PARALLEL PARTICLE IN CELL COMPUTATION OF AN ELECTRON GUN WITH GdfidL

W. Bruns, WBFB, Berlin, Germany *

Abstract

The paper describes an efficient algorithm to integrate the equations of a fast moving charge cloud of small size in a large electron gun. Particle in cell computation of a realistic electron gun is challenging due to the large discrepancy between the size of the cavity and the size of the cloud. A fine grid must be used to resolve the small volume of the charge, with a grid spacing in the order of 0.1 mm. The cavity has extensions of about 100 mm. Therefore one has to deal with about 1000 million gridcells. Such a large grid is handled best with parallel systems. Each node of the parallel system computes the electromagnetic field in its subvolume. As the extension of the charge keeps being small during the flight, at each timestep the charged particles will be located in only a few subvolumes of the nodes of the parallel system. This would lead to a strong load imbalance, if the particle related computations for each particle would be performed by the node where the particle is in. GdfidL instead spreads the data of most particles over all processors, which then perform the particle related computations, and send back the results to the processors where the particles are in.

THE PHOTO GUN

The resonator has a diameter of 10cm, and the height of the most interesting part of the gun is about 10cm. At the bottom of the gun, a charge with a diameter of 3mm is emitted via a laser-pulse. The duration of the pulse is so short, that the emitted charged cloud has a length of 5 mm. Because the gradient in the gun is about 100 MV/m, the emitted charge is accelerated very rapidly to relativistic velocities. The size of the charge cloud therefore stays small.

The geometry itself is rotational symmetric, and the charge ideally would be rotational symmetric. To investigate what effect an offset of the laser pulse from the axis would have, one needs to perform a three dimensional computation, without any planes of symmetry.

Because the charge has an extension in the order of 3 mm, one needs a gridspacing of 0.1 mm or less. Because we want to resolve small effects due to a small deviation from a rotational symmetric case, we want to compute with a homogeneous grid, minimising dispersion errors and reflections due to an inhomogeneous grid. This leads to a total number of gridcells in the order of 1000 millions. A naive implementation of the FDTD-algorithm would then require about 50 GBytes of RAM. Such large grids are best handled by clusters of PCs.

* bruns@gdfidl.de

DOMAIN SUBDIVISION

The computational volume is partitioned in many more subvolumes than the number of available processors. Each subvolume is inspected, whether it is filled only with electric conducting material. The fully electric conducting subvolumes are discarded and the remaining ones are spread evenly over the processors. This way, each processor has about the same number of interesting gridcells to compute the fields in, leading to a good load balancing for the field computation.

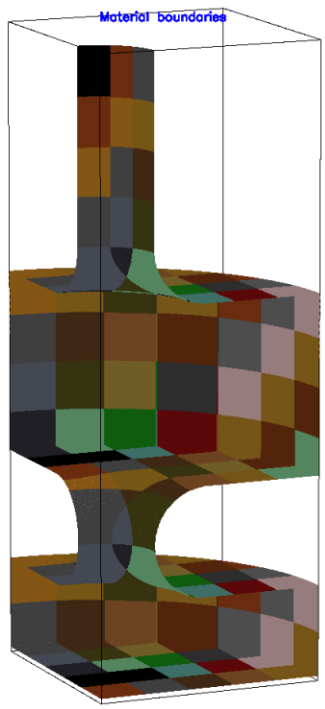

Figure 1: A model of the gun. The grid spacing is 0.1 mm. The total number of gridcells in this quarter of the total volume is 212 millions. Less than 40% of the computational volume is filled with vacuum cells. The different colours indicate the used subvolumes. The shown volume is partitioned in 325 subvolumes, of which 167 are discarded, since they do not have a single vacuum cell. A naive implementation of FDTD in this quarter of the volume would require 10 GBytes of RAM. GdfidL uses 4.3 GBytes, spread evenly over the available processors.

Local Field Computation

Each processor computes the electromagnetic fields in its local subvolumes. The tangential electric and magnetic fields of the neighbour volumes are boundary conditions

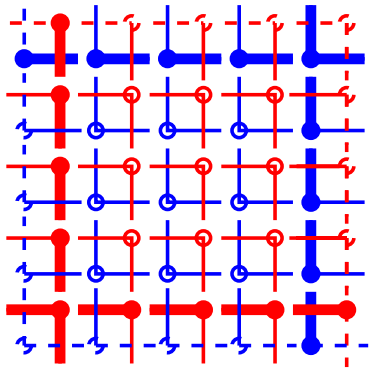

Figure 2: The blue lines and circles represent the electric field components in a local volume. The red ones represent the magnetic field components. The tangential E field components at the upper boundaries of the local volume (thick blue) and the tangential H field components at the lower boundaries (thick red) can be computed from the local information. These components are sent to the neighbour volumes. The tangential E field at the lower boundaries (dashed blue) and the tangential H field at the upper boundaries (dashed red) cannot be computed from the local information. These components are received from the neighbours.

for the local field computation, and are communicated between the processors at each timestep.

SIMULATING FREE MOVING CHARGES

Moving charges change the electric field in their immediate vicinity, and they are accelerated by the electric field and rotated by the magnetic field at their actual positions.

The computation of the acceleration and rotation is performed by an algorithm invented by Boris [Boris 1970]. The optimised implementation of Buneman, as can be found in the TRISTAN code, is used.

Moving charges show up in MAXWELLS equations as a current density. The change of electric flux through a face must equal the total amount of charge which enters a grid-cell through that face. The computation of the charge is done with an algorithm similiar to the one described by Villasenor [Villasenor 1992], extended for a grid with uneven spacings and charged particles of arbitrary size.

Since the computation of the acceleration requires knowledge about the electromagnetic field at the positions of the particles, and since the electric field near the particles are changed due to their convection current, it is natural to perform all the charge related computations on the processor who is responsible for computing the electromagnetic field within the volume where also the charge is in. While the charges are drifting through the computational volume, they eventually leave the domain of one processor and enter the domain of another processor. At each timestep, the data of the leaving charges must be sent to the processor who is responsible for the volume they are entering.

When simulating a photo-gun on a parallel system, this approach leads to a strong load-imbalance, as the charged cloud in a photo-gun has small extension during the flight through the gun. For most timesteps, almost all macroparticles are within the subvolume of a single processor. When that single processor has to perform all charge related computations, the other processors will have to wait for completion of the computations.

Instead, a small rectangular volume is selected, where almost all macroparticles are in, see figure 3. The data of the contained macroparticles are spread evenly over the processors. Each processor performs the charge related computations for its part of macroparticles within the box.

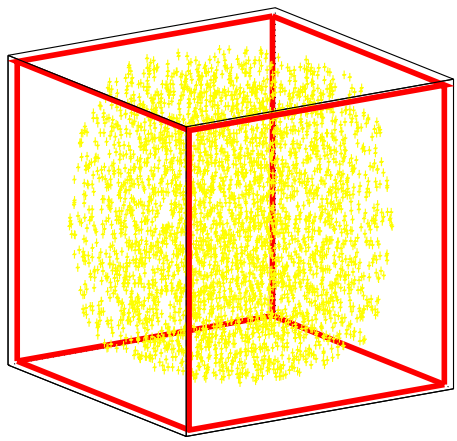

Figure 3: A cloud of macroparticles in a box which encloses 99% of the particles. The macroparticles within the box are spreaded evenly over the available processors. The electromagnetic field within this box must be sent to all processors, and the parts of the convection current due to the macroparticles is computed by each processor and sent to selected processors. The time required for these communications is proportional to the volume of the box. The time dependend borderplanes of the volume are found via adapting their positions at each timestep.

The macroparticles are accelerated by the electric field and rotated by the magnetic field. For computing this acceleration and rotation, the processors must know the electromagnetic field near the center of mass of the macroparticles. For this, the electromagnetic field within this box is gathered from the processors which know that field, and scattered to all processors.

In MAXWELLS equation, the moving charges show up as a convection current, and therefore change the electric field in their vicinity. The total convection current due to the macro-particles in the small box is computed by each processor for its macroparticles. The sum of all the convection currents is sent to the processors which are responsible for computing the electromagnetic field within the box.

The computations for the macroparticles outside of the small box are performed by the processors who are responsible for the subvolume they are in.

Additional Communication

The quoted times and numbers in this section refer to a quarter of the gun. For the photo gun, the size of the box can be so small that the box contains less than 150.000 gridcells, see figure 4. The average is 100.000 cells. This is less than 0.04 % of the total number of gridcells. In each timestep, the electromagnetic field of these 100.000 gridcells must be gathered and scattered. With a logarithmic communication scheme, this requires $2\times(\log_2(N_{CPUs})-1)$ communication steps, each transferring 100.000 x 6 x 4 Bytes. The gathering and scattering of the convection current also requires $2\times(\log_2(N_{CPUs})-1)$ communication steps, each transferring 100.000 x 3 x 4 Bytes. Using up to 16 CPUs connected via 100mbit ethernet, the communication requires about $2\times(\log_2(16)-1)\times 100.000\times(6+3)\times 4\ \text{Byte}/(10\ \text{MByte}/s) = 2.2$ seconds. Using a cluster of 12 AMD-Athlon processors, the wall clock time per timestep is 11 seconds, see figure 5. The communication due to the gathering and scattering of the box-data therefore requires 20 % of the total time.

The total wall clock time to simulate the acceleration of 2 million macroparticles in a mesh of 220 million gridcells is 25.000 seconds = 7 hours on a cluster of 12 AMD-Athlons.

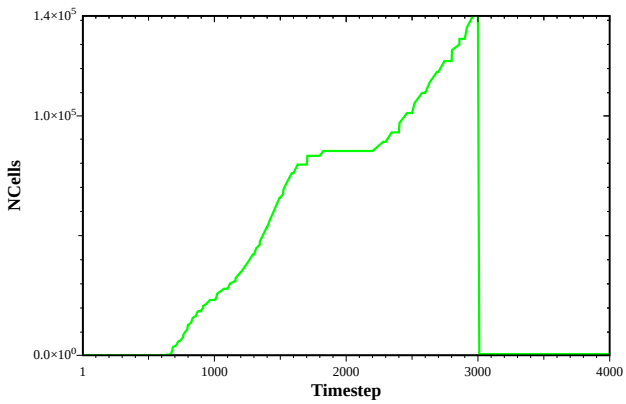

Figure 4: The number of gridcells in the charged box as a function of time. The charge is emitted near the 600.th timestep, and it exits the computational volume near the timestep 3000.

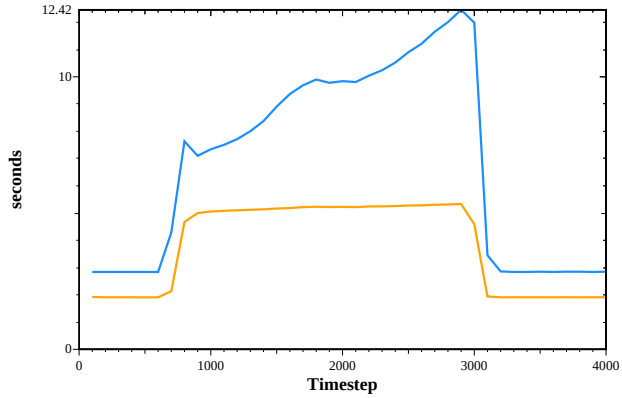

Figure 5: Upper curve: The wall clock time to compute a timestep as a function of time. Lower curve: The CPU time per processor. The wall clock time rises because the required communication rises strongly due to the growing box. The needed CPU time rises slightly because the gathering and scattering of the box-data in the growing box also consume CPU time. The charge is emitted near the 600.th timestep, and it exits the computational volume near the timestep 3000.

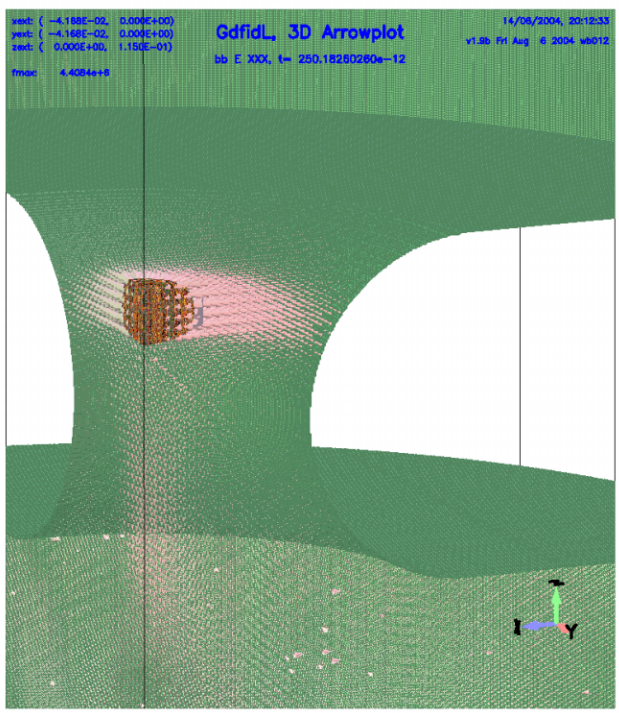

Figure 6: The bunch and the wakefield of the bunch when they just enter the second cell of the cavity. There are 2 million macroparticles in the bunch. Only the wakefield is shown. The fieldstrength of the wakefield is 5 MV/m, while the gradient of the accelerating field is 100 MV/m. The particles are already accelerated to a velocity of $\beta = 0.98$. The volume which contains 99% of the macroparticles consists of less than 60.000 gridcells.

REFERENCES

[1] Boris, J. P., "Relativistic plasma simulation-optimization of a hybrid code", in *Proceedings of the Fourth Conference on the Numerical Simulation of Plasmas* Naval Res. Lab., Wash. D.C., 1970.

[2] J. Villasenor, O. Bunemann, "Rigorous charge conservation for local electromagnetic field solvers", Computer Physics Communications 69 (1992) 306

[3] http://webserv.gsfc.nasa.gov/ESS/exchange/contrib/macneice/pic-tristan.html