DEVELOPMENT OF A SELF-CONSISTENT PARTICLE-IN-CELL (PIC) CODE USING A TIME-ADAPTIVE MESH TECHNIQUE*

S. Schnepp**, E. Gjonaj, T. Weiland, Technische Universitaet Darmstadt,
Institut fuer Theorie Elektromagnetischer Felder, Schlossgartenstr. 8, 64289 Darmstadt, Germany

Abstract

For a large class of problems the self-consistent simulation of charged particle beams in linear accelerators is necessary. Typically, Particle-In-Cell (PIC) simulations use a fixed computational grid which has to resolve the bunch adequately. This leads to enormous memory consumption. Therefore, and especially in the 3D case, only rather short sections can be simulated. A remedy to this limitation is the usage of a dynamic grid which is automatically refined in the vicinity of particles. For this purpose, a new code, SMOVE, based on a time-adaptive grid is being developed. First promising results are presented in this paper.

INTRODUCTION

The most critical issue in the simulation of long accelerating structures is the extreme multi-scale character of the problems. Considering the Photo Injector Test Facility at DESY Zeuthen (PITZ) [1] as an example, the gun section has a length of 2.4 m from the photocathode to a nine cell TESLA-like booster. At this position a global minimum in transverse emittance has to be achieved. Since the emittance at this point is crucial for the operation of the downstream *Free Electron Laser* (FEL), reliable and accurate simulations using a self-consistent algorithm including all physical effects are essential.

The laser pulse has a longitudinally flat-top time profile with a FWHM length of 20 ps and a rise and fall time of 2 ps. The emitted bunches evolve to an approximate length of 10 mm after emission from the photocathode. From previous simulations it is known that a longitudinal mesh resolution down to 20 μm in the vicinity of the cathode is needed in order to cover all aspects of the particle motion in this low-energetic region [2]. Although the spatial resolution may be decreased at a distance of some millimeters from the cathode, a resolution of about 50 mesh lines along the bunch length is still mandatory for accurate results.

To avoid discretizing the whole computational domain using this very small step size an adaptive mesh refinement (AMR) technique is proposed. Using this approach, memory requirements and computational time are significantly decreased while the accuracy of the results is not affected.

* Work supported by HGF under contract VH-FZ-005
** schnepp@temf.tu-darmstadt.de

DYNAMIC MESH REFINEMENT

Mesh Refinement and Coarsening

For the purpose of generating an adaptive mesh a coarse base grid is constructed. In a second step the mesh is refined in the transversal directions matching the bunch dimensions. Since we consider injectors and linear accelerators the transversal position of the bunch is not varying and hence the transversal discretization remains static. In the longitudinal direction the mesh is refined by hierarchical splitting of cells. In each refinement step one plane of base grid cells and their descendants are refined by bisection. The mesh step size Δ therefore decreases according to

$$\Delta = \Delta_{base}/2^N, \tag{1}$$

with N being the refinement level.

In figure 1 snapshots of the computational grid along with the corresponding space charge fields of a bunch in the 1.5-cell cavity of the PITZ gun is shown. The underlying structure of the hierarchical grid is used for providing automatic mesh refinement ahead of the bunch as well as

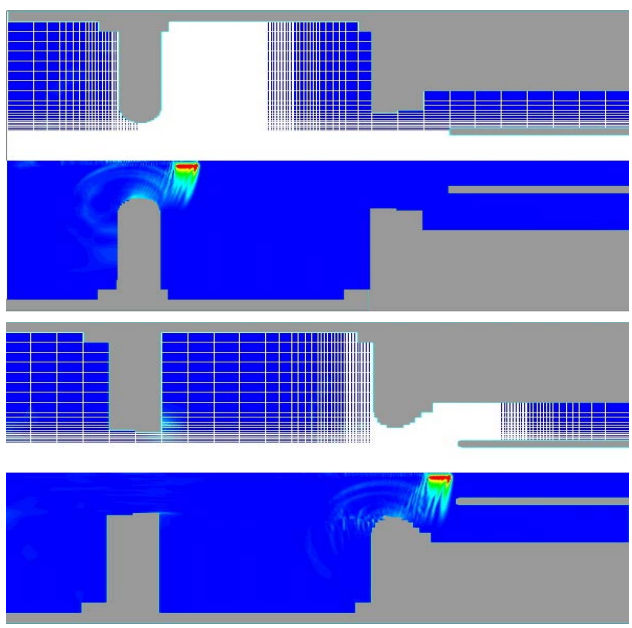

Figure 1: Computational grid and space charge fields for the PITZ gun. The discrete material representation changes with time according to the local resolution of the dynamic grid. This can be seen by comparing the metallic contours in both snapshots.

automatic coarsening in the regions left behind the particles. The difference in refinement levels of neighboring cells is limited to one in order to keep numerical reflections at a low level [3].

Interpolations

The solution of Maxwell equations for the discretizations shown in figure 1 is obtained by applying the *Finite Integration Technique* (FIT) [4]. However, in order to apply the method, the values of field components have to be interpolated to their new position after each grid refinement or coarsening process. The accuracy of the simulation results critically depends on the numerous interpolations which may even lead to numerical instabilities. Therefore, several interpolation schemes have been tested. The simplest one is the linear interpolation. Linear interpolation does not produce any overshooting (see fig. 2) which is a direct source for instabilities. However, a linear approximation of the high-frequency fields in the vicinity of the bunch is not sufficiently accurate. More accurate results were achieved using spline interpolation.

Since many splines show significant overshooting (see fig. 2) a class of sub-splines has been selected. Sub-splines differ from the general definition of splines in their order of continuous differentiability. A spline of polynomial order p is $p-1$ times continuously differentiable. In the case of sub-splines, the continuity demands are relaxed in order to minimize the overshooting effects. Thus, a less oscillatory interpolation of order $k < p-1$ is generally obtained.

In order to minimize the oscillatory behavior, continuity demands are relaxed. Sub-splines are continuously differentiable up to an order of k with $k < p-1$.

To determine the coefficients of a spline interpolation, usually a system of equations has to be solved. This makes their application very expensive in codes which use interpolations heavily. The class of Akima-splines is well suited to this problem. Their coefficients can be calculated explicitly by combining first order differences of five supporting points [5].

Furthermore, in order to reduce numerical noise the spline coefficients may be modified using the *slope limiter* technique. Slope limiters were originally introduced in reconstruction theory for *Finite Volume* methods. The slopes of neighboring intervals are compared using, e.g., the minmod operation [6]:

$$\text{minmod}(a,b) := \begin{cases} a; & \text{if } |a| < |b|, \quad a \cdot b > 0 \\ b; & \text{if } |a| > |b|, \quad a \cdot b > 0 \\ 0; & \qquad\qquad\quad a \cdot b < 0 \end{cases} \tag{2}$$

Minmod chooses the smallest argument in absolute value if they have equal sign and zero otherwise. This modification leads to a smoothing of the spline interpolation since alternating gradients in neighboring intervals are flattened (see fig. 2).

Dispersion Properties

When increasing the spatial resolution in a numerical simulation, the numerical phase velocity of propagating waves approaches the physical phase velocity. This behavior was investigated with the code SMOVE [7] working on the basis of the time-adaptive mesh algorithm described above.

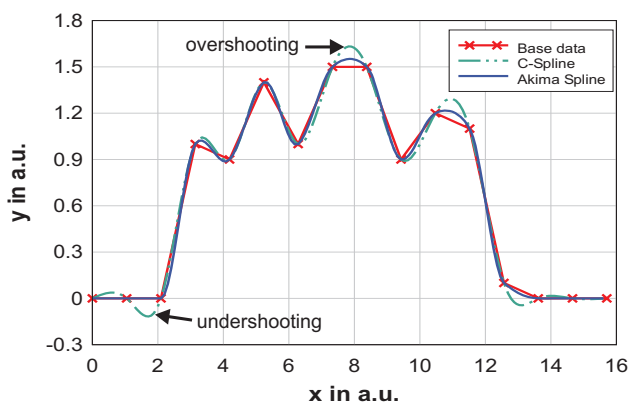

Figure 2: Comparison of a C-Spline and an Akima-Spline. The Akima-Spline shows less overshooting and because of the slope limiting it is smoothed out to a constant at both ends of the curve.

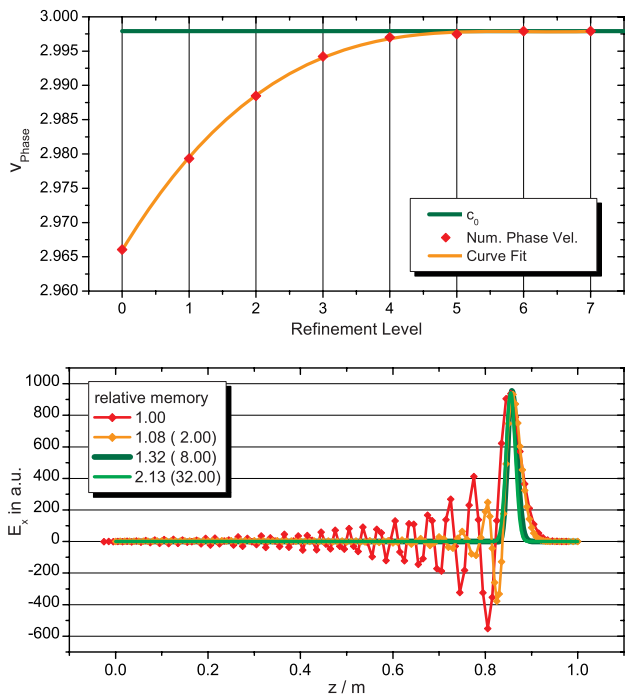

Figure 3: With increasing refinement level the resolution increases and the numerical phase velocity approaches the speed of light in vacuum (top). Gaussian packet for the refinement levels 0, 1, 3 and 5. In the legend the memory needed for the time-adaptive mesh approach and a static grid of equal resolution (in brackets) are given (bottom).

In a first example the phase velocity of a mono-frequent sinusoidal wave was determined depending on the refinement level. In figure 3 (top) the numerical phase velocity is plotted for refinement levels of zero up to seven along with a fitted curve. Level zero corresponds to the non-refined base grid resolving one wavelength with five grid points ($\Delta = \lambda/5$). A refinement level of seven corresponds to $2^7 = 128$ refinement steps resulting in a resolution of $\Delta = \lambda/640$.

In a second investigation a traveling wave of Gaussian shape was considered. Starting from a resolution of eight points, the refinement level was increased until no dispersion error in the numerical solution could be observed after a distance of five meters. Figure 3 (bottom) shows the result (on the first meter) for different refinement levels. The most remarkable result is the memory consumption. In order to achieve an accurate transmission using the time-adaptive mesh approach, memory requirements are increased by a factor of ≈ 2.13 compared to the base grid. When using a static grid of the same resolution this factor is 32.

EXAMPLES

The code is validated using the PITZ gun as an example. In figure 4 the evolution of the transversal RMS beam size

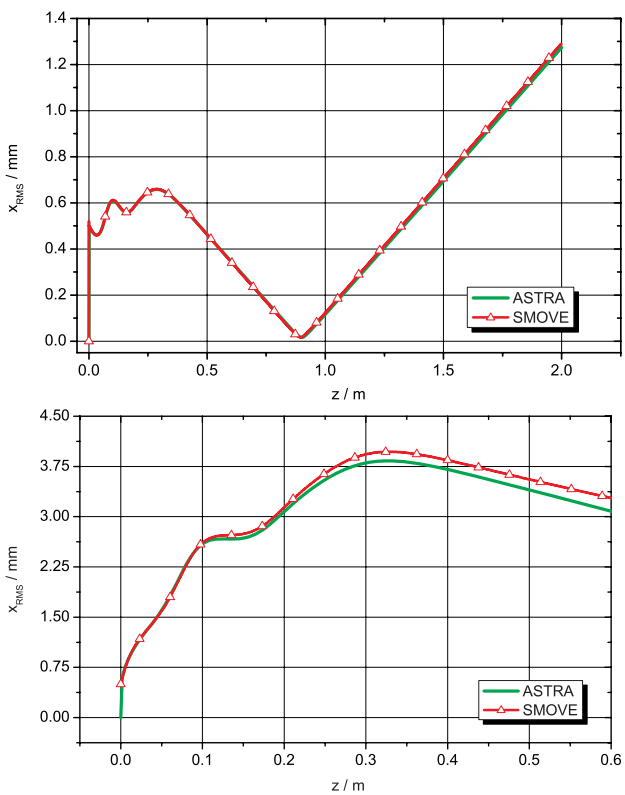

Figure 4: x_{RMS}-size calculated by SMOVE and ASTRA in tracking mode (top) and in self-consistent mode (bottom). The simulation of longer distances is easily possible without any memory problems.

in tracking mode and in self-consistent mode are given. For comparison the results obtained with the code ASTRA [8] are shown in the same diagram.

In the current implementation stage of SMOVE, there is only the so-called *kick* or *nearest grid point* scheme for the particle-to-grid interpolation available which suffers from severe numerical noise. Additionally, the resolution has to be extremely high to capture all effects of the particle motion. In the near future an interpolation scheme of higher accuracy will be implemented [9]. This will relax the demands on grid resolution, the number of computational particles needed and lead to more accurate results. Simulations using a minimum grid spacing of 11 μm showed already good agreement with the self-consistent simulations presented in [10].

CONCLUSIONS

A new approach to self-consistent simulations of charged particles including geometry effects was presented. It is based on a time-adaptive mesh which is automatically refined in the vicinity of particles. Due to the considerable reduction of memory demands it is well-suited for the application to long accelerator structures. First results obtained utilizing the code SMOVE, based on this method, were presented. They are in good agreement to results achieved using other methods. The implementation of a more accurate algorithm for the coupling of particles and fields is scheduled for the near future.

REFERENCES

[1] A. Oppelt, “The Photo Injector Test Facility at DESY Zeuthen: Results of the First Phase” Proc. LINAC 2004.

[2] M. Zhang, P. Schuett, “TESLA FEL Gun Simulations with PARMELA and MAFIA”, Computational Accelerator Physics Conference (CAP), 1996, Williamsburg, Virginia, pp. 21-26.

[3] P. Thoma, “Zur numerischen Loesung der Maxwellschen Gleichungen im Zeitbereich” Darmstaedter Dissertationen, 1997, p. 68.

[4] T. Weiland, “A Discretization Method for the Solution of Maxwell’s Equations for Six-Component Fields”, Int. J. Elec. and Comm. (AEÜ), Vol. 31 (1977), pp. 116 - 120.

[5] C. de Boor “A Practical Guide to Splines” Springer-Verlag Appl. Math. and Sc. New York, 2001, vol. 27, p. 40.

[6] C. Laney, “Computational Gas Dynamics” Cambridge University Press, 1998, pp. 158.

[7] SMOVE – A Code for Self-Consistent Particle Beam Simulations over long Distances “pdf-Documentation” http://www.temf.de/downloads/SMOVEdoc.pdf

[8] ASTRA, http://www.desy.de/∼mpyflo/

[9] J. Villasenor, O. Buneman, “Rigorous charge conservation for local electromagnetic field solvers” Comp. Phys. Comm. 69, pp. 306-316, 1992.

[10] E. Gjonaj, T. Lau, S. Schnepp and T. Weiland, “Accurate modelling of charged particle beams in linear accelerators” *Submitted to New J. of Phys.*