization of a bunch is often required for efficient space charge calculations in practice, such a technique is not implemented in many computer codes.

In this paper we present a new approach to an adaptive discretization which is based on the multigrid technique. The goal is that the error estimator needed for the adaptive distribution of mesh lines can be calculated directly from the multigrid procedure. The algorithm was implemented in the software package MOEVE and investigated for several particle distributions. It turns out that the adaptive discretization technique performs very efficiently.

INTRODUCTION

The simulation of the dynamics of high-brightness charged particle bunches demand the fast calculation of 3D non-linear space charge fields with an accuracy that matches the quality of the bunch. The particle-mesh method is a widespread model for space charge calculations. Here, adaptive discretization techniques are often required in order to satisfy both computational demands: accuracy and fast performance. Nevertheless, adaptive discretizations are implemented only in a few software packages together with space charge calculations. For instance, the FFT Poisson solver that is often applied allows only an equidistant mesh. An adaptive discretization following the particle density distribution is implemented in the GPT tracking code (General Particle Tracer, Pulsar Physics) together with a multigrid Poisson solver of the software package MOEVE (**M**ultigrid for n**o**n-**e**quidistant grids to sol**v**e Poisson's **e**quation) [5, 11]. The disadvantage of this approach is that it does not provide a hierarchical construction of meshes which could be used directly by the multigrid algorithm.

In this paper we present a new approach to an adaptive discretization which is based on the multigrid technique. The goal is that the error estimator needed for the adaptive distribution of mesh lines can be calculated directly from the multigrid procedure. The algorithm has been implemented within the framework of the software package MOEVE. It will be investigated for several particle distributions among them a particle distribution which occured during simulations for the European XFEL [1].

* Work supported by DFG under contract number RI 814/18-1

† gisela.poeplau@uni-rostock.de

3D SPACE CHARGE MODEL

The space charge model we consider here is the particle-mesh method. The distribution of particles in a bunch is modeled as distribution of macro particles. Assuming that the energy of the macro particles is within the same range the space charge field is calculated in the rest frame of the bunch. After the transformation into the rest frame a mesh is constructed around the particles of the bunch and the charge of the particles is assigned to the mesh points. Now, the potential φ can be obtained from Poisson's equation given by

$$\begin{aligned} -\Delta\varphi &= \frac{\varrho}{\varepsilon_0} && \text{in } \Omega \subset \mathbb{R}^3, \\ \varphi &= 0 && \text{on } \partial\Omega_1, \\ \frac{\partial\varphi}{\partial n} + \frac{1}{r}\varphi &= 0 && \text{on } \partial\Omega_2, \end{aligned} \tag{1}$$

where ϱ the space charge distribution, ε_0 the dielectric constant and r the distance between the centre of the bunch and the boundary. Usually, the domain Ω is a rectangular box constructed around the bunch. On the surface $\partial\Omega = \partial\Omega_1 \cup \partial\Omega_2$ ($\partial\Omega_1 \cap \partial\Omega_2 = \emptyset$) perfectly conducting boundaries ($\partial\Omega_1$) or open boundaries ($\partial\Omega_2$) can be applied. For space charge calculations within a beam pipe the domain Ω is assumed to be a cylinder with elliptical cross section. A detailed description of the 3D space charge model can be found in [8] and the model with elliptical shaped beam pipe in [4], respectively.

For the solution of the Poisson equation we applied the discretization with second order finite differences. This leads to a linear system of equations of the form

$$L_h u_h = f_h, \tag{2}$$

where u_h denotes the vector of the unknown values of the potential and f_h the vector of the given space charge density at the grid points. The step size h indicates a certain refinement level and the operator L_h is the discretization of the Laplacian.

THE POISSON SOLVERS OF MOEVE

The software package MOEVE has been developed for space charge calculations. It involves several iterative Poisson solvers among them the state-of-the-art multigrid Poisson solvers MG (multigrid) and MG-PCG (multigrid preconditioned conjugate gradients). These algorithms provide optimal convergence, i. e. the number of iteration steps

needed to obtain a certain accuracy is independent of the step size h.

The multigrid algorithm operates on a certain number of grids starting with the mesh given by the discretization of Poisson's equation which is related to equation (2). This mesh is referred to as the fine grid. Then a sequence of coarser grids is generated by cutting mesh lines. On an equidistant mesh where the number of mesh lines equals $N = 2^t + 1$, every second mesh line can be removed. Of course this process can be also performed vice versa, i. e. we start with a coarse mesh and refine the mesh (adaptively) until a certain fine level is achieved. More details of the Poisson solvers of MOEVE can be found elsewhere, for instance in [5, 6].

Since the bunch is located only in a certain part of Ω, an adaptive discretization strategy is required for an efficient solution method. Successfully applied for space charge calculations, one possible method is the adaptive distribution of mesh lines according to the distribution of the particles inside the bunch as implemented in the tracking code GPT [10]. However, the main drawback of this approach is that the mesh has no natural relation to a multigrid hierarchy. Together with the multigrid Poisson solver of MOEVE, a special coarsening strategy has been developed for these adaptive meshes in order to achieve optimal multigrid convergence [9].

SELF-ADAPTIVE MULTIGRID

The generation of adaptive meshes providing both an appropriate approximation of the bunch and a hierarchy of meshes for the multigrid Poisson solver is an important task for the development of efficient 3D space charge algorithms. The grid refinement should be self-adaptive, i. e. the grid refinement is carried out dynamically during the solution process. It is controlled by some refinement criterion.

A common criterion which will be tested in this paper is the τ-criterion [12]. Before this criterion can be defined some notations are necessary. The step sizes h and $2h$ refer to the step sizes on the fine and the next coarser grid (usually with double mesh size), respectively. The operators I_h^{2h} and $\widehat{I}_h^{2h}$ denote different restriction operators. For the numerical tests of the next section the injection was chosen for $\widehat{I}_h^{2h}$ and the full weighting restriction for I_h^{2h}. The τ-criterion is based on the so-called (h,$2h$) relative truncation error τ_h^{2h} with respect to the restriction operators I_h^{2h} and $\widehat{I}_h^{2h}$. It is defined by

$$\tau_h^{2h} := L_{2h}\widehat{I}_h^{2h}u_h - I_h^{2h}L_h u_h \,. \qquad (3)$$

The values of τ for the "XFEL bunch" (see section on numerical investigations below) are represented in Figure 1.

By means of the refinement criterion a hierarchy of locally refined grids can be generated. The self-adaptive multigrid scheme is given as follows:

Algorithm: Self-Adaptive Multigrid

1. Start on a relatively coarse mesh.
2. Perform a few multigrid cycles on equation (2).
3. Calculate τ_h^{2h}.
4. Add mesh lines locally, where $|\tau_h^{2h}| > \varepsilon$.
5. Proceed from 2) as long as $|\tau_h^{2h}| > \varepsilon$.

The algorithm is constructed such that the mesh is refined at locations where $|\tau_h^{2h}| > \varepsilon$. Hence, the structure of the resulting mesh corresponds to the tensor product mesh implemented in MOEVE. This strategy allows the direct application of the MOEVE MG Poisson solvers on the refined mesh. This algorithm was first given in [6].

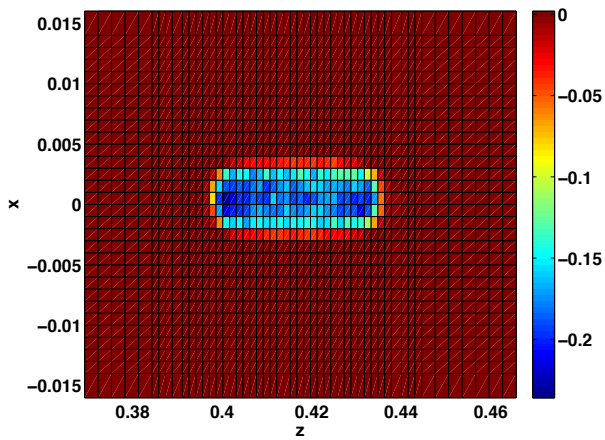

Figure 1: Values of τ for the XFEL bunch. The values are shown in the (x,z)-plane at x=0.0 m.

NUMERICAL INVESTIGATIONS

In this section the performance of the adaptive discretization strategy is investigated. We applied both MG (multigrid) and MG-PCG (multigrid preconditioned cojugate gradients) as multigrid Poisson solvers implemeted in MOEVE. Although MG-PCG takes more time for one single step because of additional caculations it is often more stable in real life applications. The results for the adaptively refined grid were compared to those obtained with an equivalent equidistant grid. All Poisson solvers were applied until the relative residual had achieved a value of less than 10^{-2}. This value seems to be rather low, but further iterations would not improve the numerical error, because the space charge densities of the following particle distributions are discontinues.

Cylindrical Particle Distribution

First, the refinement algorithm was applied to a cylindrical particle distribution with 100,000 macro-particles and

$N_x \times N_y \times N_z$	MG steps	CPU time
$17 \times 17 \times 33$	2	0.05 s
$33 \times 33 \times 65$	3	0.16 s
$65 \times 65 \times 129$	4	1.36 s
$129 \times 129 \times 257$	4	10.80 s
$257 \times 257 \times 513$	4	88.90 s
$N_x \times N_y \times N_z$	MG-PCG steps	CPU time
$17 \times 17 \times 33$	2	0.05 s
$33 \times 33 \times 65$	4	0.20 s
$65 \times 65 \times 129$	4	1.38 s
$129 \times 129 \times 257$	4	11.10 s
$257 \times 257 \times 513$	4	92.2 s

Table 1: Performance of MG and MG-PCG for a cylindrical particle distribution on equidistant grids.

$N_x \times N_y \times N_z$	MG steps	CPU time
$17 \times 17 \times 33$	2	0.04 s
$33 \times 33 \times 59$	3	0.20 s
$47 \times 47 \times 97$	4	0.87 s
$65 \times 65 \times 163$	7	4.36 s
$95 \times 95 \times 283$	18	41.10 s
$N_x \times N_y \times N_z$	MG-PCG steps	CPU time
$17 \times 17 \times 33$	2	0.04 s
$33 \times 33 \times 59$	4	0.23 s
$47 \times 47 \times 97$	4	0.93 s
$65 \times 65 \times 163$	6	4.01 s
$85 \times 85 \times 283$	8	21.80 s

Table 2: Performance of MG and MG-PCG for a cylindrical particle distribution on the adaptively constructed mesh.

a radius of $R = 2$ mm and a length of 17.3 mm. Such a cigar-like bunch shape was chosen because these long distributions usually pose a problem to the Poisson solvers [7]. The macro particles of the bunch were uniformly distributed. Th bunch had a total charge of $Q = -1$ nC. The computational domain Ω was constructed as $\Omega = [-0.0085, 0.0085] \times [-0.0085, 0.0085] \times [-0.017, 0.017]$ and open boundary conditions were applied. For the adaptive algorithm we started with an equidistant mesh i. e. the step size at the coarsest level is approximately equal for all coordinate directions. This is not necessary in general but for long bunches it improves the performance.

In the following the performance of the straigth-forward multigrid technique on equidistant grids was compared to the performance on the adaptive grid. The CPU times include all calculations for space charge i. e. also the assignment of the particle's charge to the grid points.

In Table 1 the CPU time is given for calculations with the MG and MG-PCG solver on equidistant grids. The results approve the linear behaviour of the MG and MG-PCG algorithms.

Table 2 represents the results for the self-adaptive multigrid procedure. The algorithm starts with a grid of $17 \times 17 \times 33$ mesh points. Then the mesh will be refined successively until all values of τ are less than 10^{-2}. The refinement of each level is given in Table 2. The smallest step size of each level coincides with the step size of the corresponding level of the equidistant discretization. Since the adaptive discretization requires only a fraction of the mesh points of the equidistant grid the performance time for MG-PCG reduces to a quarter of the time required for the same refinement level on an equidistant grid.

XFEL bunch

The particle distribution in this section was taken from simulations for the European XFEL [1]. The bunch had a total charge of -1 nC. At the cathode it was started with a radius of 3 mm and the laser was modeled as a flat top shape with 20 ps full width half maximum and a rise and fall time of 2 ps. The bunch was tracked through the rf gun with the tracking code ASTRA [3] with the settings given in the technical desing report [2].

$N_x \times N_y \times N_z$	MG steps	CPU time
$33 \times 33 \times 65$	3	0.15 s
$65 \times 65 \times 129$	3	1.4 s
$129 \times 129 \times 257$	3	3.19 s
$257 \times 257 \times 513$	2	51.20 s
$N_x \times N_y \times N_z$	MG-PCG steps	CPU time
$33 \times 33 \times 65$	4	0.12 s
$65 \times 65 \times 129$	4	1.34 s
$129 \times 129 \times 257$	4	10.70 s
$257 \times 257 \times 513$	2	53.00 s

Table 3: Performance of MG and MG-PCG for the simulated XFEL bunch at 0.07 m on an equidistant mesh.

For this numerical test the particle distribution at the longitudinal position of 0.07 m after the cathode was chosen. Here, the bunch had a radius of 3.0 mm, a length of 8.0 mm and an energy of 2.5 MeV. For the simulation the bunch contained 100,000 macro particles. The bounding box in the rest frame was constucted as $\Omega = [-0.016, 0.016] \times [-0.016, 0.016] \times [0.037, 0.047]$ and Dirichlet boundary conditions were applied.

Table 3 and 4 present the CPU time for the equidistant and for the adaptively constructed grids, respectively. It turns out that the computational effort can be reduced enormously by the application of the adaptive discretization strategy. Considering the finest dicretization the CPU time of the MG method was reduced nearly to a tenth on the adaptive mesh.

Figure 2 and 3 show the adaptive discretization with $45 \times 45 \times 85$ grid points.

CONCLUSIONS

In this paper we constructed an adaptive discretization technique for space charge calculations which is based on the τ-criterion. By means of this approach a hierarchy of grids can be generated which can be directly used by the multigrid scheme of the software package MOEVE. The

$N_x \times N_y \times N_z$	MG steps	CPU time
$31 \times 31 \times 53$	2	0.06 s
$45 \times 45 \times 85$	3	0.52 s
$59 \times 59 \times 141$	2	2.00 s
$83 \times 83 \times 243$	3	5.96 s
$N_x \times N_y \times N_z$	MG-PCG steps	CPU time
$31 \times 31 \times 53$	2	0.14 s
$45 \times 45 \times 85$	4	0.65 s
$59 \times 59 \times 141$	6	2.16 s
$83 \times 83 \times 243$	6	7.96 s

Table 4: Performance of MG and MG-PCG the XFEL bunch at 0.07 m on the adaptively constructed mesh.

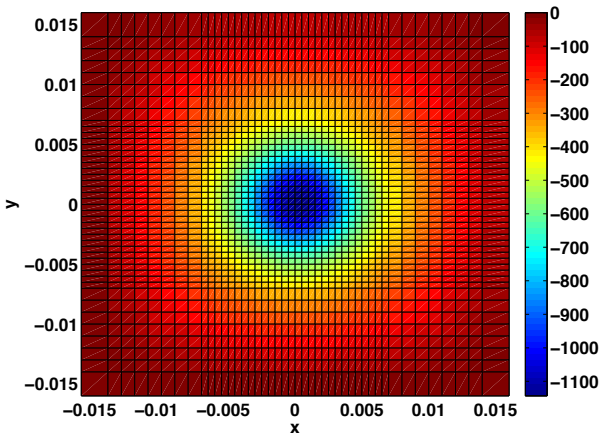

Figure 2: Adaptive discretization (transversal) for the XFEL bunch. The potential in the rest frame is plotted in the transversal plane through the longitudinal centre of the bunch.

implementation into the framework of MOEVE allowed a relatively simple approach. The numerical tests showed that the adaptive discretization reduced the number of mesh points enormously. Hence, the space charge calculations were performed very efficiently.

REFERENCES

[1] M. Clemens, M. Dohlus, S. Lange, G. Pöplau, T. Limberg, and U. van Rienen. Microbunch amplification in the european XFEL. TESLA-FEL-2009-02, DESY, Hamburg (`flash.desy.de/reports_publications/tesla_fel_reports/tesla_fel_2009/`), 2009.

[2] M. Altarelli et al., editor. *The European X-Ray Free-Electron Laser Technical design report*. DESY 2006-097, 2007.

[3] K. Flöttmann. *ASTRA*. DESY, Hamburg, `www.desy.de/~mpyflo`, 2000.

[4] A. Markoviḱ. Numerical computation of space-charge fields of electron bunches in a beam pipe of elliptical shape. TESLA-Report 2005-21, DESY, Hamburg (`tesla.desy.de/new_pages/TESLA/TTFnot05.html`), 2005.

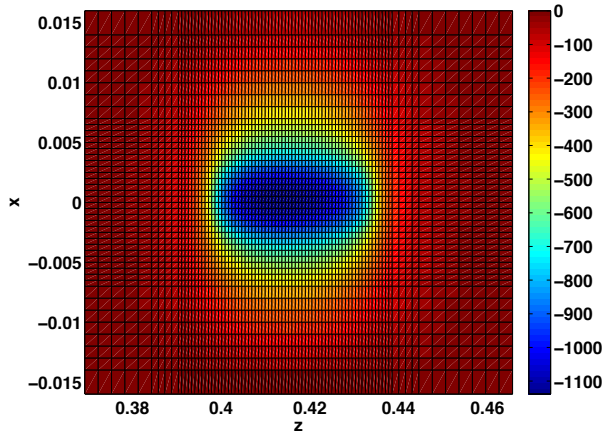

Figure 3: Adaptive discretization (longitudinal) for the XFEL bunch. The potential in the rest frame is plotted in the (x,z) plane at y=0.0 m.

[5] G. Pöplau. *MOEVE 2.0: Multigrid Poisson Solver for Non-Equidistant Tensor Product Meshes*. Universität Rostock, 2006.

[6] G. Pöplau and U. van Rienen. A self-adaptive multigrid technique for 3D space charge calculations. *IEEE Transactions on Magnetics*, 44(6):1242–1245, 2008.

[7] G. Pöplau, U. van Rienen, and K. Flöttmann. 3D space charge routines: The software package MOEVE and FFT Poisson solvers. In H. Fukuma, E.S. Kim, and K. Ohmi, editors, *Proceedings of the International Workshop on Electron-Cloud Effects (ECLOUD'07), Daegu, Korea, 9-12 April 2007*, KEK Proceedings 2007-10, pages 181–188, 2007.

[8] G. Pöplau, U. van Rienen, S.B. van der Geer, and M.J. de Loos. Fast calculation of space charge in beam line tracking by multigrid techniques. In W.H.A. Schilders, E.J.W. ter Marten, and S.H.M. Houben, editors, *Scientific Computing in Electrical Engineering*, number 4 in Mathematics in Industry, pages 329–336, Berlin, 2004. Springer-Verlag.

[9] G. Pöplau, U. van Rienen, S.B. van der Geer, and M.J. de Loos. Multigrid algorithms for the fast calculation of space-charge effects in accelerator design. *IEEE Transactions on Magnetics*, 40(2):714–717, 2004.

[10] G. Pöplau, U. van Rienen, S.B. van der Geer, and M.J. de Loos. A multigrid based 3D space-charge routine in the tracking code GPT. In M. Berz and K. Makino, editors, *Computational Accelerator Physics 2002 (Proceedings of the 7th International Computational Accelerator Physics Conference, East Lansing, Michigan, USA)*, number 175 in Institute of Physics Conference Series, pages 281–288, Bristol and Philadelphia, 2005. Institute of Physics Publishing.

[11] Pulsar Physics, Burghstraat 47, 5614 BC Eindhoven, The Netherlands, `www.pulsar.nl/gpt`. *General Particle Tracer (GPT)*, release 2.70 edition, 2004.

[12] U. Trottenberg, C. Oosterlee, and A. Schüller. *Multigrid*. Academic Press, San Diego, 2001.