Fast Multigrid Algorithms for the Tracking of Electron Beams

Gisela Pöplau[†] and Ursula van Rienen

Institute of General Electrical Engineering, University of Rostock, Albert–Einstein–Str. 2, D–18051 Rostock, Germany

(Dated: October 19, 2000)

The simulation of the motion of a particle beam in linear colliders demands fast and robust numerical algorithms which solve the relativistic equation of motion for a large number of time steps and unknowns. The physical grid–based model for the tracking of space charge dominated electron beams requires the solution of Poisson's equation on a rectangular grid with large differences in the step sizes in each of the coordinates. While multigrid techniques are very suitable as fast Poisson solvers, the anisotropic grid rapidly slows down standard algorithms and makes adaptions necessary. In this paper several adaptive multigrid methods are investigated.

INTRODUCTION

An important task in design and operation of future linear colliders is to simulate the behaviour of the particle beam. The related tracking algorithm realized in the tracking code **Q** [10] considers nonlinear space charge effects in electron beams. Consequently, the space charge fields have to be computed in the beam's rest-frame via Poisson's equation for a large number of time steps ($\sim$ 1000) and unknowns ($\sim$ 1 million). This fact demands efficient and robust algorithms. In this paper we are going to investigate the multigrid technique as fast Poisson solver. It turns out that the construction of adaptive multigrid methods is necessary in the tracking context in order to ensure acceptable convergence.

For a short description of the tracking procedure let N be the number of macroparticles in the beam, where the set of macroparticles represents the distribution of all particles in the beam. The i-th macroparticle ($i = 1, \ldots, N$) is assumed to have the position $\vec{r}_i$ and the momentum $\vec{p}_i$. The particle itself has the rest mass m_0 and the charge q. Then the relativistic equations of motion are given by [7]

$$\frac{\partial \vec{r}_i}{\partial t} = \frac{\vec{p}_i}{\gamma_i m_0},$$
$$\frac{\partial \vec{p}_i}{\partial t} = q(\vec{E} + \frac{\partial \vec{r}_i}{\partial t} \times \vec{B}), \qquad i = 1, \ldots, N$$

with the Lorentz factor $\gamma_i = \sqrt{1 + \frac{p_i^2}{m_0^2 c^2}}$, the electric field $\vec{E}$ and the magnetic flux density $\vec{B}$. The nonlinear space charge forces are determined in the beam's rest-frame from the electrostatic potential φ' as a solution of Poisson's equation

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

with the charge density ϱ' and the dielectric constant ε_0. The computational domain Ω is considered to be longitudinally (z–direction) invariant and to have an elliptic or polygonal cross–section in the (x, y)–plane. Transversally, an ideal conducting pipe is assumed (Dirichlet boundary conditions). The boundary condition in longitudinal direction ought to be open describing the decay of the electrostatic potential of the beam. Nevertheless, we restict our considerations to Dirichlet boundary conditions in order to compare the convergence of the proposed algorithms in the numerical studies free of any influence of the formulation for an open boundary condition. The computational domain has to be chosen large enough in this case. The necessary space charge field can be computed by Lorentz transformation of the electrostatic field $\vec{E}' = -\vec{\nabla}\varphi'$ into the laboratory frame.

The three–dimensional rectangular grid is generated adaptively by the grid–based tracking algorithm [10] depending on the distribution of the particles. This discretization results in an anisotropic grid with small step sizes in the centre of the beam and step sizes which become wider near the boundary. Another anisotropy is caused by the Lorentz transformation. There can be a large difference between the smallest mesh sizes of the transversal and the longitudinal direction (see Figure 1 for a small anisotropy). The adaptive discretization doesn't allow the use of the Fast Fourier Transformation as direct Poisson solver. Among the iterative methods the application of multigrid algorithms is state–of–the–art. Further, it is taken into consideration, that a solver for a large number of time steps is needed. In a multigrid iteration, this allows to use the information from the previous time step, which saves a lot of computational work

[†]supported by a research grant from DESY, Hamburg

compared to FFT. In the present paper we investigate only the convergence of multigrid algorithms within one time step. The study of the behaviour of these methods applied to a large number of time steps is foreseen in the next future.

I. ADAPTIVE MULTIGRID ALGORITHMS FOR THE COMPUTATION OF SPACE CHARGE

The discretization of Poisson's equation by finite differences (seven–point stencil) leads to a system of equations

$$A_h u_h = f_h,$$

where h denotes the vector of step sizes on the original (fine) grid. The generation of the grid is adapted to the actual distribution of the particles in the beam. Figure 1 gives an example for such a discretization.

The system of equations above can be solved efficiently by a suitable multigrid algorithm. The multigrid method is characterized by its fast convergence. Compared to classical iteration methods (Jacobi, Gauss–Seidel) the convergence speed remains the same when the discretization is refined. Consequently, the computational work is proportional to the number of unknowns in the system of equations. In this section we give only a short description of the multigrid algorithm. More details can be found e.g. in [2, 11].

The multigrid scheme works as follows:

Step 1 (pre–smoothing): Perform a few steps of a relaxation method (Gauss–Seidel) on the equation

$$A_h u_h = f_h$$

with some initial guess. Compute the residual r_h from the resulting approximation v_h by

$$r_h = f_h - A_h v_h.$$

Step 2 (restriction): Restrict the values of the residual vector r_h to a coarser grid with $r_H = I_h^H r_h$, where H denotes the vector of stepsizes on this level. The coarser grid is obtained by the removal of every second grid line in the case of equidistant meshes. For non–equidistant meshes an adaptive coarsening scheme has to be applied as it is explained later in this section.

Step 3: Solve the system of equations

$$A_H e_H = r_H$$

if the number of unknowns is sufficiently reduced. Otherwise, perform again a few steps of relaxation with initial guess $e_H = 0$, compute the related residual and go down to the next coarser level. The matrix A_H is obtained by applying the finite difference scheme for the discretization on the coarser grid.

Step 4 (interpolation): In order to go up from the coarsest level to the original fine grid the values of the vector e_H have to be interpolated to the next finer level with $e_h = I_H^h e_H$. In the three dimensional case, the interpolation I_H^h is usually the trilinear interpolation.

Step 5 (coarse grid correction): The vector e_h is referred to as the coarse grid correction and it improves the iterate v_h from step 1 with

$$v_h^{new} = v_h + e_h.$$

Step 6 (post–smoothing): Another relaxation step on

$$A_h u_h = f_h,$$

with initial guess v_h^{new} will smooth high–frequency error components which result from the interpolation.

The multigrid scheme explained above is called V-cycle. It goes strictly down to the coarsest level and then again up to the finest level. The reason for the efficiency of the multigrid method can be explained as follows: The relaxation rapidly reduces high–frequency error components, while low–frequency errors are scarcely damped. On the coarser grid the low–frequency error components appear as high frequencies and will be smoothed again by relaxation.

In the case of an adaptive discretization, as in the tracking context, the strategy for the construction of the coarser grids is a crucial point to retain the efficiency of the multigrid algorithm. There are two possibilities: first the geometrical method, that is the removal of grid lines and second the algebraic method which operates on the matrix entries. Advantages of the geometrical strategy are the fast performance and simple implementation. Unfortunately it only works acceptable if the discretization is suitable for the chosen coarsening. Otherwise the algorithm considerably slows down as shown in [6] or it even diverges.

The *geometric multigrid scheme* for the tracking procedure is performed with a *semi–coarsening* technique. The rule for semi–coarsening is as follows: Do not remove a grid line, if the subintervals related to the seven–point stencil are still two times larger than the overall minimal step size of the corresponding level. The objective of this strategy is to obtain, on a certain level, an equidistant grid as far as possible. Conversely, it is most efficient to construct the discretization of the original problem such that it is equidistant at least on the coarsest level.

If the approximation of the pacticle distribution in the tracking process requires a discretization which doesn't support the performance of the geometrical coarsening, the *algebraic multigrid* method (AMG) [8, 12] can be applied. As already mentioned above the algebraic multigrid algorithm uses the matrix entries for the construction of the coarser levels and the interaction between the

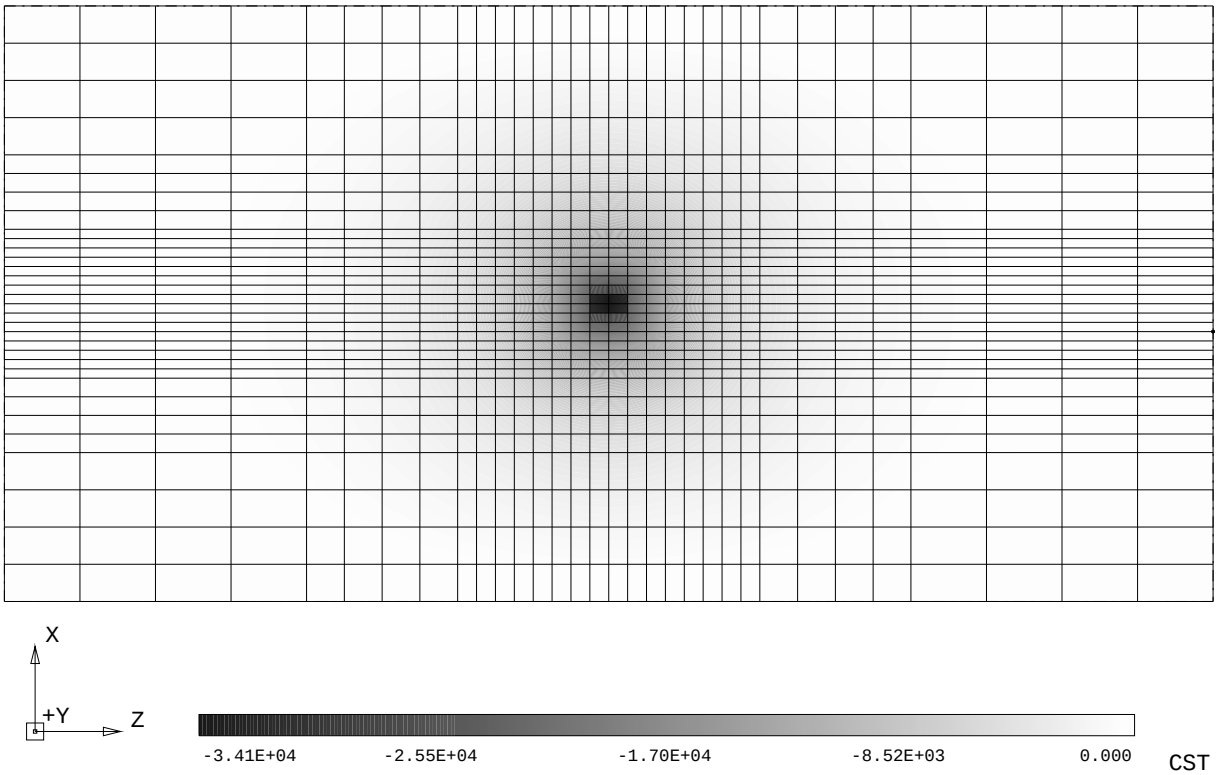

FIG. 1: Discretization of a beam pipe, longitudinal cross–section. The potential φ' of a beam with Gaussian distribution of the space charge density is shown.

levels. The effect of the geometric multigrid with semi–coarsening is similar to the algebraic method, which distinguishes between strong and weak coupled variables for the selection of the coarse grid points. The advantage of AMG is that the adaption to the non–equidistant grid is done automatically. Thus, the algorithm can be used as a black box solver as we did for our numerical tests with the AMG code of Ruge and Stüben [9]. On the other hand the setup phase in AMG, i.e. the choice of the coarser levels and the computation of the interpolation and restriction operators, takes a lot of time.

II. NUMERICAL RESULTS

Numerical experiments have been performed with the data of a 10 MeV beam in a cylindrical pipe. The space charge density has Gaussian distribution in longitudinal and equidistribution in transversal direction.

Geometric and algebraic multigrid are tested and compared to a conjugate gradient method with ILU(3) preconditioner as it is implemented in the software package MAFIA [1]. The geometric multigrid algorithm performed as a V–cycle has the components [2, 11]: red–black Gauss–Seidel relaxation with two pre–smoothing steps and one post–smoothing step, 'half'–weighting restriction, trilinear interpolation and semi–coarsening for the construction of the coarser levels. The algebraic multigrid method is applied as black box solver with the AMG code of Ruge and Stüben [9].

First the numerical results on an equidistant grid with 65x65x65 (=274 625) mesh points are given. Here, only a large anisotropy with $h_z \approx 25h_x \approx 25h_y$ poses a difficulty. Thus, the standard coarsening is performed and semi–coarsening is only applied in the following sense: do not coarsen the mesh of the longitudinal direction, if the step size is still two times larger than the step size of the transversal directions. Figure 2 shows that the best results are obtained with the geometric multigrid method. Neglecting the setup phase, the convergence speed of AMG has nearly the same slope. The preconditioned conjugate gradient method (PCG) needs a considerable amount of work for the preconditioning, which is rather costly for the ILU(3)–preconditioner.

The situation changes on the non–equidistant grid. Two different examples with respect to the influence of the Lorentz transformation have been investigated: a small anisotropy with $h_{z\,min} = 2h_{x\,min} = 2h_{y_{min}}$ and a large anisotropy with $h_{z\,min} = 10h_{x\,min} = 10h_{y_{min}}$. Both examples have been performed on a discretization with either 33x33x33 (=35 937) or 65x65x65 (=274 625) mesh points, that is the second grid is twice as fine as the first grid. Figure 1 shows the discretization in the case of the small anisotropy on the 33x33x33–mesh. Only for the

numerical studies the numbers of grid points are chosen equally in each coordinate. In real application, the tracking procedure would generate a grid with more points in longitudinal and perhaps less points in transversal direction.

The numerical results for the discretization with the small anisotropy are given in Figures 3 and 4. It turns out, that the convergence of the geometric multigrid with semi–coarsening is still good, but not as good as in the equidistant case. After the setup phase, the convergence of AMG is slower than for the geometric method and the setup phase itself is very long in the case of a small anisotropy.

Figures 5 and 6 show that the convergence speed of the different multigrid algorithms has changed. The geometric multigrid remarkably slows down compared to the discretization with the small anisotropy. The reason for this behaviour is the varying step size on the coarsest grid. Consequently, the discretization should be changed to reach a better convergence. The performance of AMG has comparatively improved, that means the setup phase is much shorter and the convergence speed is better than in the previous example. The algebraic multigrid is faster for discretizations with large anisotropies, because the differences in the magnitude of the matrix coefficients are larger than in the case of small anisotropies. Thus, the construction of less levels is necessary [12].

For both examples the conjugate gradient algorithm converges very fast after the preconditioning process, independently of the anisotropy. This behaviour can be explained by the spectrum of the matrix A_h. The eigenvalues for the non–equidistant discretization form clusters. This property leads to a faster convergence than in the equidistant case [3].

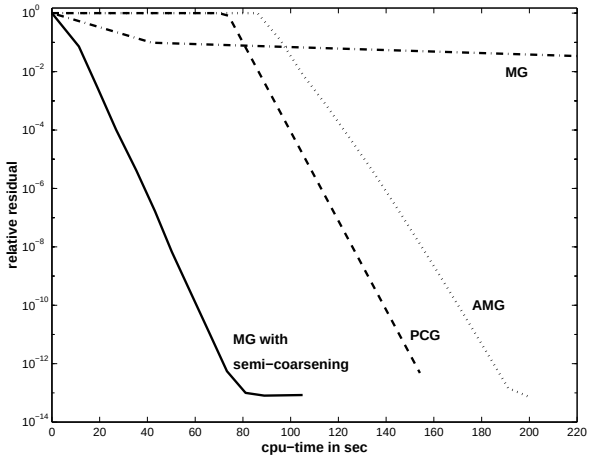

FIG. 2: Convergence of several algorithms (MG: multigrid, AMG: algebraic multigrid, PCG: preconditioned conjugate gradient) for an equidistant discretization on a 65x65x65–mesh with a large anisotropy in longitudinal direction.

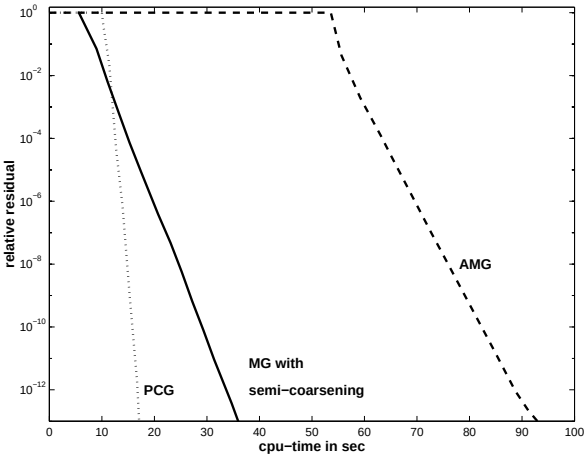

FIG. 3: Convergence of several algorithms for a non–equidistant discretization on a 33x33x33–mesh with a small anisotropy in longitudinal direction.

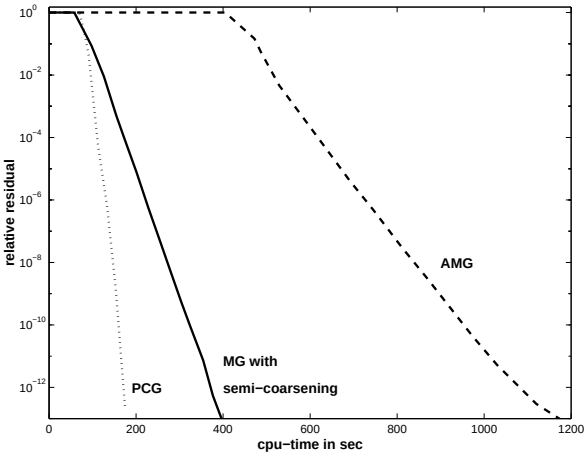

FIG. 4: Convergence of several algorithms for a non–equidistant discretization on a 65x65x65–mesh with a small anisotropy in longitudinal direction.

III. CONCLUSIONS

Two multigrid strategies have been investigated in this paper. The numerical results show, that geometric multigrid with semi–coarsening works very fast on equidistant grids. Yet, non–equidistant meshes ought to be constructed appropriate to the coarsening process. The best way to do so is to start with an equidistant grid and to refine it in accordance to the distribution of the particles of the beam. Investigations in that field are still going on.

It is possible to apply an algebraic multigrid method if a suitable discretization for a geometric multigrid algorithm would not approximate the real tracking problem sufficiently. The disadvantage of the long setup phase can probably be overcome with the recent development

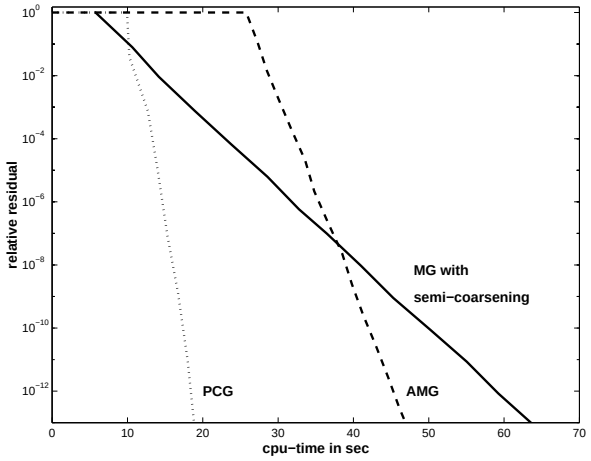

FIG. 5: Convergence of several algorithms for a non–equidistant discretization on a 33x33x33–mesh with a large anisotropy in longitudinal direction.

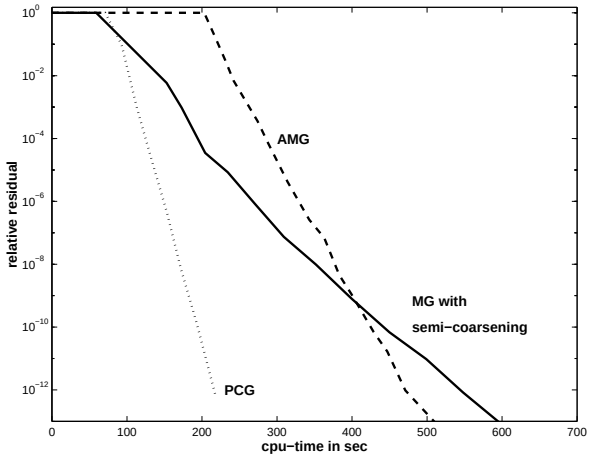

FIG. 6: Convergence of several algorithms for a non–equidistant discretization on a 65x65x65–mesh with a large anisotropy in longitudinal direction.

of AMG [12], which is not yet available for common use.

The good convergence of the conjugate gradient algorithm for the non–equidistant grid encourages further investigations with respect to the application of multigrid as preconditioner instead of ILU(3).

Last but not least the choice of the solution method for the tracking problem will depend on its behaviour for the simulation of the particle motion over a large number of time steps. Investigations in that field are planned for the next future.

REFERENCES

[1] MAFIA Version 4.0, CST GmbH, Büdinger Str. 2a, D-64289 Darmstadt, Germany.

[2] W. Hackbusch, *Multi-Grid Methods and Applications*, (Springer, Berlin, 1985).

[3] W. Hackbusch, *Iterative Lösung großer schwachbesetzter Gleichungssysteme*, (B. G. Teubner, Stuttgart, 1991).

[4] G. Pöplau, U. van Rienen, *Multigrid Algorithms for the Tracking of Electron Beams*, in Multigrid Methods VI, (E. Dick, K. Riemslagh, J. Vierendeels, eds.), LNSCE, Vol. 14, (Springer, Berlin, 2000, 214–220).

[5] G. Pöplau, U. van Rienen, J. Staats, T. Weiland, *Fast Algorithms for the Tracking of Electron Beams*, to appear in the proceedings of the EPAC 2000, (Vienna, 2000).

[6] G. Pöplau, U. van Rienen, *Fast Multigrid Solvers for Poisson's Equation in Computational Electromagnetics*, submitted to the proceedings of the Conference on Scientific Computation in Electrical Engineering (SCEE), (M. Günther, U. Langer, U. van Rienen, eds.), to appear in LNSCE, (Springer, Berlin, 2001).

[7] M. Reiser, *Theory and Design of Charged Particle Beams*, (John Wiley & Sons, Inc., New York, 1994).

[8] J. Ruge, K. Stüben, *Algebraic Multigrid (AMG)*, in Multigrid Methods, (S.F. McCormick, ed.), SIAM, Frontiers in Applied Mathematics, (Philadelphia, 1987).

[9] J. Ruge, K. Stüben, AMG1R5, release 1.5 Oct. 1990.

[10] J. Staats, T. Weiland, S. Kostial, A. Richter, *Tracking of Electron Beams with Numerically Determined Space Charge Forces*, in Proceedings of the 1999 Particle Accelerator Conference PAC'99, (New York, 1999, 2740-2742).

[11] K. Stüben, U. Trottenberg, *Multigrid Methods: Fundamental Algorithms, Model Problem Analysis and Applications*, GMD-Studien 96, (St. Augustin, 1984).

[12] K. Stüben, *Algebraic Multigrid (AMG): An Introduction with Applications*, GMD Report 53, (St. Augustin, 1999).