Fast and Accurate Evaluation of Enclosures With the Method of Moments by Using Splay Trees

Femke Olyslager ^{#1}, Jan Fostier ^{#2}, Joris Peeters ^{#3}, Daniël De Zutter ^{#4}

[#]*Department of Information Technology, Ghent University
Sint-Pietersnieuwstraat 41, 9000 Ghent, Belgium*

¹femke.olyslager@intec.ugent.be

²jan.fostier@intec.ugent.be

³joris.peeters@intec.ugent.be

⁴daniel.dezutter@intec.ugent.be

Abstract—In this contribution we extend our work on the application of boundary integral equations to accurately predict the shielding properties of complex enclosure geometries. The focus of this contribution is on the power of the splay tree algorithm to efficiently extract symmetry properties from a geometry. We will show that by using these splay trees in two as well as three dimensions significant savings in CPU time can be achieved allowing for the evaluation of ever more complex structures within a given computation time. We show different results for enclosures in two dimensions and also illustrate the time savings of splay trees in three dimensions. At the time of the conference we will also show three-dimensional shielding problems.

I. INTRODUCTION

The shielding efficiency of metallic enclosures has been the subject of many studies, see e.g. [1] for an overview. Recently we have shown [2] that advanced Method of Moments (MoM) techniques allow for more detailed simulations of more complicated enclosures at higher frequencies. In [2] a boundary integral equation was presented that was discretised using a Galerkin MoM. If the number of unknowns in the MoM is N then a MultiLevel Fast Multipole Algorithm (MLFMA) allows to reduce the CPU-time and memory requirements to $O(N \log N)$.

In [2] the importance of an accurate evaluation of the near interactions in the MoM matrix was outlined. It was also explained that further acceleration was possible by using Singular Value Decompositions (SVD) for the near interactions and that in the iterative solution process the usage of a block Jacobi preconditioner was very beneficial. In this contribution we want to focus on another aspect that is very important for accelerating the solution process. Due to symmetries in the geometry often many elements of the MoM matrix will be equal. However, the problem is to extract these symmetries. For N unknowns there are N^2 interactions (= elements in the MoM matrix) requiring N^4 CPU-time to extract symmetry when a brute force method is used. The splay tree algorithm [3] is an advanced algorithm that allows symmetry extraction in $O(N \log N)$ CPU-time making it compatible for use with the MLFMA algorithm.

We refer the reader to [2], [4] and [5] for an explanation of the MoM technique for two-dimensional TM-scattering as well

as for the MLFMA acceleration, the SVD, the preconditioner and the accurate evaluation of the near interactions. In the next section we will briefly explain the splay tree algorithm. We will demonstrate the performance of splay trees for two-dimensional and three-dimensional scattering. For the three-dimensional problems a Combined Field Integral Equation (CFIE) is used discretised with Rao-Wilton-Glisson basis functions [6]. The resulting MoM matrix was again solved iteratively. Then we will show a number of two-dimensional shielding problems. Three-dimensional shielding problems are not yet shown but at the time of the conference we will also show shielding efficiencies of three-dimensional enclosures. Although the numerical examples are limited to lossless structures the method works as well for structures including lossy materials.

II. NEAR INTERACTIONS - SPLAY TREES

Often large structures contain symmetries where two pairs of interacting segments are geometrically equal. This obviously means that the corresponding two elements in the MoM matrix are equal. Computing time during the setup stage can be saved when these symmetries are recognized, because the corresponding interactions only need to be calculated once. If one has N segments then there are N^2 interactions. A brute force method comparing all these interactions with each other would lead to $O(N^4)$ computing time, which obviously would jeopardize the whole algorithm. Even if one restricts symmetry extraction to the near interactions this would still lead to $O(N^2)$ computing time overwhelming the $O(N \log N)$ complexity of the MLFMA. To avoid this we extract symmetry for the near interactions using a splay tree [3]. A splay tree is a special kind of self-balancing binary tree with an amortized complexity of $O(\log N)$ for store and search operations. In contrary to other self-organizing data structures such as the red-black tree or the AVL tree, it does not require any additional metadata to keep track of its balancing, which makes it memory efficient. Furthermore, frequently accessed elements are stored closer to the tree root, making it very fruitful for implementing caches.

Whenever an interaction between two segments needs to be evaluated a search operation is performed on the splay tree.

If the interaction is not yet stored in the tree, it is evaluated and stored so it can be reused for future geometrically equal interactions. Because the time needed for a search or store operation is much smaller than the time needed for numerical integration over the segments, this yields an efficient way of working. This is illustrated in Table I where the setup time for the near interactions for the TM-scattering at several simple shapes is given. Each shape is modelled with approximately 500 unknowns. Even for a hyperbolic-plano lens, where most interactions are unique, we see some savings, indicating that the overhead imposed by unsuccessful search operations is limited. Note that in the situation where all interactions were unique, the time complexity for building a splay tree would be $O(N \log N)$.

TABLE I
COMPARISON BETWEEN NEAR INTERACTION SETUP TIMES WITH AND WITHOUT SPLAY TREE (ST)

Case	Unique interactions	No splay tree	With splay tree
Circle	0.38%	41.23s	0.57s
Square	10.92%	64.10s	7.78s
Hyp. lens	86.38%	61.03s	52.93s

The splay tree will require somewhat more memory, but this memory needs to be allocated only temporary, after which it can be reused in the MLFMA.

As another example of the performance of splay trees we consider the scattering at two perfectly conducting tori. For a wavelength $\lambda = 2\pi$ m we first consider a torus with radii $R = 3.5$ m and $r = 1.5$ m respectively (see Fig. 1). The torus is illuminated by a plane wave incident in the plane of the torus and with the magnetic field perpendicular to that plane. $N = 2165$ unknowns were needed for the discretisation of the surface current density. Using the splay tree this problem was solved in 619 seconds using 52 iterations. There were only 1.8% unique interactions. Without the splay tree the simulation takes about 8 hours of CPU-time. Fig. 1 shows the amplitude of the induced surface current density by the incident plane wave.

Finally Fig. 2 shows the amplitude of the surface current density on another torus with $R = 8.0$ m and $r = 0.5$ m under the same excitation. In this case there are $N = 2121$ unknowns and 549 seconds of CPU-time were needed for the setup time and 61 iterations.

III. NUMERICAL EXAMPLES

Consider a square perfectly conducting enclosure with sizes $50 \text{ cm} \times 50 \text{ cm}$ filled with 10 equidistant dielectric slabs as shown on Fig. 3. The relative permittivity of the slabs is $\epsilon_r = 4$ and the slabs have a thickness of 4 mm and a length of 40 cm. The distance between the first slab and the front is 6 cm and between the last slab and the back 8 cm. The measure point is 4 cm from the back of the enclosure and the source point is 50 cm in front of the enclosure. The shielding effectiveness is shown in Fig. 4 (Note that the corresponding result in [2] was erroneous.). The number of unknowns at the highest

Fig. 1. Amplitude of the induced surface current density of a thick torus.

Fig. 2. Amplitude of the induced surface current density of a thin torus.

Fig. 3. Enclosure filled with dielectric slabs.

Fig. 4. Shielding effectiveness for the structure of Fig. 3.

Fig. 5. Electric field density inside the enclosure for the structure of Fig. 3 at 2GHz.

frequency was 3400. Using the block Jacobi preconditioner, where the matrices describing the empty enclosure and each of the inserted planes were inverted, the number of iterations in the iterative solution at the highest frequency was only 72. The Bicgstab iterative solution algorithm was used. One iteration took about 10 ms on a standard 3 GHz personal computer. The total memory requirements were only 12 MByte. Fig. 5 and Fig. 6 show the electric field density inside the enclosure at 2 GHz and 4 GHz respectively for the configuration of Fig. 3. In [2] also the shielding effectiveness of the enclosure filled with perfectly conducting metal planes is presented.

In the second example we consider the structure of Fig. 7 where the same enclosure is filled with objects resembling a personal computer tower. The power source, the CDROM and the two hard disks all are perfectly conducting. There are

Fig. 6. Electric field density inside the enclosure for the structure of Fig. 3 at 4GHz.

Fig. 7. Model for a personal computer tower.

three printed circuit boards (PCBs) represented as dielectric slabs with relative permittivity of 4. In the aperture there is a dielectric screen with a relative permittivity of 2. Fig. 8 shows the shielding effectiveness over a wide frequency range.

The number of unknowns at 4 GHz is 2994. At 4GHz the setup time without using the SVD is 11.84 s and with the SVD 12.11 s, the solution time (only the iterations) is 9.65 s and 7.32 s respectively for 506 iterations of the TFQMR iterative solution algorithm. Hence, the loss in CPU-time at setup is highly compensated in the iteration process. The memory requirements are reduced by 48.28% when using the SVD. The effect of the preconditioner is dramatic, it reduces at 1 GHz the number of iterations from 1102 to 45 and at 4 GHz from 370584 (useless to use an iterative solver in this

Fig. 8. Shielding effectiveness for the structure of Fig. 7.

Fig. 9. Electric field density inside the enclosure for the structure of Fig. 7 at 2GHz.

case) to 506 iterations. The effect of the splay tree technique is also significant, at 1 GHz the setup time reduces from 10.52 s to 6.99 s and at 4 GHz from 61.43 s to 12.11 s. However, the memory usage increases significantly from 8 MByte to 72 MByte at 1 GHz and from 22 MByte to 132 MByte at 4 GHz. All these results were obtained on an Opteron 270 running at 2 GHz.

Figure 9 and Fig. 10 show the electric field density inside the structure of Fig. 7 at 2 and 4 GHz respectively. Note the surface waves that propagate in the PCBs and the coupling between these surface waves across the three parallel PCBs in Fig. 10.

Fig. 10. Electric field density inside the enclosure for the structure of Fig. 7 at 4GHz.

IV. CONCLUSIONS

We have shown that the usage of splay trees can drastically accelerate the evaluation of the near interactions in the MoM for two- and three-dimensional scattering problems. We have shown shielding efficiency results for filled enclosures in a two-dimensional TM scattering setting. At the time of the conference we will fully demonstrate the power of the MoM to accurately simulate three-dimensional shielding enclosures with complex geometries up to very high frequencies.

ACKNOWLEDGMENTS

The work of J. Fostier was supported by a doctoral grant from the Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT-Vlaanderen).

REFERENCES

- [1] F. Olyslager, E. Laermans, D. De Zutter, S. Criel, R. De Smedt, N. Lietaert, and A. De Clercq, "Numerical and experimental study of the shielding effectiveness of a metallic enclosure," *IEEE Trans. Electromagn. Compat.*, vol. EMC-41, pp. 202–213, Aug. 1999.
- [2] F. Olyslager and J. Fostier, "Fast and accurate evaluation of the shielding effectiveness of complex enclosures," in *17th Int. Zurich Symp. on Electromagnet. Compat.*, Singapore, Feb. 2006.
- [3] Sleator and Tarjan, "Self-adjusting binary search trees," *JACM*, Vol. 32, pp. 652–686, July 1985.
- [4] J. Fostier and F. Olyslager, "A parallel multilevel fast multipole algorithm for GRID computing allowing optical full-wave simulations," in *Int. Symp. on Antennas and Propagat.*, Singapore, Nov. 2006.
- [5] J. Fostier and F. Olyslager, "A GRID Computer Implementation of the Multilevel Fast Multipole Algorithm for Full-Wave Analysis of Optical Devices," conditionally accepted for *IEICE Transactions on Communication*.
- [6] W. C. Chew, J. M. Jin, E. Michielssen, and J. Song, *Fast and Efficient Algorithms in Computational Electromagnetics*, Boston: Artech House, 2001.