ISBN: 978-3-95450-132-8 doi:10.18429/JACoW-IPAC2014-TUXB01

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2014). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

RECENT PROGRESS IN 3D WAKEPOTENTIAL COMPUTATIONS

W. Bruns, Warner Bruns Feldberechnungen, Berlin, Germany

Abstract

The 3D electromagnetic Field Simulator GdfidL computes Wakepotentials on standard CPUs with a Speed comparable to GPU-Based Implementations. This is achieved via Computing only in interesting Cells, having the FD-Coefficients in compressed Form, traversing the Grid in a Cache-friendly Order and applying a blocked Update Scheme which is NuMA-aware. A Dispersion optimised Scheme is described. Fields in dispersive Materials are computed via solving the Equations of the Electron Hulls of the Material. Moving Mesh Computations have the Grid-generation on the Fly.

DISPERSIVE MATERIALS

Frequency dependent Materialparameters are taken into account by directly simulating the Dynamics of the Electron Hull of Molecules. For each Fieldcomponent in a dispersive Material with N Poles, the Equations of Motion (v Velocity, Q Charge, k Spring-constant, R Damping Term, m Mass of the electron Hull) are solved

$$\begin{aligned}\frac{d}{dt}v_i &= \frac{Q}{m}E - \frac{k}{m}x_i - \frac{R}{m}v_i \\ \frac{d}{dt}x_i &= v_i \\ \frac{d}{dt}E &= \frac{1}{\varepsilon}\nabla\times H - \frac{1}{\varepsilon}\sum_{i=1}^{N} Q_i v_i\end{aligned}$$

PERFORMANCE BECAUSE OF SMALL MEMORY FOOTPRINT

On a current MultiSocket Server, total Cost 7000 Euro, one Timestep in a Grid of 1000 Million Cells, inclusive lossy dispersive Materials, Impedance Boundary Conditions [1] at electric Surfaces and Napoly Integration [2] typically takes less than a Second. The Preprocessing, i.e. the Discretisation of the Materialdistribution and the Computation of the FD-Coefficients typically takes less than an Hour.

Such Performance is achieved because of several Program Design Decisions:

- The Computation of the Material Approximation and FD-Coefficients is **not** separated from the Field Computation. This saves Time for writing to and reading from Files.

- Essentially everything is parallelised using multiple Threads working on a shared Dataset. Not only the Field Update, but also the Material-Approximation and the FD-Coefficients are computed using multiple Threads.

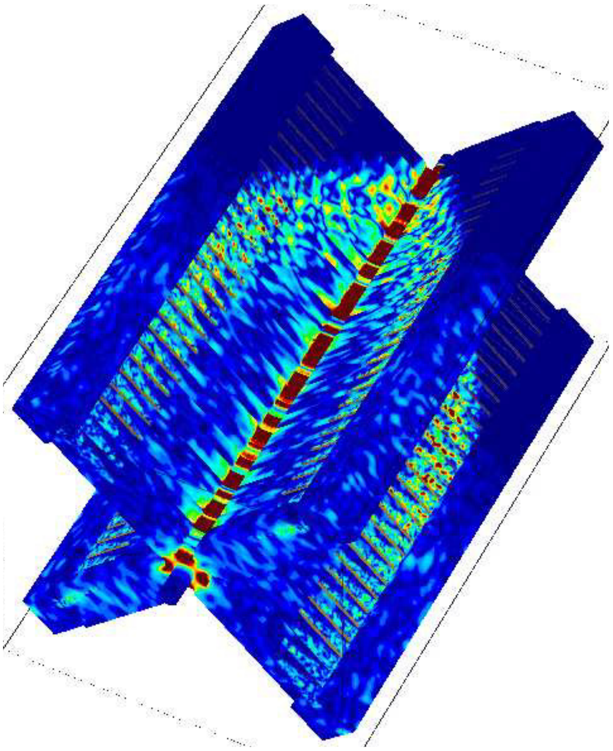

Figure 1: Wall Currents in a 19 Cells CLiC-Section. For this 1000 Million Cells Example, all the Pre-Processing needed before the Time Domain Computation itself starts takes 46 Minutes on a current medium priced Server. The Device is decribed via CAD-Files having 90720 Triangles in it. After performing 4000 Fieldupdates in 30 Minutes, the Wakepotential up to s=0.1 Metres is known. Each additional Metre of Wakepotential needs additional 10000 Fieldupdates computed in 75 Minutes. The Wakepotential up to s=2 Metres then is computed within three Hours, occupying 25 GBytes of RAM. The Device is 0.26 Metres long, the Gridspacing is 185 um. All metallic Surfaces are treated with Impedance boundary Conditions. 5 % of the dielectric Cells are filled with dispersive Material. 16 % of the computational Volume needs to be processed. Napoly Integration is applied. 64 bit floating Point Numbers are used for all Variables. The Times refer to a 4 Socket Opteron 6370P Server, total Cost 7000 Euro, using 32 Cores at 2 GHz. On this particular Server with 256 GByte of RAM, that Device can also be handled with 12000 Million Gridcells. The needed RAM, using 32 bit Field Components, then is 190 GByte. A single Timestep then takes 3.45 Seconds, corresponding to a Gross Cell Update Rate of 12 × $10^9/3.45$ = 3500 MCells per Second. The net Cell Update Rate, as there are only 16 % of the Cells really interesting, still is 560 MCells per Second.

- Only Fieldcomponents are computed that are really time dependent. Cells which are filled by perfectly conducting Materials are ignored and not much Memory is needed to handle them (32 bits per Cell for Grids up to 2 × 10^9 dielectric Cells, 64 bits per Cell for larger Grids).
- The FD-Coefficients are compressed such that the Memory for the FD-Coefficients is just 1/24 of the Memory for the Field Components.
- The Field Update is implemented with Tricks which can be found in the Literature, eg:
  - Non Uniform Memory Access aware Field Access via pinning Field Segments to Threads and Threads to CPU-Cores.
  - Cache aware Field Access via blocked Schemes.
- and then some, which cannot be found in the Literature, eg. the H & E Components are updated not in separate Loops but in a single Loop per Block. This is another Cache-Optimisation.

A Source-Code Fragment which demonstrates how these Tricks might be implemented is below. If a single Thread performs such blocked Update with the Blocks in a proper Sequence, no additional Logic is needed. When multiple Threads are performing such blocked Updates simultaneously, additional Logic for Fieldcomponents at the Borders of the Blocks is needed.

```
   FOR iz IN Block.iz1 .. Block.iz2 LOOP
      FOR iy IN Block.iy1 .. Block.iy2 LOOP
         FOR ix IN Block.ix1 .. Block.ix2 LOOP
            i:= NrofCell(ix,iy,iz);
            IF i>0 THEN                -- Dielectric Cell.
               iType:= CellType(i);    -- Index of the FD-Coefficients of the Cell
               ixp1:= NrofCell(ix+1,iy  ,iz  );  -- Number of Neighbour in +x.
               iyp1:= NrofCell(ix  ,iy+1,iz  );
               izp1:= NrofCell(ix  ,iy  ,iz+1);
Hds(1,i):= Hf1(1,iType) * Hds(1,i)
         - Hf2(1,iType) * ( Eds(3,iyp1) - Eds(3,i) - Eds(2,izp1) + Eds(2,i) );
Hds(2,i):= Hf1(2,iType) * Hds(2,i)
         - Hf2(2,iType) * ( Eds(1,izp1) - Eds(1,i) - Eds(3,ixp1) + Eds(3,i) );
Hds(3,i):= Hf1(3,iType) * Hds(3,i)
         - Hf2(3,iType) * ( Eds(2,ixp1) - Eds(2,i) - Eds(1,iyp1) + Eds(1,i) );

               --
               -- Immediately update the Eds of that very Cell:
               -- If the Blocksize is small enough, all these
               -- Field-Accesses are fulfilled from the Cache.
               --
               ixm1:= NrofCell(ix-1,iy  ,iz  );
               iym1:= NrofCell(ix  ,iy-1,iz  );
               izm1:= NrofCell(ix  ,iy  ,iz-1);
Eds(1,i):= Ef1(1,iType) * Eds(1,i)
         + Ef2(1,iType) * ( Hds(3,i)-Hds(2,i)+Hds(2,izm1)-Hds(3,iym1) );
Eds(2,i):= Ef1(2,iType) * Eds(2,i)
         + Ef2(2,iType) * ( Hds(1,i)-Hds(3,i)+Hds(3,ixm1)-Hds(1,izm1) )
Eds(3,i):= Ef1(3,iType) * Eds(3,i)
         + Ef2(3,iType) * ( Hds(2,i)-Hds(1,i)+Hds(1,iym1)-Hds(2,ixm1) )

            END IF;
         END LOOP;
      END LOOP;
   END LOOP;

 NrofCell : Integer-Array with nx*ny*nz Elements.
 Eds, Hds : Real-Arrays with (3 * Number of dielectric Cells) Elements.
 Celltype : Integer-Array with (Number of dielectric Cells) Elements.
 Hf1, Hf2, Ef1, Ef2 : FD-Coefficients. Real-Arrays with typically
                      less than 30000 Elements.
```

If one wants to have really huge Grids, much larger than 1000 Million Cells, one can use a Cluster of Multi-Core/MultiSocket Machines. On a BNL-Cluster [3] Computations with more than 30 × 10^9 dielectric Cells were performed.

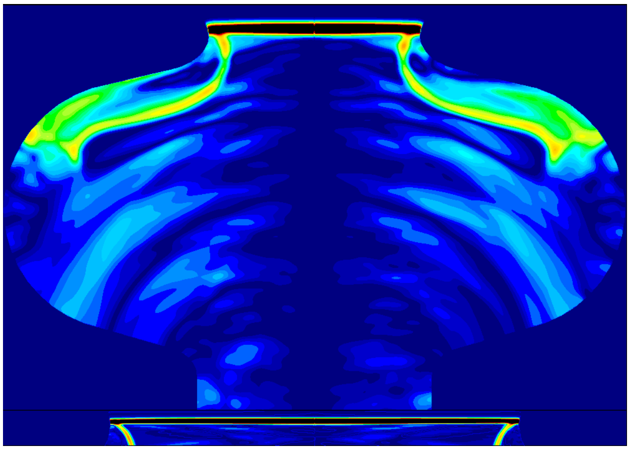

Figure 2: Fieldplots of a Linecharge in a ILC-Module, entering the second Cavity. Above: Sigma=1.3 mm, sHigh= 96 Sigma = 0.125 Metres. dx=dy=dz= Sigma/6 requires 577 Million Cells in the moving computational Volume. Below: In a transversely restricted Volume. Sigma=0.3mm, dz=Sigma/6 = 50 um, sHigh= 12 Sigma = 3.6 mm requires 340 Million Gridcells.

MOVING MESH

When computing in a moving Mesh [4], every few Timesteps a new Gridplane is reached and the FD-Coefficients of this Plane must be known. When using low Dispersion Schemes [5], the Grid sensibly is such that there are in the Order of 200 z-Planes and much more x- and y-Planes. This implies that the FD-Coefficients of a Gridplane are used only during as much Timesteps as it takes until that Gridplane has left the computational Volume. That is then after N x 200 Timesteps when using a Scheme that has a Timestep $c\Delta t = \Delta z/N$. Strang-Splitting wants N=1, Yee wants N=2. The Preparation of of a fresh Gridplane should take less Time than it takes to advance the Field of the whole Grid with 200 Gridplanes, otherwise that Preparation dominates the total Time to get the Result. The Preparation needs to be parallelised as well. This is implemented.

Figure 2 shows the Field in a moving computational Volume. The upper Plot is of a Volume which is ridiculously Large, as there is no Sense in knowing the Wakepotentials in a Range up to 100 Sigma, but not longer. Is there any Charge in the Range 30 Sigma < s < 100 Sigma where one would want to know how it is kicked? The lower Plot is of a Volume with a sensible z-Length. The upper Plot is nicer, though.

Figure 3 shows the Wakepotential of a 200 um Linecharge in a 9 Cavity ILC-Module with Couplers and Bellows. Using Strang-Splitting in a Grid of nx=ny=2924, nz=200 on 32 Cores of a medium priced Server, the Computation takes three Days. 33000 Gridplanes were traversed.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2014). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2014). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

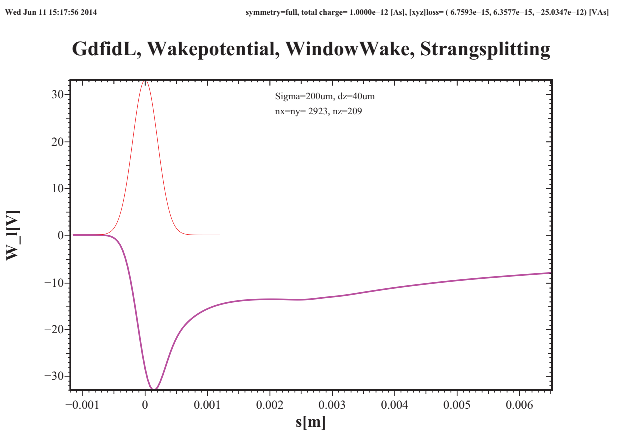

Figure 3: Wakepotential of a 9-Cell ILC-Module with Couplers. Sigma= 200um, dz=Sigma/5, computed with Strang-Splitting.

DISPERSION OPTIMISED FDTD

The first three Terms of the Taylor-Series for $H(t)$ give

$$\begin{aligned} H(t+\Delta t/2) &= H(t-\Delta t/2) + \Delta t \frac{d}{dt} H(t) \\ &\quad + \frac{(\Delta t)^3}{24} \left(\frac{d}{dt}\right)^3 H(t) + \cdots \end{aligned}$$

Maxwells Equations: Replace the Time-Derivatives of E & H by the spatial Derivatives of H and E. A true higher Order FDTD-Scheme then computes the H-Values at a Time $t = (n+1)\Delta t$ via the discretised Form of

$$\begin{aligned} H(n+1/2) &= H(n-1/2) - \Delta t \frac{1}{\mu} \nabla\times E(n) \\ &\quad - \frac{(\Delta t)^3}{24} \frac{1}{\mu} \nabla\times \frac{1}{\varepsilon} \nabla\times \frac{1}{\mu} \nabla\times E(n) \end{aligned}$$

Such a Scheme has a Convergence of fourth Order, if the Curl-Operators are approximated by a Scheme of Order 4. Such higher Order spatial Schemes are problematic at material Discontinuities. If the Curl-Operators are approximated by the standard FDTD-Curl-Operators, the resulting Scheme is still second Order convergent as the Yee-Scheme is, but the Directions in which zero Dispersion occurs may be chosen. The standard Yee-Scheme has zero Dispersion when the highest possible Timestep $c\Delta t = \Delta x \frac{1}{\sqrt{3}}$ is used and the Wave propagates in Direction of a Grid-Diagonal. The dispersion optimised Scheme has very low Dispersion for Waves propagating in x-, y- or z-Direction, when a Timestep of $c\Delta t = \Delta x \frac{1}{4}$ is used and the $(\Delta t)^3 \nabla\times^3$ Term is weighted by 32/51, not 1/24. Because the Scheme is constructed of Curl-Operators, the resulting Fields have no spurious Divergence.

Figure 4 gives the Phase-Error of the Scheme, when it is optimised for low Error at Phi=0. Figure 5 gives the Phase Error, when it is optimised for low Error at Phi=22 Degrees.

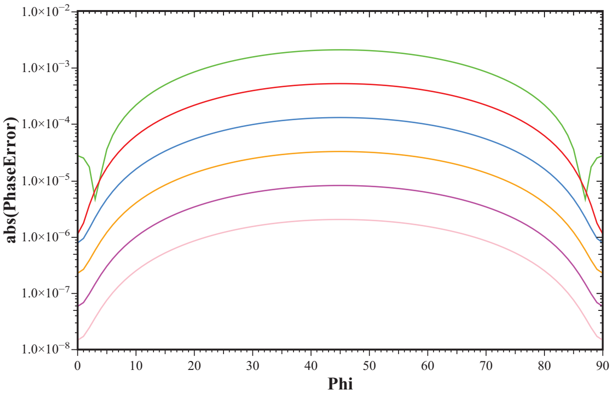

Figure 4: Phase Error of the mFDTD-Scheme, when the Rot3 Term is weighted by 32/51. The different Lines are the Error when the Gridspacing is 1/20, 1/40, 1/80 etc of the Wavelength.

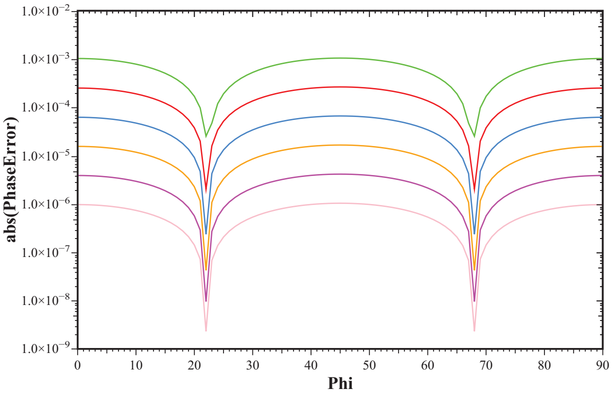

Figure 5: Phase Error of the mFDTD-Scheme, when the Rot3 Term is weighted by 32/69.

REFERENCES

[1] Peter Thoma, ‘Zur numerischen Lösung der Maxwellschen Gleichungen im Zeitbereich’, Darmstädter Dissertation, 1997.

[2] H. Henke, W. Bruns, ‘Calculation of Wake Potentials in general 3D Structures’, Proceedings of the EPAC 2006, Edinburgh, UK, WEPCH110.

[3] A. Blednykh, ‘Impedance Analysis of Insertion Devices’, TWIIICE 2014, Synchrotron SOLEIL, Saint-Aubin, 2014; http://indico.cern.ch/event/277919

[4] K. Bane, T. Weiland, ‘Short range wakepotentials computed in a moving window’, 1983.

[5] T. Lau, E. Gjonaj, T. Weiland, ‘Numerical Dispersion Error Reduction in EM-Calculations for Accelerators’, Proceedings of the 2005 Particle Accelerator Conference, Knoxville, Tennessee, USA.