Performance of Multigrid in the Context of Beam Dynamics Simulations

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

Abstract— Precise and fast 3D space charge calculations for bunches or clouds of charged particles are of growing importance in design studies for future linear accelerators and light sources. One of the possible approaches is the computation of the potential of the bunch in the rest frame by means of Poisson's equation. The software package MOEVE has been developed for space charge calculations on non-equidistant grids. It consists of several iterative Poisson solvers (MOEVE Poisson solvers), among them the state-of-the-art multigrid Poisson solver. Furthermore, the MOEVE Poisson solvers have been implemented in the tracking code Astra and GPT. In this paper the algorithms of the software package MOEVE will be described and the performance will be tested for very large linear systems of equations. The numerical results will show that the conditions for the Poisson solvers have to be chosen carefully in order to achieve optimal results for real life applications.

1. INTRODUCTION

The design of future light sources and colliders requires increasingly precise 3D beam dynamics simulations. In so-called tracking simulations the particle trajectory is determined which is described by the relativistic equation of motion. The equation of motion is solved by means of an appropriate time integration scheme. In regimes of rather low energy, this implies that the space charge fields have to be taken into account in each time step of the numerical integration. Recently, the efficient calculation of 3D space charge fields gained particular importance in the context of electron cloud studies for the ILC (International Linear Collider) damping rings.

Based on the geometric multigrid technique fast Poisson solvers have been developed and successfully applied for 3D space charge simulations. In theory, these multigrid Poisson solvers have optimal performance, i.e., the numerical effort depends linearly on the number of mesh points. Unfortunately, this optimal convergence rate can sometimes not be achieved in simulations of real life problems. In this paper the performance of the geometric multigrid technique is investigated in the context of space charge calculations, in particular, for huge numbers of mesh points, i.e., up to 4 million. Another problem is the behavior of the multigrid Poisson solver within the particle tracking procedure. Since space charge fields have to be computed in each time step of the numerical integration, the calculated fields of the previous time step can be used as initial guess for the iterative Poisson solver. With this approach the effort for the new space charge calculation can be reduced. The question under investigation is how multigrid can compete with other iterative algorithms for real life tracking simulations.

The iterative Poisson solvers are available as software package MOEVE 2.0 (MOEVE: **M**ultigrid for n**o**n-**e**quidistant grids to sol**v**e Poisson's **e**quation) [8]. Furthermore, these Poisson solvers are implemented in the tracking code Astra (DESY, Hamburg, Germany) [3] and the tracking code GPT (Pulsar Physics, Eindhoven, The Netherlands) [13].

2. MATHEMATICAL MODEL

Space charge calculations for beam dynamics studies are performed within a tracking procedure. The tracking is a method to determine the trajectories of the particles which are described by the relativistic equations of motion [1]. The equations of motion are solved by means of an appropriate time integration scheme. The space charge fields have to be taken into account in each time step of the numerical integration. The space charge calculations are performed in the rest frame of the bunch by means of Poisson's equation given by

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

where φ denotes the potential, ϱ the space charge distribution, ε_0 the dielectric constant and r the distance between the center 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 [11] and the model with elliptical shaped beam pipe in [7], respectively.

The discretization of Equation (1) with second order finite differences leads to a linear system of equations

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

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 discretization and the matrix A_h is the discretization of the Laplacian.

3. THE MOEVE POISSON SOLVERS

In the software package MOEVE, different iterative Poisson solvers are implemented for the solution of (2): multigrid (MG) and multigrid pre-conditioned conjugate gradients (MG-PCG); a pre-conditioned conjugate gradient method (PCG) with Jacobi pre-conditioner; (mainly for comparison reasons) the successive over relaxation (SOR); and biconjugated gradients (BiCG) and BiCGSTAB as a stabilized version of BiCG.

The implementation of the methods PCG, SOR, BiCG and BiCGSTAB is simple but these algorithms suffer from the drawback that the number of iterations grows with $\mathcal{O}(N^2)$. Here, N denotes the number of mesh lines in each coordinate direction. Descriptions of these methods can be found elsewhere, e.g., in [4].

As already mentioned above, multigrid has optimal performance. Consequently, it is applied as the state-of-the-art Poisson solver, but the implementation is very complex. We give here only a rough idea of our specific multigrid scheme. For more details see [2, 5].

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 or the fine level. 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. For space charge calculations, a special coarsening strategy for non-equidistant grids has been developed [12]. Here, the removal of mesh lines follows the rule: two neighboring steps h_1 and h_2 remain in the next coarser grid as long as either $h_1 \geq sh_{\min}$ or $h_2 \geq sh_{\min}$, where $h_{\min}$ denotes the overall minimal step size of the corresponding fine level. The factor s is chosen as $s = 1.6$ or $s = 1.7$ in order to obtain a decreasing aspect ratio of the mesh spacing.

Now the system of Equations (2) is solved iteratively in the following way: first a raw approximation of the solution of the system of equations is obtained by the application of a few steps of a relaxation scheme (e.g., Gauss-Seidel) which is called pre-smoothing. This approximation is then improved by a correction vector obtained on the coarser grids (the so-called coarse grid correction), where restriction and interpolation work as grid transfer operators. After applying interpolation another few steps of relaxation are necessary, these are the post-smoothing steps. For the space charge calculations a multigrid V-cycle is realized. This scheme goes strictly down from the fine to the coarsest grid and then up again to the fine level. The V-cycle is performed with the following components: 2 pre- and 2 post-smoothing steps with red-black Gauss-Seidel relaxation, full-weighting restriction and trilinear interpolation.

Furthermore, multigrid can be applied as a preconditioner for the conjugate gradient algorithm (MG-PCG). This method leads to a better convergence at least in cases where a plain multigrid scheme converges too slowly [6].

Detailed convergence studies for MG, MG-PCG, PCG and SOR in the context of space charge calculations can be found in [11, 12]. In the next section we investigate MG, MG-PCG and PCG for Dirichlet and open boundaries, in particular, for very large linear systems of equations.

4. PERFORMANCE OF THE MULTIGRID POISSON SOLVERS

The objective of this section is to investigate the performance of the multigrid algorithms of MOEVE for a huge number of grid points up to more than 4 millions. The algorithms MG, MG-PCG and PCG were performed for the Poisson Equation (1) on $\Omega = [-0.5, 0.5]^3$ with Dirichlet and

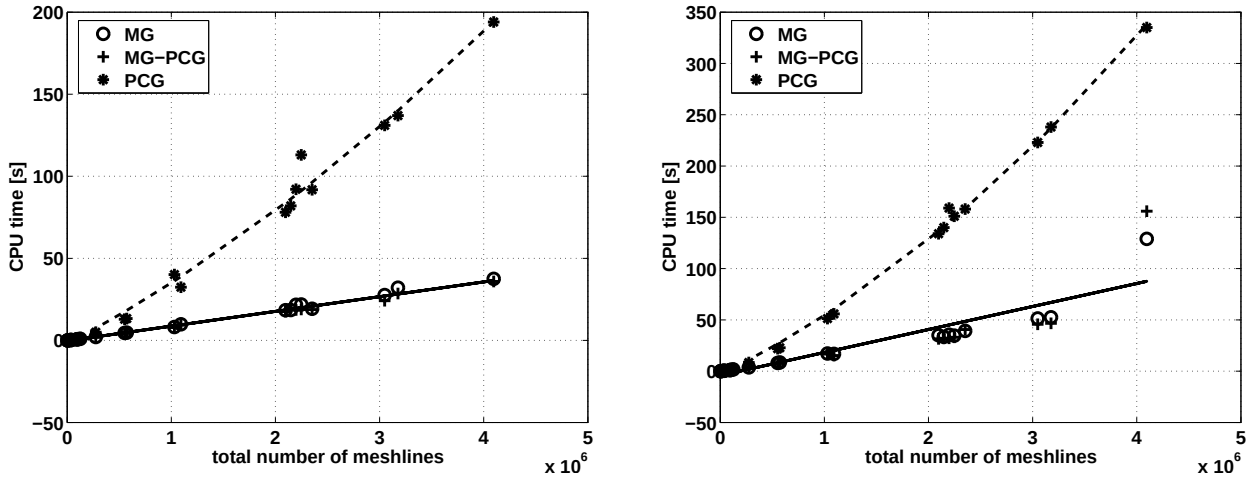

Figure 1: The performance of MG, MG-PCG and PCG for Poisson's equation with Dirichlet boundary conditions on equidistant grids (left) and on non-equidistant grids (right).

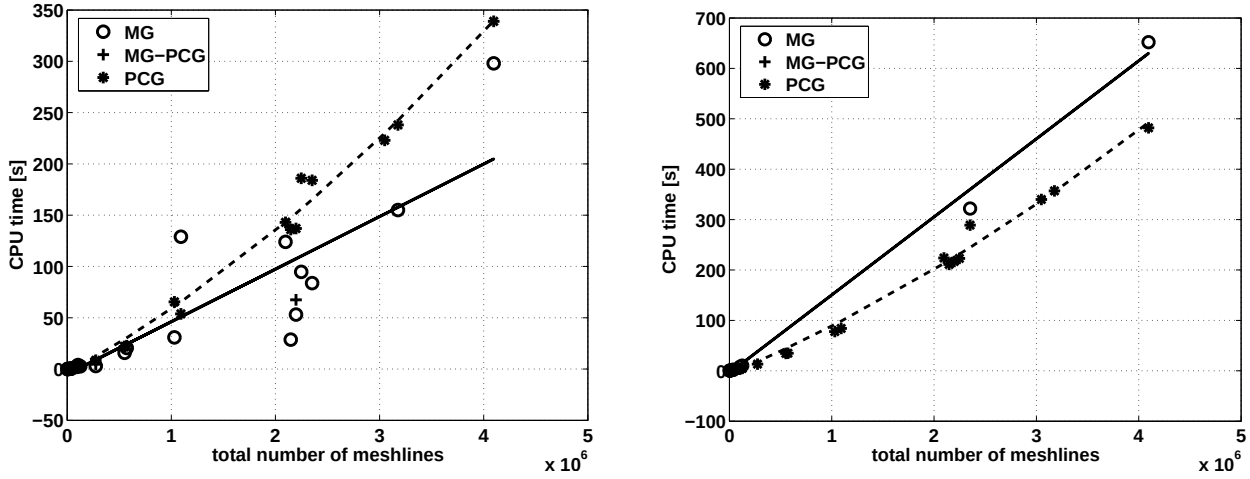

Figure 2: The performance of MG, MG-PCG and PCG for Poisson's equation with open boundary conditions on equidistant grids (left) and on non-equidistant grids (right).

open boundary conditions, respectively. The discretization was chosen equidistant as well as non-equidistant. The non-equidistant distribution of the mesh lines x_i in x-direction was given by the function

$$x_i = \frac{\sinh(6(-0.5 + i/N_x))}{2\sinh(3)}, \quad \text{for } i = 0, \ldots, N_x,$$

where N_x denotes the number of mesh lines in x-direction. The mesh lines in y- and z-direction are distributed in the same way.

A bunch of ellipsoidal shape was chosen as the model's problem, because numerical simulations of so-called pancake shaped bunches have gained in importance in recent years [10, 14]. For the numerical test the ellipsoidal bunch has the extensions $a = 0.2\,\text{mm}$, $b = 0.2\,\text{mm}$, $c = 0.1\,\text{mm}$, where a, b and c denote the length of the half axes in x-, y- and z-direction, respectively. The ellipsoid has a uniformly distributed charge of $Q = -1\,\text{nC}$.

The algorithms where performed until the maximum norm of the relative residual had reached a value less than 10^{-9}. In Figure 1 the results for Dirichlet boundary conditions are presented. MG and MG-PCG achieve optimal convergence on the equidistant grid (Figure 1, left), i.e., the convergence does not depend on the total number of mesh points N_p. The value of the maximum norm of the relative residual was after 9-10 V-cycles less than 10^{-9}. On the non-equidistant grid the optimal performance is not achieved for 4 million grid points (Figure 1, right), here the algorithms MG and MG-PCG require 25 and 24 V-cycles, respectively. The number of V-cycles for $N_p < 4 \cdot 10^6$ was between 10 and 14. Compared to that the numerical effort of the PCG algorithm

grows quadratically with the number of mesh points.

Open boundary conditions pose a problem because the matrix A_h in (2) becomes singular with increasing number of mesh points (see [7] for the structure of the matrix A_h). Figure 2 (left) shows that the behavior of the multigrid algorithms on equidistant meshes becomes unpredictable, if $N_p > 10^6$. The MG-PCG method does not converge, if $N_p > 2.5 \cdot 10^6$, i.e., the norm of the relative residual had not achieved a value of less than 10^{-9}. The performance of multigrid becomes worse on non-equidistant grids. Figure 2 (right) shows that PCG has the better performance compared to multigrid. MG and MG-PCG converge only for a certain number of mesh points, MG-PCG only, if $N_p < 50,000$. Here, further investigations are necessary.

The problem with open boundary conditions has not yet occurred for space charge calculations for several reasons: the algorithms are only performed until the maximum norm of the relative residual is less than 10^{-2}, Dirichlet boundary conditions are applied for meshes with large aspect ratio, the total number of mesh points was less than 1 million. The performance of the multigrid algorithms in a typical tracking simulation will be demonstrated in the next section.

5. MULTIGRID POISSON SOLVERS WITHIN A TRACKING PROCEDURE

The MOEVE Poisson solvers have been implemented in the tracking code Astra (**a** **s**pace **c**harge **tr**acking **a**lgorithm) [3]. The program package Astra has been successfully used in the design of linac and rf photoinjector systems. The Astra suite tracks macro particles through user-defined external fields including the space charge field of the particle cloud.

As a tracking simulation a bunch of 30,000 macro particles representing electrons was chosen. The initial particle distribution is Gaussian with $\sigma_x = \sigma_y = 1.0\,\text{mm}$ and $\sigma_z = 0.5\,\text{mm}$. Further the bunch has a total charge of $-1\,\text{nC}$ and an average energy of 1 MeV. It is tracked over a distance of 2.0 m. Additionally a quadrupole (length 0.2 m, gradient $0.1\,\text{T/m}$) is placed at $z = 1.2\,\text{m}$. The quadrupole stretches the bunch into the transversal y-direction. This behavior requires full 3D space charge calculations.

For the space charge calculations the MG and MG-PCG algorithm was chosen from the MOEVE Poisson solvers. For comparison reasons the same calculations were done additionally using the PCG method. All algorithms were performed until the maximum norm of the relative residual had reached 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 source term ϱ is discontinues in general.

In Table 1 the performance times are given for the equidistant discretization as well as for a non-equidistant discretization. The non-equidistant mesh is constructed as follows: around the bunch the mesh is equidistant, then the mesh is expanded with double step size. Furthermore, the iterative Poisson solvers have the possibility to set their initial guess, in Table 1 denoted by $u_{h,0}$, as the solution of the space charge calculation of the previous time integration step, i.e., $u_{h,0} \neq 0$. This strategy saves iteration steps and consequently CPU time. In particular, the PCG algorithm profits from this strategy so much that it can be nearly as fast as the MG routine (case $N_p = 24,389$). But this behavior depends of course on the problem and cannot be generalized. In the present simulation MG and MG-PCG needed half of the iteration steps with $u_{h,0} \neq 0$ (for instance 1 or 2 steps instead of 3 or 4 with $u_{h,0} = 0$) while PCG could save up to 3/4 or more of the iteration steps (i.e., it had to perform for example only 10 iterations instead of 40).

Table 1: CPU times for a tracking simulation with different Poisson solvers, 988 space charge calculations were performed.

	Total number of grid points	
Poisson solver	24,389	68,921
MG, non-equidistant, $u_{h,0} = 0$	754 s	902 s
MG, non-equidistant, $u_{h,0} \neq 0$	738 s	828 s
MG-PCG, non-equidistant, $u_{h,0} = 0$	790 s	995 s
MG-PCG, non-equidistant, $u_{h,0} \neq 0$	759 s	873 s
PCG, non-equidistant, $u_{h,0} = 0$	806 s	1239 s
PCG, non-equidistant, $u_{h,0} \neq 0$	739 s	881 s

During the whole tracking simulation, space charge was calculated 988 times. Table 1 shows that in general the tracking times together with the MG method are the fastest. Furthermore it

turns out that performance speed of PCG is also very good as long as the number of unknowns is relatively low. Nevertheless comparing the cases $N_p = 24,389$ and $N_p = 68,921$ the computational time rises more for the PCG routine than for the MG algorithm considering $u_{h,0} = 0$.

6. CONCLUSION

The objective of this paper was to test the performance of iterative Poisson solvers, in particular, for large linear systems of equations. It turned out that the multigrid methods MG and MG-PCG achieve the optimal convergence behavior for the Poisson problem with Dirichlet boundary conditions on equidistant and non-equidistant grids, respectively. The solution of Poisson's equation with open boundary conditions pose a problem to the multigrid solvers. Here, further investigations are necessary. Consequently, boundary conditions and solvers have to be chosen carefully in real life applications. Such, it is recommended to avoid open boundary conditions for very large linear systems of equations. For instance, space charge calculations performed on a larger computational domain with Dirichlet boundary conditions provide sufficiently exact results. Another consequence from the numerical results of this paper is a careful choice of the resolution of the problem. A better alternative to a high resolution is an adaptive meshing. First investigations for space charge calculations on adaptive meshes were done in [9]. The presented tracking simulation takes all this considerations into account. It demonstrates the efficiency and robustness of the MOEVE Poisson solvers.

ACKNOWLEDGMENT

This work was supported by DESY, Hamburg, Germany.

REFERENCES

1. Birdsall, C. K. and A. B. Langdon, *Plasma Physics via Computer Simulation*, The Adam Hilger Series on Plasma Physics, New York, 1991.
2. Briggs, W. L., E. Van Henson, and S. McCormick, *A Multigrid Tutorial (2nd Edition)*, Philadelphia, 2000.
3. Flöttmann, K., *Astra*, DESY, Hamburg, www.desy.de/mpyflo, 2000.
4. Greenbaum, A., *Iterative Methods for Solving Linear Systems of Frontiers in Applied Mathematics*, Vol. 17, Philadelphia, 1997.
5. Hackbusch, W., *Multi-grid Methods and Applications*, Springer-Verlag, Berlin, 1985.
6. Jung, M. and U. Langer, "Applications of multilevel methods to practical problems," *Surv. Math. Ind.*, Vol. 1, 217–257, 1991.
7. Markoviḱ, A., "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.
8. Pöplau, G., "MOEVE 2.0: Multigrid Poisson solver for non-equidistant tensor product meshes," Universität Rostock, 2006.
9. Pöplau, G. and U. Van Rienen, "A self-adaptive multigrid technique for 3D space charge calculations," *IEEE Transactions on Magnetics*, 2008.
10. Pöplau, G., U. Van Rienen, and K. Flöttmann, "3D space charge calculations for bunches in the tracking code Astra," *Proceedings of EPAC 2006 (10th European Particle Accelerator Conference)*, 2203–2205, Edinburgh, Great Britain, 2006.
11. Pöplau, G., U. Van Rienen, S. B. Van De Geer, and M. J. De Loos, "Fast calculation of space charge in beam line tracking by multigrid techniques," *Scientific Computing in Electrical Engineering*, W. H. A. Schilders, E. J. W. ter Marten, and S. H. M. Houben, editors, No. 4 in Mathematics in Industry, 329–336, Springer-Verlag, Berlin, 2004.
12. Pöplau, G., 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*, Vol. 40, No. 2, 714–717, 2004.
13. Pulsar Physics, Burghstraat 47, 5614 BC Eindhoven, www.pulsar.nl/gpt. *General Particle Tracer (GPT)*, release 2.70, The Netherlands, 2004.
14. Qiang, J., S. Lidia, R. D. Ryne, and C. Limborg-Deprey, "Three-dimensional quasistatic model for high brightness beam dynamics simulation," *Phys. Rev. ST Accel. Beams*, Vol. 9, No. 4, Apr. 2006.