# A Wavelet-Transformed Finite-Difference Time-Domain Scheme with a static Multi-Resolution Grid

A. Rennings, F. Wittmann, M. Walter, P. Waldow and I. Wolff

Department of Electrical Engineering (ATE), University Duisburg-Essen,
Bismarckstr. 81, 47048 Duisburg, Germany

*Abstract* — **Recently an effective implementation for wavelet transformed operators on three-dimensional data was derived by us. This novel data structure is used to transform a FDTD scheme with a fast WT into a multi-resolution version. Our approach combines the favorable characteristics of a FDTD scheme and the possiblility to resolve the electromagnetic field on locally different resolutions. Since small wavelet coefficients are not updated during the time loop the six field components are computed within such a multi-scale grid instead of the single-resolution Yee grid of FDTD. Finally, resonator structures have been simulated to evaluate the efficiency and applicability of our approach.**

## I. Introduction

Since 1996 quite a lot of research has been done in the area of wavelet based field simulation techniques. The schemes can be categorized into two groups in terms of the general approach. The *multi-resolution time-domain* (MRTD) method, first published by Krumpholz and Katehi [1], is one category. This method usually starts with the coarsest resolution level and adds locally wavelets with higher resolutions. The so-called connection coefficients between the scaling function and all involved wavelets have to be determined before starting the simulation itself. Within the approach presented by Werthen and Wolff [2] (the other category) the connection coefficients $r(i)$ (also called derivative stencil) are set up only for the finest resolution and then a fast *wavelet-transform* (WT) is used to obtain the coefficients for the different scales. Another important issue is the way to determine those coefficients. For the derivative operator, e.g. $\frac{\partial}{\partial y}[\cdot]$, one can either use the scaling function to evaluate $r(i) = \int \phi(y + i\Delta y + 1/2\Delta y)\frac{d}{dy}\phi(y)dy$ (an analytical way to calculate $r(i)$ was presented in [3]), or use the *higher-order* (HO) FDTD stencils directly. We will follow the later technique, since HO FDTD stencils with the same length as the scaling function versions cause a lower dispersion error [4]. As a first step a FDTD for the finest resolution is set up, afterwards a *three-dimensional* (3D) fast WT is applied to all involved operators. Since small wavelet coefficients of the six field components are not updated within the *leapfrog* time loop, the EM field is represented within a grid with locally different rersolutions. One can choose the order of the FDTD scheme independently of the transform filter and vice versa - that is another advantage of our technique.

## II. Effective Implementation of Linear Operators on Three-Dimensional Wavelet Transformed Data

The main issue for our technique was to find an efficient data structure for the involved wavelet transformed operators $\widetilde{\mathbf{A}}[.]$. Let us assume such a operator is applied to a 3D array, $\mathbf{R}$, with wavelet coefficients of a field component, e.g. $E_x(i,j,k)$. The result of this operation is another array, $\widetilde{\mathbf{L}} = \widetilde{\mathbf{A}}[\widetilde{\mathbf{R}}]$ ($R$ and $L$ for right and left hand side). The tilde should indicate the transformed data. Since the operators in Maxwell's equations and the WT itself are both linear, the transformed operator $\widetilde{\mathbf{A}}[\cdot]$ is linear as well. Therefore a wavelet coefficient of the array $\widetilde{\mathbf{L}}$ with a index $\vec{l}$, $\widetilde{L}(\vec{l})$, is a linear combination of some $\widetilde{R}(\vec{r}_n)$, i.e., $\widetilde{L}(\vec{l}) = \sum_n f_n \widetilde{R}(\vec{r}_n)$. In our data structure the sequences $f_n$ and $\vec{r}_n$ are stored for each $\vec{l}$. Since every operator $\widetilde{\mathbf{A}}[\cdot]$ has its own factor and vector lists, they are labeled by $f\{\widetilde{\mathbf{A}}\}(\vec{l})$ and $\vec{r}\{\widetilde{\mathbf{A}}\}(\vec{l})$, respectively. With these lists the above operation, $\widetilde{\mathbf{L}} = \widetilde{\mathbf{A}}[\widetilde{\mathbf{R}}]$, can be implemented in the following mathematical form:

$$\widetilde{L}(\vec{l}) = \sum_n f_n\{\widetilde{\mathbf{A}}\}(\vec{l})\,\widetilde{R}\left(\vec{r}_n\{\widetilde{\mathbf{A}}\}(\vec{l})\right) \quad \forall \vec{l} \qquad (1)$$

Within the computer each operator, $\widetilde{\mathbf{A}}[\cdot]$, consists of two 3D pointer arrays, $f\{\widetilde{\mathbf{A}}\}$ and $\vec{r}\{\widetilde{\mathbf{A}}\}$. $f\{\widetilde{\mathbf{A}}\}(\vec{l})$ points to a sequence with the factors $f_n\{\widetilde{\mathbf{A}}\}(\vec{l})$ to calculate $\widetilde{L}(\vec{l})$. In the other pointer array the element $\vec{r}\{\widetilde{\mathbf{A}}\}(\vec{l})$ points to a list of vector indices, $\vec{r}_n\{\widetilde{\mathbf{A}}\}(\vec{l})$. These are the indices of the used coefficients of the 3D array, $\mathbf{R}$. In Fig. 1 the data structure $f\{\widetilde{\mathbf{A}}\}$ is



0-7803-7695-1/03/$17.00 © 2003 IEEE

depicted. The other structure for the vector indices $\vec{r}\{\tilde{\mathbf{A}}\}$ of $\tilde{\mathbf{R}}$ is similar.

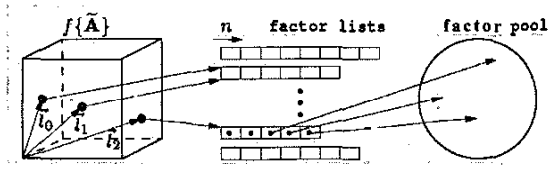


Fig. 1. Data structure to implement a linear operator $\tilde{\mathbf{A}}$

## III. Wavelet based Sub-Gridding Scheme

The cross section of the microstrip line in Fig. 2 is used to explain the wavelet based sub-gridding scheme. In the upper part of the figure a typical FDTD discretization is displayed. To resolve the field components properly the user of a FDTD simulator should vary the resolution as depicted. Restrictively, for FDTD it is only possible to vary Ax along the x-direction, $\Delta y$ along the y-direction and so on - that has a few disadvantages. For our example, the field is oversampled (in x-direction) within the ellipses in Fig. 2 (top). Another issue is that in some areas (especially in the ellipses) the difference between Ax and $\Delta y$ is relatively large, i.e., the derivative in one direction is approximated very well, but not in the other. The discretization in the lower part of Fig. 2 is the favored one, since the above mentioned problems do not occur here. In general, our approach can handle an arbitrary number, $J$, of resolution levels ($J = 3$ in our example). The areas with different resolutions are defined in a so-called *resolution last*. It has the following form $\Omega_{res} = \{\Omega^1, \cdots, \Omega^{J-1}\} = \{\{\Omega_1^1, \cdots, \Omega_{N_1}^1\}, \cdots, \{\Omega_1^{J-1}, \cdots, \Omega_{N_{J-1}}^{J-1}\}\}$. The upper index labels the resolution level of the particular area. $\Omega_{res} = \{\Omega^1, \Omega^2\} = \{\{\Omega_1^1, \Omega_2^1\}, \{\Omega_1^2, \Omega_2^2, \Omega_3^2\}\}$ is the *resolution list* for our example. The two black squares with the finest resolution at the edges of the strip in the upper part of Fig. 3 are named $\Omega_1^1 \cup \Omega_2^1 = \Omega^1$ in the list. For the intermediate resolution there are three members colored in dark grey. The light grey colored area does not need to be defined, since it is the rest of the whole computational domain. This area is discretized roughly, due to the smooth field distribution. In the lower part of Fig. 3 the corresponding wavelet coefficients of the areas, $\Omega_k^j$ are depicted. These coefficients are elements of the sets $\tilde{\Omega}^j$. $\tilde{\Omega}^j$ contains all the wavelet coefficients of the resolution level $j$ ($j = 1$ is the finest resolution and $j = 3$ is the coarsest level). The coefficients to be updated

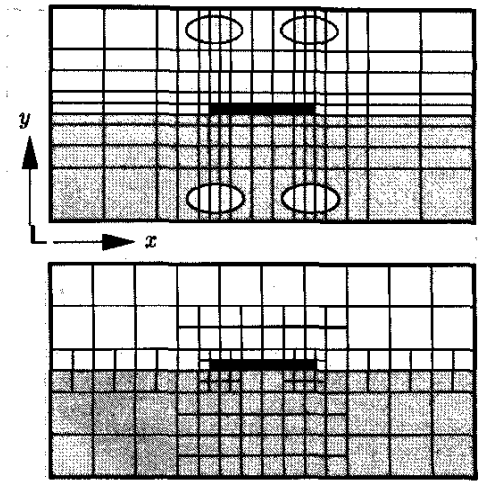


Fig. 2. Typical FDTD discretization (top) and favored grid

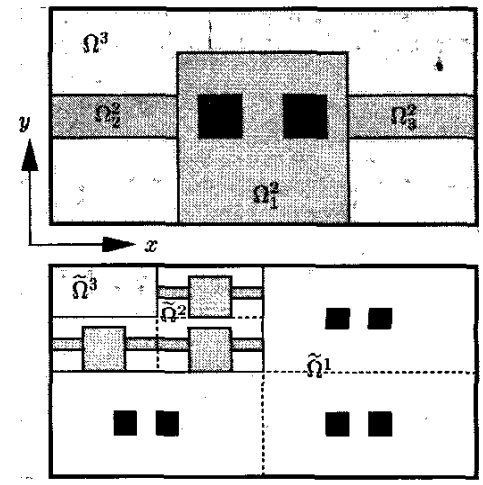


Fig. 3. Areas $\Omega_k^j$ with different resolutions, $j$, (top) and corresponding wavelet coefficients in the sets $\tilde{\Omega}^j$ (bottom)

are called *active* and they are defined in the so-called *active coefficient list*, $\tilde{\Omega}_{act} = \{\tilde{\Omega}_{act}^1, \cdots, \tilde{\Omega}_{act}^{J-1}, \tilde{\Omega}^J\} = \{\{\tilde{\Omega}_1^1, \cdots, \tilde{\Omega}_{M_1}^1\}, \ldots, \{\tilde{\Omega}_1^{J-1}, \ldots, \tilde{\Omega}_{M_{J-1}}^{J-1}\}, \tilde{\Omega}^J\}$. $\Omega_{act}^j$ is usually a small subset of the whole $\tilde{\Omega}^j$ (expecially for the fine resolution levels). The ratio of the number of elements in the active coefficients list to the number of all grid points is named *grid ratio* $\tilde{\omega}_{act}$. For our example the active–grid list has the form: $\tilde{\Omega}_{act} = \{\tilde{\Omega}_{act}^1, \tilde{\Omega}_{act}^2, \tilde{\Omega}^3\} = \{\{\Omega_1^1, \ldots, \Omega_6^1\}, \{\tilde{\Omega}_1^2, \ldots, \tilde{\Omega}_9^2\}, \tilde{\Omega}^3\}$. The coefficients of $\tilde{\Omega}^3$ belong to the coarsest resolution level - the base discretization. All coefficients of

this scale are in $\widetilde{\Omega}_{act}$ because at least this (base) discretization is needed in the whole computation domain without any refinement. The rest of the list is determined by an algorithm that finds out which wavelet coefficient in $\widetilde{\Omega}^j$ needs to be updated to get a resolution $j$ in the areas $\Omega_k^j$. If only the coefficients in $\widetilde{\Omega}_{act} = \{\widetilde{\Omega}^1_{act}, \widetilde{\Omega}^2_{act}, \widetilde{\Omega}^3\}$ are updated within the *leapfrog* time-stepping scheme, the electromagnetic field is calculated in the multi-resolution grid depicted in Fig. 2 (bottom), according to our objective.

## IV. Simulations

The algorithms mentioned in the previous sections have been implemented in the programming language *PYTHON* (the numerical Python package is extensively used). During a preprocessing all operators (derivative/material) are built up in the non-transformed domain with the data the simulator gets from the user via the GUI. The user can choose between FDTD stencils from 2nd up to 6th order for the spatial derivatives. For the longer stencils the mirror principle is used to handle PECs and PMCs at the boundaries or inside the computational domain. Depending on the number of resolution levels, $J$, a fast WT with $J-1$ transformation steps is used to transform all involved operators. Since only PEC and PMC boundaries are implemented yet, resonator structures have been simulated to determine their resonant frequencies. The duration of the time domain simulations was always 10.000 $At$. The largest possible Courant number for a 2nd order FDTD instead of the lower one for MRTD was used in all simulations - except for the simulations with a 4th order accurate stencil. Here $At$ was multiplied with 6/7, which yields the maximum possible time step for 4th order schemes. In this paper the results of two cavities are presented. The first one is an air-filled cavity with an dielectric block inside as depicted in Fig.4. The field component $E_x\,(i = h = 2, j, k)$ in Fig. 5 has been calculated within a multi-resolution grid with two different levels. The smooth surface plot at the interface air/dielectric indicates an area with a fine resolution. The rest is computed with the coarse resolution - thus the surface plot is *stepped*. In this case a Haar wavelet has been used for the WT and a 2nd order stencil for the derivative. In table I the results for the dominant resonant frequency of the above resonator computed with different stencil orders and transform wavelets are summarized. In the 1st and 2nd column the frequency has been determined by a 2nd and 4th order FDTD in an overall fine grid. In the 3rd and 4th column the result has been calculated within a multi-resolution grid transformed with the Haar respectively the Daubechies wavelet $D_2$. The frequency in the 4th case is exactly the same than in the fine grid, but just 66% of all wavelet coefficients have been updated.

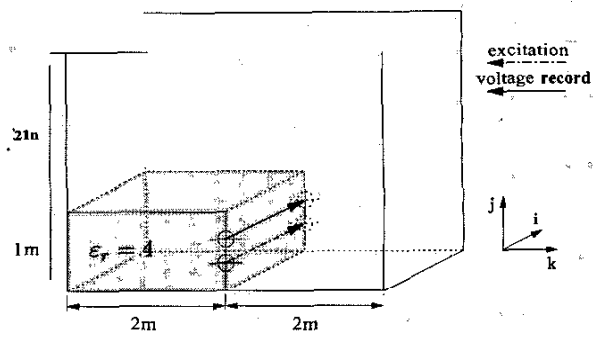


Fig. 4. Resonator with dielectric block ($\varepsilon_r = 4$)

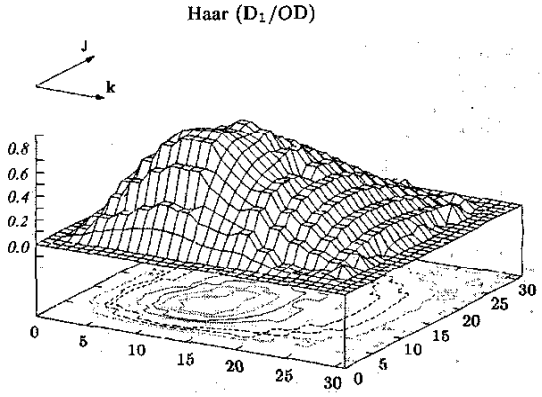


Fig. 5. $E_x\,(h\,j,k)$ computed in the multi-resolution grid

| diel.block | $FDTD_2$ | $FDTD_4$ | *Haar* | *Daub* |
|---|---|---|---|---|
| $f_0$ [MHz] | 50.0795 | 50.8053 | 51.3859 | 50.0795 |
| $\widetilde{\omega}_{act}$ [%] | 100.00 | 100.00 | 53.15 | 65.71 |

Table I. Resonant frequency of cavity w/ dielectric block

The 2nd resonator was an air-filled one with a metal plate inside as depicted in Fig. 6. Additionally, the fine grid is displayed with a dark grey area in the figure. These areas are located around the plate (especially at its edge) and at the PEC boundaries, since the field distribution has a sharp bend (high resolution wavelets are needed to represent that). The rest in light grey is discretized with the coarse grid, due to a smooth field. The field component $E_x(h, j, k)$ computed in the multi-resolution grid transformed with

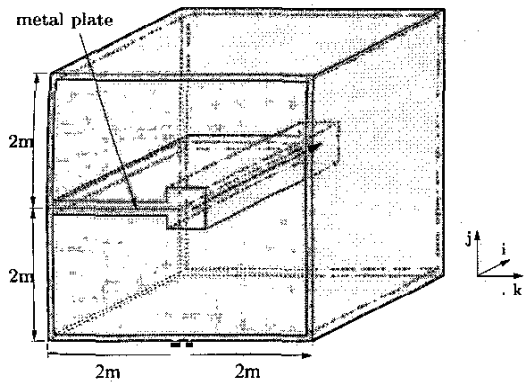


Fig. 6. Cavity w/ metal plate, areas w/ fine resolution in dark grey

Haar wavelets is plotted in Fig. 7. As in the first example the areas with a fine resolution are represented by a smooth field plot in the vicinity of the plate and at the PEC–boundaries. The corresponding wavelet coefficients $E_x(h,j,k)$ have been depicted in Fig. 8. All coefficients in the coarse scale are needed for the base discretization. Non-zero coefficients in the higher resolution levels are located around the plate, especially at its edge, and at the PEC boundaries. The here not presented field plots for the FDTD 2nd order and the Daubechies case are similar to each other. Therefore the results for the resonant frequency given in table II are exactly the same. The value for the Haar wavelet is slightly lower. During our simulations

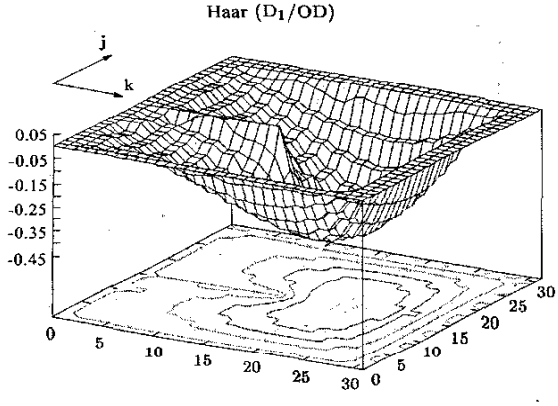


Fig. 7. $E_x(h,j,k)$ computed in the multi-resolution grid

| met. plate | $FDTD_2$ | *Haar* | *Daub* |
|---|---|---|---|
| $f_0$ [MHz] | 135.2142 | 133.0334 | 135.2142 |
| $\omega_{act}$ [%] | 100.00 | 49.05 | 62.72 |

Table II Resonant frequency of the cavity w/ metal plate

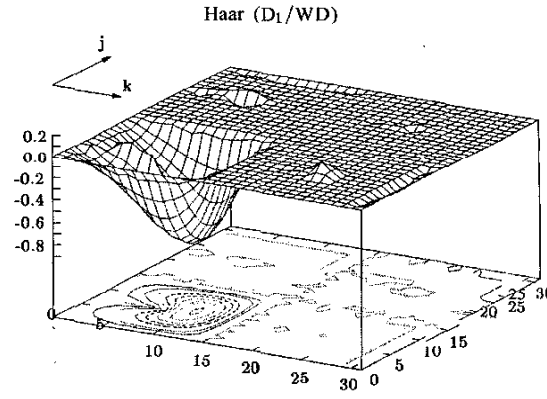


Fig. 8. Wavelet coefficients $\widetilde{E}_x(h,j,k)$ of the above depicted field component $E_x(h,j,k)$

we made the experience that the multi-resolution grid generated by Daubechies wavelets $D_2$ yields very good results in terms of the field distribution compared to FDTD (fine grid) - although only 60% of all coefficients are updated. Whereas with the Haar wavelet the field distribution is just a good approximation and therefore the frequencies were slightly different.

## V. Conclusions

A FDTD scheme with a multi-resolution grid was presented in this paper. The approach was based on a novel data structure for operators on 3D field components. The fact that there is no explicit subgridding algorithm is the main advantage of our technique. The WT and inverse WT operators are transforming the FDTD scheme to a multi-resolution version, i.e., the complicated algorithm is implicitly done by them.

## References

[1] M. Krumpholz and L. P. B. Katehi, "MRTD: New time domain schemes based on multiresolution analysis", in *IEEE Trans. Microwave Theory Tech.*, vol. 44, pp. 555–561, April 1996.

[2] M. Werthen and I. Wolff, "A novel wavelet based time domain simulation approach", in *IEEE Microwave Guided Lett.*, vol. 6, pp. 438–440, December 1996.

[3] A. Rennings, S. Otto, M. Walter, P. Waldow and I. Wolff, "The representation of derivative operators in Yee's staggered grid in bases of compactly supported Daubechies wavelets", in 2002 *European Microwave Conference Proc., Milan, Italy*, vol. 1, pp. 355–356, September 2002.

[4] K. L. Shlager and J. B. Schneider, "A comparison of the dispersion error of higher-order finite-difference time-domain schemes with Daubechies' multi-resolution time-domain scheme", in *Proc. 2001 IEEE AP-S Int. Symp., Boston, MA*, vol. 1, pp. 72–75, July 2001.