# Solving Moment Method Problems by Partitioning

Jürgen v. Hagen, Alexander Herschlein, Werner Wiesbeck

Institut für Höchstfrequenztechnik und Elektronik
Universität Karlsruhe, Kaiserstr. 12, 76128 Karlsruhe
Federal Republic of Germany

## 1 Introduction

Efficient solutions of electromagnetic field problems are more and more needed as numerical methods are increasingly applied. Many numerical methods require - even with higly sophisticated computers - considerable computational resources and time. Hence, efficient methods are sought for which would extend the range of applicability of present day methods.

One of the major methods to compute scattering and radiation from arbitrary structures is the Method of Moments [1]. It uses a linear system of equations with $n$ unknowns related to the unknown currents. Unlike the setting up of the linear system with a computational cost of $\mathcal{O}(n^2)$, the solving of the linear system in the Method of Moments is quite costly: a general $LU$ factorization has a computational cost proportional to $\mathcal{O}(n^3)$. For larger problems this becomes prohibitive. A crucial point to extend the range of applicability is to improve the solution of the linear system.

A partitioning of the Method of Moments matrix allows to efficiently apply the Method of Moments to several special classes of problems. The first class is an optimization problem where small parts of a big system have to be tuned, whereas the bigger part remains constant. The second class are problems too big to fit into the computer's memory. There, the partitioning allows to efficiently use internal and external memories and to minimize data transfer between the two.

After the outline of the method two numerical examples are presented. The first is the optimization of an antenna on top of a handset, the second is a very large electromagnetic problem which does not fit into the computers memory. A conclusion frames the presentation.

## 2 Exemplary Problems

The two exemplary problems are as follows:

The first is a handset of a mobile telephone or wireless communication system. The outer shape of the handset is fixed by the requirements of volume, shape and design. The antenna on top of the handset may be modified to use monopoles, forked monopoles or helix antennas. The discretization of the handset can be fixed once for all computations, the unknowns on the antenna change with the actual antenna computed. The number of unknowns on the handset is usually much larger as on the antenna as vectorial surface elements are needed on the former, on the latter filaments are sufficient.

The second problem is a problem with 2923 unknown of an air-plane. 2923 unknowns require at least 130 MB of memory which may be more than a given computer system has installed.

0-7803-6369-8/00/$10.00 ©2000 IEEE

## 3 The Partitioning Method

Consider the Method of Moments equation

$$Z\,I = V \tag{1}$$

with $Z$ being the $n \times n$ Moment matrix, $I$ the $n$-vector with the unknown currents and $V$ the $n$ known excitations.

For the first problem the matrix consists of four submatrices. $Z_{11}$ is the interaction of the currents on the handset $I_1$. $Z_{12}$ models the action of the currents on the antenna $I_2$ onto the currents on the handset, $Z_{21}$ the opposite. $Z_{22}$ finally is the interaction matrix of the currents on the antenna. Of these blocks, $Z_{11}$ is the biggest and remains the same for all antennas attached to the handset. The number of unknowns on the handset $n_1$ is usually much larger as $n_2$. The Method of Moments equations is as follows

$$\begin{pmatrix} Z_{11} & Z_{12} \\ Z_{21} & Z_{22} \end{pmatrix} \begin{pmatrix} I_1 \\ I_2 \end{pmatrix} = \begin{pmatrix} V_1 \\ V_2 \end{pmatrix} \tag{2}$$

By using matrix equivalences, $I_1$ and $I_2$ can be solved for using the following equations.

$$I_1 = Z_{11}^{-1} V_1 - Z_{11}^{-1} Z_{12} I_2 \tag{3}$$

$$I_2 = \left(Z_{22} - Z_{21} Z_{11}^{-1} Z_{12}\right)^{-1} \left(V_2 - Z_{21} Z_{11}^{-1} V_1\right) \tag{4}$$

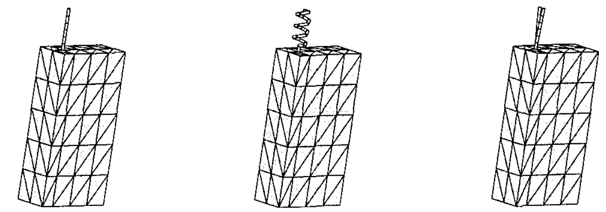
All inversions should not be seen as inversions of a matrix, but rather as the solution of a linear system of equations with several right hand sides. Cai [2] used a somewhat similar approach for a Boundary Element Method, however eleminated all subdiagonal blocks, thereby lengthing the approach.

In the equations (3) and (4), the inversion (or better the $LU$ factorization) of $Z_{11}$ is the most costly. However, it needs to be carried out only once for all computations. Also the multiplication of $Z_{11}^{-1} V_1$ is needed two times, but can be carried out once and the result stored in place.

The startup cost to use the above equation is $\mathcal{O}(n_1^3/3) + \mathcal{O}(n_1^2)$ to compute the $LU$ decomposition of $Z_{11}$ and the solution for $V_1$. The solution step for $I_2$ has a computational cost of $\mathcal{O}(n_2^3/3) + \mathcal{O}\left(n_2^2(n_1+2)\right) + \mathcal{O}(n_1^2 n_2) + \mathcal{O}(n_1 n_2)$. Finally the cost to obtain $I_1$ is only $\mathcal{O}(n_1^2) + \mathcal{O}\left(n_1^2(n_2+1)\right) + \mathcal{O}(n_2^2)$.

The longest part in the above steps is the computation of $Z_{11}^{-1}$, the remaining steps are much faster. For systems with $n_2$ much smaller that $n_1$, the use of equations (3) and (4) is faster than the use of equation (1) and to decompose it with a cost of $\mathcal{O}\left((n_1+n_2)^3/3\right)$.

For our second application, the number $n$ might be so large so that the Moments Matrix $Z$ might not fit into the computer's main memory (of course, it is always possible to use bigger main memories. However this only postpones the onset of the mentioned problem. Another possibility may be to use so-called fast methods [3] possibly achieving improvements on very big problems in excess of several tens of thousands unknowns). The matrix $Z$ is partitioned and the system is solved using equations (3) and (4). During each stage, the submatrices are stored in the external mass memory, only the active matrices used for computations are in the internal RAM memory. The sub-system sizes $n_1$ and $n_2$ are chosen in such a way that the data transfer rates used to store and retrieve the matrices are optimized.



(a) Monopole Antenna (b) Helix Antenna (c) Forked Monopole Antenna

Figure 1: Handset with Three Different Antennas

## 4 Numerical Examples

The first example comprises three antennas on a handset, Fig. 1. The handset has a size of 60 mm by 25 mm by 110 mm. The discretization using FEKO [4] resulted in 228 unknowns for the handset. The different added antennas added 9 unknowns for the monopole, 21 for the helix, and 18 for the forked monopole respectively.

The solution for one right hand side (corresponding to a voltage source on the antenna) and the three different problems required a total of 29,362,327 flops (detailed numbers are collected in Tab. 1). If the same problems are solved by partitioning, the total flop count is reduced to only 13,982,189 flops or by more than 50 %. Higher improvements on the computational cost can be experienced when larger matrices (especially for the sub-matrix $Z_{11}$) are involved.

For the second problem, the matrix is partitioned into 3 blocks of 975 unknowns. Equation (2) is modified accordingly for inclusion of a third block. Then, each block can be held in a small memory so that disk accesses during the factorization step are minimized. During the process, two matrices ($Z_{12}$ and $Z_{13}$) are modified and must be written out to mass-storage. All other sub-matrices are only read at most two times. The read accesses can be ordered in such a way that the slow mass-storage is used more rarely than the computer's operation system would require when simply swapping out parts of the matrix.

For a three by three partitioning of the matrix, and with the best choice of equal block sizes, 11 read accesses and 2 write accesses are needed. For a factorization and solution with the complete matrix, the matrix must be accessed at least twice (or the nine blocks must be accessed 18 times): once during the factorization and once during the solution.

Table 1: Flop Count for the Solution of the Problems in Fig. 1.

| Configuration | $LU$-factorization | Solution | Total |
|---|---|---|---|
| Monopole | 8,846,578 | 113,286 | 8,959,864 |
| Helix | 10,261,124 | 124,998 | 10,386,122 |
| Forked Monopole | 9,894,325 | 122,016 | 10,016,341 |
| Total | | | 29,362,327 |

(a) Direct Solution

| Computation Step | Build $Z_{22}$ & $LU$ | Solution | Total |
|---|---|---|---|
| $LU$ of $Z_{11}$ | | | 7,875,538 |
| Monopole | 981,381 | 218,403 | 1,199,784 |
| Helix | 2,409,967 | 230,127 | 2,640,094 |
| Forked Monopole | 2,039,631 | 22,7142 | 2,266,773 |
| Total | | | 13,982,189 |
| Reduction | | | 15,380,138 |
| | | | -52 % |

(b) Solution by Partitioning

## 5 Conclusion

The partitioning and subsequent factorization of Method matrices was presented to reduce the required computational time in two cases: the first is the computation of large scatterers when only some unknowns are modified. The second is the solution of large problems not fitting into the computers main and fast-access memory. The solution by partitioning is somewhat similar to a factorization by blocks with storing the intermediate results on mass-storage.

## References


[1] R. F. Harrington, *Field Computation by Moment Methods.* New York: The Macmillan Company, 1968.

[2] R.-Y. Cai, Z.-J. Zeng, and F. CHen, "A partitioning solution of non-symmetrical fully populated matrix system in the boundary element method and its subroutines," in *Applications in Fluid Mechanics and Field Problems* (M. Tanaka, C. Brebbia, and T. Honma, eds.), pp. 463–476, Southampton, Boston: Computational Mechanics Publications, co-published with Springer, 1990.

[3] N. Geng and L. Carin, "Fast multipole method for targets above or buried in lossy soil," *IEEE Ant. and Prop. Symp. Digest,* Orlando, FL, USA, vol. 3, pp. 644–648, July 1999.

[4] U. Jakobus, *Users' Guide to FEKO.* EMSS, `http://www.emss.co.za`, Stellenbosch, South Africa, 1999.