

GdfidL ON MASSIVE PARALLEL SYSTEMS

W. Bruns, TU-Berlin, Berlin

Abstract

The electromagnetic field solver GdfidL has been extended to also run efficiently on loosely coupled parallel systems, such as massive parallel processors and clusters of workstations. The computational volume is subdivided such that each processor has about the same computational and memory load. This is achieved even for highly irregular geometries.

When computing wakepotentials, GdfidL uses a hollow beam, which excites the same wakefields as a linecharge near the axis, but which travels very near to the inhomogeneties. This reduces the error due to the grid dispersion, especially when computing devices with a low wakepotential, such as tapers.

A new solver to compute lossy eigenvalue problems has been implemented. The solver is capable of finding a whole spectrum of damped resonances in cavities with absorbers, in a single run.

1 PARALLEL COMPUTATION

The Finite Difference Method in cartesian coordinates is easily parallelised, since the subdivision of the total rectangular computational volume is a no-brainer, assuming one restricts oneself to rectangular subvolumes. One just has to partition the grid such, that each processor has about the same number of gridcells. This approach works well, when electromagnetic fields need to be computed in a large fraction of the volume.

However, most realistic RF-devices, if computed in a rectangular volume, do not lead to a grid where most grid-cells are filled with vacuum or a dielectric. The opposite is the case: Complicated devices, for which the computation inherently is time consuming, have an enclosing rectangular box of which 90% or more is free of fields. If one subdivides such a volume into as many subvolumes as there are processors, most processors will run idle since their parts of the volume are uninteresting.

There is a way out: You won't find in the ten commandments, that each processor is limited to working on a single subvolume. If we subdivide the total volume in many more subvolumes than we have processors, we can discard the subvolumes where no fields need to be computed, and spread the remaining ones evenly over the available processors. This approach is halfway between classical Finite Difference Grids and the complicated topology of Finite Element Meshes.

For a typical example, in figure 1 we present a model of a synchrotron cavity.

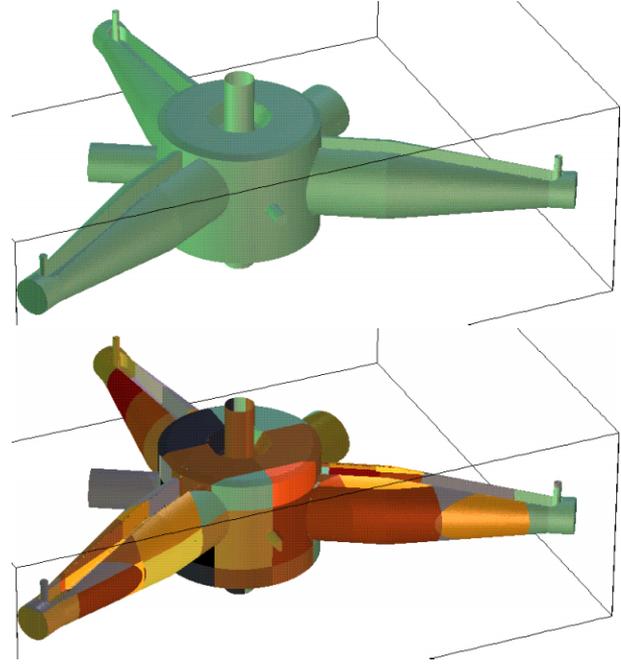


Figure 1: Above: A model of a synchrotron cavity with attached waveguides and tuning plungers. Since the attachments are at different heights, no symmetry plane exists. In this case, less than 10% of the computational volume is filled with vacuum cells. The total number of grid cells used is about 16 millions. Below: The same model with different colours, indicating the used subvolumes. The total volume is subdivided in $8 \times 24 = 192$ subvolumes, of which 122 can be discarded, since they do not have a single vacuum cell. About 3 GBytes of RAM and six hours wall clock time on an eight processor PC Cluster (total cost 8.000 USD) are needed to accurately compute the first 120 resonant fields.

1.1 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 the discretised form of

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

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

and one finds resonant fields in lossfree structures by searching for the eigenvalues of the discretised form of

$$\frac{1}{\epsilon} \nabla \times \frac{1}{\mu} \nabla \times \vec{E} = \omega^2 \vec{E} \quad (3)$$

Most of the CPU-time is spent in applying these discretised curl operators. However, they are quite easily parallelised. For example, when performing a FDTD calculation, the algorithm for each subvolume is:

```

For all Timesteps: D0
  Compute local H by applying the local
  curl operator to the local E
  For all Directions: D0
    Send tangential H to the neighbour
    Receive tangential H from neighbour
  ENDDO For all Directions
  Compute local E
  For all Directions: D0
    Send tangential E to the neighbour
    Receive tangential E from neighbour
  ENDDO For all Directions
ENDDO For all Timesteps
    
```

The tangential H-components at the lower boundaries of the local volumes must be sent to the neighbour volumes in negative directions. Correspondingly, the tangential E-components at the upper boundaries must be sent to the neighbour volumes in positive directions. For correct results, the tangential components from a neighbour in eg. x-direction must be received before data can be sent in eg. y-direction.

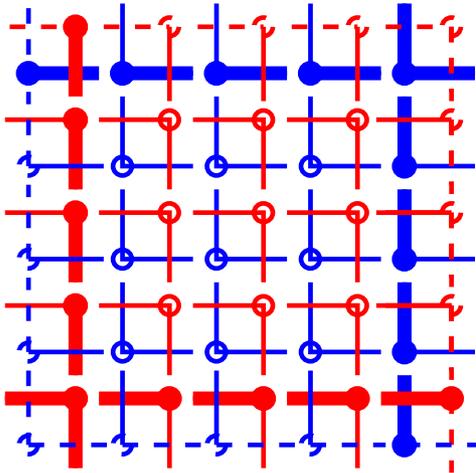


Figure 2: The blue lines and circles represent the electric field components in a local volume. The red ones represent the magnetic field components. The tangential E field components at the upper boundaries of the local volume (thick blue) and the tangential H field components at the lower boundaries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

2 WAKE COMPUTATION WITH A HOLLOW BEAM

Normally, a wakepotential computation for a three dimensional structure is carried out by exciting the fields by discretising the real exciting bunch at its real position, near the axis. Because of dispersion errors, the computed primary field of a relativistic charge will not be exactly orthogonal to the direction of flight [1]. It will somewhat lag behind the charge, giving rise to an effect similar to CHERENKOV-radiation. The lag grows with the distance from the charge.

Because of this lag, wakepotential computations in rotational symmetric structures most often are not performed by exciting with a linecharge at the center, but with a suitable hollow charge traveling at the outermost radius of the beampipe. This is allowed, since in circular symmetric structures, a hollow circular-cylindrical charge excites the same fields as a linecharge, at least outside the cylinder. Since the wakepotentials are dependent only on the fields that are scattered from the inhomogeneities outside of the cylinder, the wakepotential of a hollow charge is the same as the wakepotential of a linecharge. The suitable charge is a single circular sheet charge, independent of the coordinate φ and with the same z-dependence as the real line charge. Since the hollow charge has a smaller distance to the scattering inhomogeneities, the primary field has less chance to spread due to the dispersion errors.

In the general case, when the beampipe cross section is not a circle, a single sheet charge is not sufficient to excite the same wakefields as the original line charge. One needs at least two sheets. These two sheets should nestle to the beampipe, giving the primary field of the sheets the least possible chance to spread erroneously.

Figure 3 illustrates the field of the equivalent sheets in an elliptical like beampipe, figure 4 illustrates an achieved error reduction when computing the wakepotential of a taper.

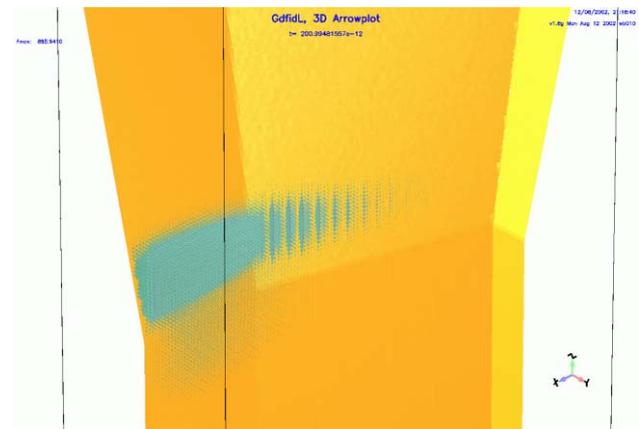


Figure 3: The two sheet charges just entering a taper. The field between the charge and the wall is a better approximation of the real field, than if one would discretise a line charge near the axis. The ratio σ/Δ is 5.

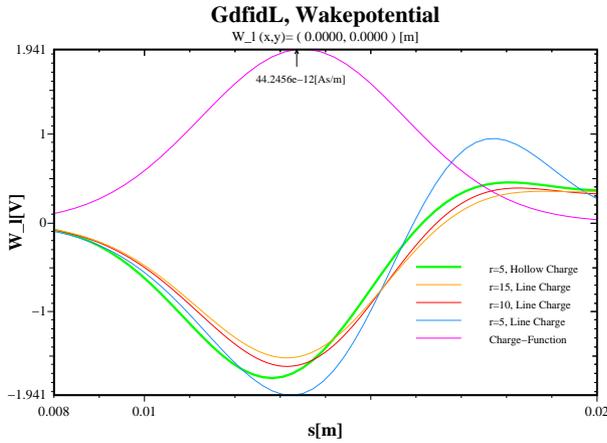


Figure 4: The computed wakepotential of a smooth taper, with an opening angle of 13 degrees. The three thinner curves are the computed short range wakepotentials when the ratio of the charge length to grid-spacing is $\sigma/\Delta = 5, 10, 15$ respectively, and the exciting charge is a line charge on the axis. The amplitude decays when a finer mesh is used. The thick green curve is the computed wakepotential when a hollow beam is used, and the ratio σ/Δ is 5. The quality of that solution is somewhere between the quality of $\sigma/\Delta = 5$ and $\sigma/\Delta = 10$ of the conventional algorithm. Since the memory consumption grows with the third power of σ/Δ , and the needed CPU-time grows with the fourth power, the gain in efficiency is five or more.

3 EIGENVALUES OF DEVICES WITH LOSSY DISPERSIVE MATERIALS

When computing resonant fields in structures with absorbers, one could search for the eigenvalues of the discretised form of

$$\frac{1}{\varepsilon} \nabla \times \frac{1}{\mu} \nabla \times \vec{E} = \omega^2 \vec{E} \quad (4)$$

where now ε and/or μ are complex. The drawback of this is that no efficient algorithm for large numbers of sought eigenvalues is known, when the number of gridcells becomes large. But this is not the only possible eigenvalue equation. For example, one may search for the eigenvalues $e^{j\omega\Delta t}$ of the lossy FDTD-operator. When the losses are described via complex material parameters ε and μ , that eigenvalue equation may be written as

$$\begin{pmatrix} I & -\frac{\Delta t}{\mu} \nabla \times \\ \frac{\Delta t}{\varepsilon} \nabla \times & I - \frac{\Delta t}{\varepsilon} \nabla \times \frac{\Delta t}{\mu} \nabla \times \end{pmatrix} \begin{pmatrix} \vec{H} \\ \vec{E} \end{pmatrix} = e^{j\omega\Delta t} \begin{pmatrix} \vec{H} \\ \vec{E} \end{pmatrix} \quad (5)$$

Because the region in the complex plane where all eigenvalues are located is known a priori to be the inside of the unit circle, the highly efficient SAP-algorithm of Tückmantel[2] [3] can be used almost unchanged. The SAP algorithm basically starts with a set of random vectors and filters out the contribution of the unwanted eigenvectors in these vectors by multiplying with a matrix-polynom.

The roots, the minima, of this polynom have to spread over the region of the unwanted eigenvalues. The only needed modification of the Tückmantel algorithm is finding these complex roots suitable distributed in the region of the unwanted eigenvalues. Since one normally is interested in the first few resonant fields, the ones with lower resonant frequencies, one also knows that these eigenvalues $e^{j\omega\Delta t}$ are very near to the unit circle. Since the real part of ω is known to be small, its imaginary part also cannot be large for resonances with realistic Q-values. The polynom therefore should be small everywhere in the unit-circle, except near the wanted eigenvalues. For a given number of gridcells, the computational load to solve this eigenvalue problem is about ten times the load of the lossless case.

Actually, the matrix multiplication in equation 5 is just one FDTD-step. It is not necessary to implement that matrix times vector operation from scratch, if one already has a routine to perform a single FDTD-step. That routine just has to be rewritten to work on complex vectors instead of real ones. Therefore, since GdfidL can compute time dependent fields in dispersive media, it can also compute eigenvalues in devices with such materials.

4 CONCLUSION

Most capabilities of GdfidL[4], ie. computation of resonant fields in lossfree or lossy, possibly dispersive media, with periodic boundary conditions etc., computation of scattering parameters and wakepotentials, are now also available on high performance massive parallel systems, as well as on less expensive clusters of workstations.

A better wakefield excitation enables more accurate wakepotential computations, especially for devices with small wakepotentials, such as tapers.

An stunningly simple, yet highly efficient algorithm for computing resonant fields in lossy devices is implemented.

5 REFERENCES

- [1] Lloyd N. Trefethen: "Group Velocity in Finite Difference Schemes", *SIAM Review*, Vol. 24, No. 2, April 1982, pp. 113-136
- [2] J. Tückmantel, "Urmel with a High Speed and High Precision Eigenvector Finder", CERN/EF/RF 83-5, 11 July 1983
- [3] J. Tückmantel, "An improved version of the eigenvector processor SAP applied in URMEL", CERN/EF/RF 85-4, 4 July 1985
- [4] <http://www.gdfidl.de>