

PARALLEL PARTICLE IN CELL SIMULATIONS WITH GDFIDL

W. Bruns, WBFB, Berlin, Germany

Abstract

The electromagnetic field solver GdfidL has been extended to compute with free moving charges. For computing in parallel, GdfidL partitions the computational volume in many small subvolumes. Each processor computes the electromagnetic field in its part of the whole volume. In addition to the normal field update equations, the movement of the particles must be computed from the Lorentz-force, and the convection current due to the moving charges must be computed and be used to change the electric field near the particle. For each particle, these computations are performed by the processor which is responsible for the volume where the particle is in. Details of the parallel implementation of the used algorithm, Particle in Cell, are given.

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.

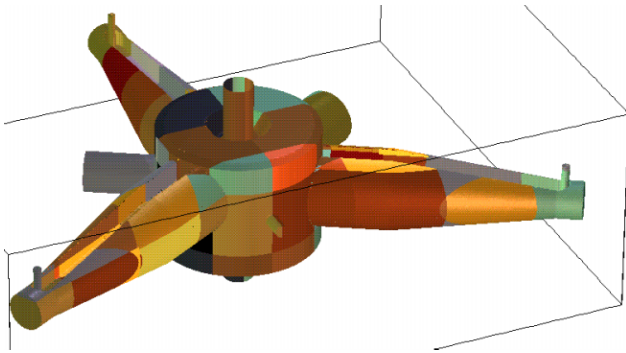


Figure 1: A model of the BESSY cavity with attached waveguides and tuning plungers. Less than 10% of the computational volume is filled with vacuum cells. The different colours indicate the used subvolumes. The total volume is partitioned in 192 subvolumes, of which 122 can be discarded, since they do not have a single vacuum cell.

Local field computation

The core of the Finite Difference Method is the discretisation of the curl-operators. With these discretised curl operators, one computes time dependent fields (FDTD) via

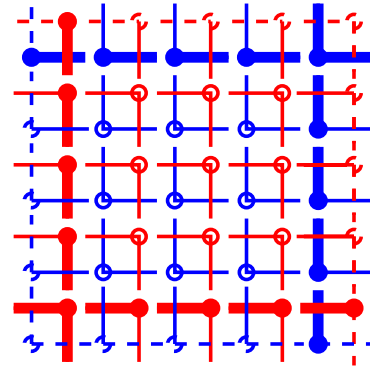


Figure 2: The blue lines and circles represent the electric field components in a local volume. The red ones represent the magnetic field components. The tangential E field components at the upper boundaries of the local volume (thick blue) and the tangential H field components at the lower boundaries (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.

the discretised form of

$$\vec{H}((n + 1/2)\Delta t) = \vec{H}((n - 1/2)\Delta t) - \Delta t \frac{1}{\mu} \nabla \times \vec{E}(n\Delta t)$$

$$\vec{E}((n + 1)\Delta t) = \vec{E}(n\Delta t) + \Delta t \frac{1}{\epsilon} \nabla \times \vec{H}((n + 1/2)\Delta t)$$

Most of the CPU-time is spent in applying these discretised curl operators. However, they are quite easily parallelised. The field update for more than one subvolume per processor can be done via:

```

For all Subvolumes: DO
  Compute local H by applying the local
  curl operator to the local E
ENDDO For all Subvolumes
For all Subvolumes: DO
  For all Directions: DO
    Send tangential H to the neighbour
    Receive tangential H from the neighbour
  ENDDO For all Directions
ENDDO For all Subvolumes
For all Subvolumes: DO
  Compute local E
ENDDO For all Subvolumes
For all Subvolumes: DO
  
```

```

For all Directions: DO
  Send tangential E to the neighbour
  Receive tangential E from the neighbour
ENDDO For all Directions
ENDDO For all Subvolumes
    
```

SIMULATING FREE MOVING CHARGES

Moving charges change the electric field in their immediate vicinity, and their velocity is changed by the electric and magnetic field in their immediate vicinity. Therefore 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.

Free moving charges are simulated by the 'Particle in Cell'-algorithm. The algorithm uses a large number of charges of finite size and fixed shape, see figure 3.

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. That charge amount, multiplied by a factor of $\frac{1}{\int \epsilon dA}$, is the amount by which the electric field component changes, in addition to the change due to the curl of H.

Integration of the convection current

In the simulation, each charge is treated as a homogeneously charged rectangular box. Then the current through a face is proportional to the area of the box within the cellface, multiplied by the velocity of the box. For a cellface in the x-y-plane, the charge entering through the face is the integrated current:

$$Q \sim v_z \int_{t=t_1}^{t=t_2} f_x(t) f_y(t) f_z(t) dt \quad (1)$$

Here, v_z is the z-component of the drift velocity, f_x, f_y are piecewise linear functions, and $f_z(t)$ is a rectangular window function. $f_z(t)$ is non-zero only for times t, when the charged box actually is crossing the x-y-plane. The actual shape of the functions f_x, f_y, f_z do depend on the position of the gridplanes, the velocity, the position and the size of the charge. The integration can be performed exactly.

Particle pushing

The charge is accelerated by the electric field and rotated by the magnetic field at its actual position. The computation of the acceleration and rotation is performed by an

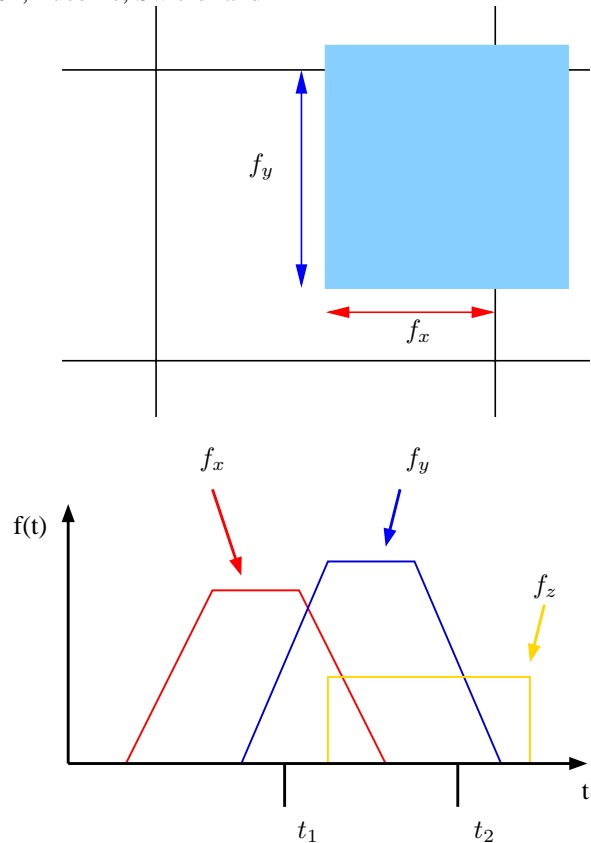


Figure 3: The current through a face with plane normal in z-direction is proportional to the area in the x-y plane which is occupied by the charged box, and which is also within the cellface. This area is the product of piecewise linear functions f_x, f_y and a rectangular window function f_z .

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

For that algorithm, the electric and magnetic field must be known at the same time. The normal FDTD-algorithm does not compute the magnetic field at the same times as the electric field. Instead, the magnetic field is only known half a timestep away. The natural assumption, that the magnetic field at times $n\Delta t$, is the average of the magnetic field at $(n-1/2)\Delta t$ and $(n+1/2)\Delta t$ leads to the computation of the magnetic field in two half steps. One step, to compute the magnetic field at the time when it is needed to compute the LORENTZ-force on the charges, and the other half step to compute the magnetic field at the time $t = (n+1/2)\Delta t$.

$$\vec{H}(n\Delta t) = \vec{H}((n-1/2)\Delta t) - \frac{\Delta t}{2} \frac{1}{\mu} \nabla \times \vec{E}(n\Delta t)$$

$$\vec{H}((n+1/2)\Delta t) = \vec{H}(n\Delta t) - \frac{\Delta t}{2} \frac{1}{\mu} \nabla \times \vec{E}(n\Delta t)$$

Since the magnetic field at times $n\Delta t$ is only required in the vicinity of charges, the algorithm with the two half steps is only needed for subvolumes where charges are in. The charge free subvolumes can be treated with the standard FDTD algorithm.

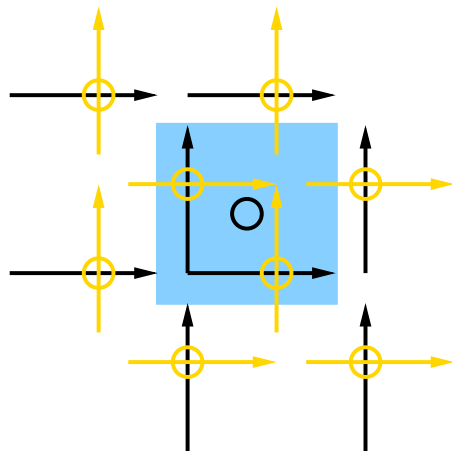


Figure 4: The dark arrows represent the electric field components. The light ones represent the magnetic field components. For integrating the Lorentz-force, the force due to the magnetic field must be known at the same time as the force due to the electric field. The fields at the center of mass of the charge (black circle) is computed by linear interpolation of the fieldcomponents nearest to the position of the center of mass. The circles are at the positions of the fieldcomponents which are involved in computing the x- and y-components.

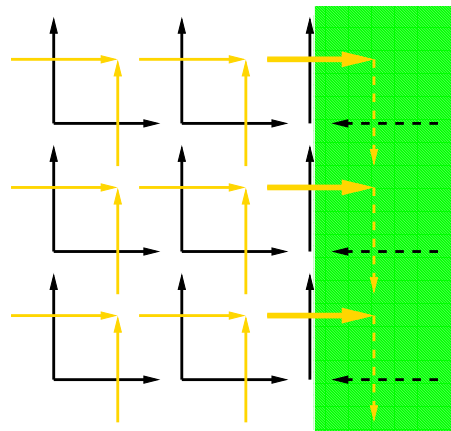


Figure 5: The dark arrows represent the electric field components. The light ones represent the magnetic field components. The shaded area is outside of the computational volume. At the border of the computational volume, magnetic boundary conditions shall be applied. For interpolating the Lorentz-force, the tangential magnetic components outside of the computational volume (dashed light arrows) must be set to the negative value of the corresponding magnetic components just inside. The normal electric components outside (dashed dark arrows) must be set to the negative value of the corresponding electric components inside.

Use of symmetries for particle pushing

The electromagnetic field at the position of a particle is computed by an interpolation of $2 \times 3 \times 8$ nearby field components, see figure 4. This interpolation must be modified for particles which are close to a plane of symmetry, see figure 5. GdfidL uses an auxiliary gridplane. The field-components in this auxiliary gridplane are computed such, that the fields near the plane of symmetry obey the wanted symmetry. For a magnetic wall: The normal electric field component in the auxiliary plane is the negative one of the normal component inside of the computational volume.

Since no current may traverse a magnetic wall, it must be ensured that no net charge traverses the magnetic wall. When, due to the particle pushing, the particle has traversed the symmetry plane, the position of the particle is reflected back into the computational volume. The sign of the normal component of the particles velocity is changed. This is sketched in figure 6.

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>

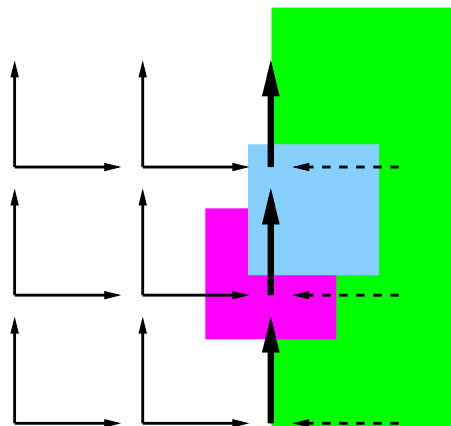


Figure 6: The dark arrows represent the electric field components, or convection currents. The shaded area is outside of the computational volume. The two quadratic patches are a charged box before and after the particle pushing. The upper box is after the pushing. At the border of the computational volume, magnetic boundary conditions shall be applied. Since the box has traversed the magnetic wall due to the particle pushing, its convection current outside of the computational volume must be reflected back into the computational volume. The normal components outside (dashed dark arrows) must be subtracted from the corresponding normal components inside. The tangential convection currents must be doubled (thick dark arrows).