Computational Mechanics 32 (2003) 423–429 © Springer-Verlag 2003
DOI 10.1007/s00466-003-0511-7

Application of the adaptive cross approximation technique for the coupled BE-FE solution of symmetric electromagnetic problems

S. Kurz, O. Rain, S. Rjasanow

Abstract Electromagnetic devices can be analysed by the coupled BE-FE method, where the conducting and magnetic parts are discretized by finite elements. In contrast, the surrounding space is described with the help of the boundary element method (BEM). This discretization scheme is well suited especially for problems including moving parts (see [12]). The BEM discretization of the boundary integral operators usually leads to dense matrices without any structure. A naive strategy for the solution of the corresponding linear system would need at least $O(N^2)$ operations and memory, where N ist the number of unknowns. Methods such as fast multipole [6] and panel clustering [9] provide an approximation to the matrix in almost linear complexity. These methods are based on explicitly given kernel approximations by degenerate kernels, i.e. a finite sum of separable functions, which may be seen as a blockwise low-rank approximation of the system matrix. The blockwise approximant permits a fast matrix-vector multiplication, which can be exploited in iterative solvers, and can be stored efficiently. In contrast to the methods mentioned we will generate [2] the low-rank approximant from the matrix itself using only few entries and without using any explicit a priori known degenerate-kernel approximation. Special emphasis is put on the handling of symmetry conditions in connection with ACA. The feasibility of the proposed method is demonstrated by means of a numerical example.

Keywords Boundary element methods, low-rank approximation, Symmetry exploitation

1 Introduction

The application of the boundary element method for the solution of linear electromagnetic problems has many advantages. Only the boundaries of the considered domains need to be discretized, open boundary problems pose no additional difficulties, and problems including motion can be treated elegantly. The BE-FE coupling procedure is elaborated in [12]. However, application of the BEM leads to dense matrices. The storage requirements and computational costs are of $O(N^2)$, where N is the number of unknowns. One remedy could be the use of an approximation algorithm with almost linear complexity. Section 2 presents the adaptive cross approximation (ACA) algorithm [13] which generates blockwise low rank approximants for the BEM matrices without using an explicit kernel expansion. The exploitation of symmetry is another possibility to reduce computational costs and has been presented in [1, 4, 5] using linear representation theory for finite groups. The aim is a decomposition of function spaces into orthogonal subspaces of symmetric functions, such that each subproblem is defined on a so called symmetry cell. The global solution can then be reconstructed from these components. Sections 3.1–3.2 give an overview of using symmetry in FE and BE methods. Next, in Sect. 3.3 the major contribution of this paper, a symmetry-exploiting ACA algorithm is discussed, which not only reduces the problem size due to symmetry but also yields additional benefit by compression of BEM matrices and possesses an almost linear complexity w.r.t. the number of unknowns. Numerical results obtained by using the symmetry-exploiting ACA algorithm are presented in Sect. 4.

S. Kurz, O. Rain (✉)
Robert Bosch GmbH, Postfach 10 60 50,
70049 Stuttgart, Germany
E-mail: oliver.rain@de.bosch.com

S. Rjasanow
Universität des Saarlandes,
Fachbereich Mathematik,
66041 Saarbrücken, Germany

2 Adaptive cross approximation

Large dense matrices coming from integral equations have no explicit structure in general. However, it is possible to find a permutation so that the matrix with permuted rows and columns contains rather large blocks close to some low-rank matrices [2, 3, 7, 8].

To find a suitable permutation, a cluster tree is constructed by recursively partitioning the collocation points according to some geometrical criterion. A simple example for such a clustering is given in Fig. 1. A large distance between two collocation points results in a large difference of the respective equation numbers. Next, cluster pairs which are geometrically well separated are identified. They will be regarded as "admissible" cluster pairs, i.e. the clusters $\{1, 2, 3, 4, 5\}$ and $\{8, 9, 10\}$ in Fig. 1. The cluster tree together with the set of "admissible" cluster pairs allows the matrix to be split into a collection of blocks of various sizes. The block structure for the simple example is shown in Fig. 2. Since the off-diagonal blocks which describe remote interactions are close to some low-rank matrices, it might be a good idea to approximate them by low-rank matrices. We are thus led to the following matrix

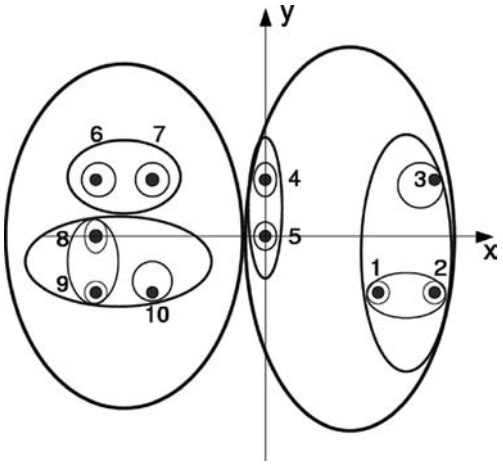

Fig. 1. Clustering for a simple example with 10 collocation points. A large distance between two collocation points results in a large difference of the respective equation numbers

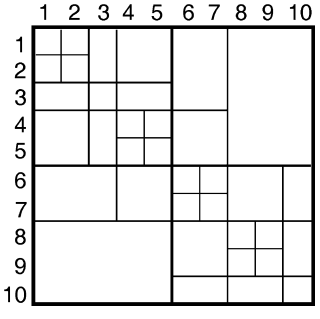

Fig. 2. The permuted matrix for the example depicted in Fig. 1 contains rather large off-diagonal blocks which describe remote interactions and which are close to some low-rank matrices

approximation problem for the individual blocks of the given matrix.

Given a matrix $A \in \mathbb{R}^{n\times m}$ and an accuracy $\varepsilon > 0$, find an approximant $\tilde{A} \in \mathbb{R}^{n\times m}$ with $\|A - \tilde{A}\|_F \le \varepsilon \|A\|_F$ and provide the minimal possible value for $\operatorname{rank}(\tilde{A})$.

Here $\|A\|_F$ denotes the Frobenius norm of the matrix A. The solution of this problem is given by the singular value decomposition (SVD) of the block A

$$\tilde{A} = \sum_{k=1}^{r} \sigma_k u_k v_k^T$$

where (σ_k, u_k, v_k) denote the greatest singular triples of the matrix A and the rank r is chosen so that the required accuracy of the approximation is fulfilled.

Since the SVD requires the computation of the whole matrix A in advance and since SVD is rather expensive with respect to numerical work $O(n^3 + m^3)$ this analytical solution is not practicable.

We now present the algorithm of ACA, which allows the matrix A to be approximated by generating only few of its rows and columns.

Let $S_0 = 0$, $i_1 = 1$ and for $k = 0, 1, 2, \ldots$ compute

1. $e_{i_{k+1}}^T R_k = e_{i_{k+1}}^T A - \sum_{l=1}^{k} (u_l)_{i_{k+1}} v_l^T$,
2. $j_{k+1} : |(R_k)_{i_{k+1} j_{k+1}}| = \max_j |(R_k)_{i_{k+1} j}|$,
3. $v_{k+1} = e_{i_{k+1}}^T R_k / (R_k)_{i_{k+1} j_{k+1}}$,
4. $u_{k+1} = A e_{j_{k+1}} - \sum_{l=1}^{k} (v_l)_{j_{k+1}} u_l$,
5. $i_{k+2} : |(u_{k+1})_{i_{k+2}}| = \max_{i \neq i_{k+1}} |(u_{k+1})_i|$,
6. $S_{k+1} = S_k + u_{k+1} v_{k+1}^T$.

This algorithm produces a sequence of decompositions of the matrix A into a sum $A = R_k + S_k$, where S_k is a low-rank matrix ($\operatorname{rank}(S_k) \le k$) and R_k denotes the error of the approximation. It is important to remark that neither the matrix A nor the error R_k will be computed completely. In the first step of the algorithm the row with index i_{k+1} of the matrix A will be generated and the corresponding row of the error R_k will be computed. During this computation the position and the value of the maximum element in the $i_{k+1}-$row of R_k will be determined (`Step 2`). This element will be called the pivot element. In `Step 3`, the i_{k+1}-row of R_k will be normalized and denoted by v_{k+1}. Since the position j_{k+1} of the pivot element in the i_{k+1}-row of R_k is known we are able to compute the corresponding column of this matrix and denote it as u_{k+1} (`Step 4`). During the computation the position of the next pivot element in the j_{k+1} column will be fixed (i_{k+2}) in `Step 5`. The last step of the algorithm updates the approximation S_k to S_{k+1}. Note that the approximation S_k contains the exact pivot rows and pivot columns of the matrix A for all $k \ge 1$. An appropriate stopping criterion is given by

$$\|u_k\|_F \|v_k\|_F \le \varepsilon \|S_k\|_F \ .$$

Since the matrix A will not be generated completely only the norm of the approximation S_k is available. This norm can be computed recursively the following way

$$\|S_k\|_F^2 = \|S_{k-1}\|_F^2 + 2 \sum_{j=1}^{k-1} (u_j, u_k)(v_j, v_k) + \|u_k\|_F^2 \|v_k\|_F^2 \ .$$

The amount of numerical work required by the ACA algorithm is $O(r^2(m+n))$. Thus if the numerical rank r of the approximation remains constant (which is usually the case) then the total numerical work for the approximation and the memory requirements are both of the order $O(m+n)$.

3 Symmetries in the boundary element methods

The exploitation of symmetry properties allows to decrease the dimension of the problem and therefore achieve a significant reduction of computational costs. Symmetry is a basic principle in nature and describes the invariane of a problem under a certain transformation group. Several methods have been proposed in the literature (e.g. [1, 4, 5]) which allow a reduction of the entire problem to subproblems and following reconstruction of the global solution.

3.1 FE domain

A (complete) geometrical symmetry of the FE domain Ω is given if there exists a finite group $\mathcal{Q}$ of isometries of $\mathbb{R}^3$, such that Ω is invariant under $\mathcal{Q}$. For each element of the symmetry group $\mathcal{Q}$ there is an orthogonal matrix $Q \in \mathbb{R}^3$

(i.e. $QQ^T = Q^TQ = I$) and a symmetry point $x_0 \in \mathbb{R}^3$ such that

$$x' = x_0 + Q(x - x_0) \in \Omega, \quad \forall\, x \in \Omega \ . \tag{1}$$

For the sake of simplicity we assume in the sequel $x_0 = 0$. For a symmetric problem only a part, the so called symmetry cell $C \subset \Omega$, needs to be discretized and considered. C is the smallest subdomain which generates the entire domain Ω under the action of the symmetry group.

In the FE domain we consider the Coulomb gauged $\vec{A} - \phi$ formulation [12]

$$\begin{aligned} \textbf{curl}\ \nu\ \textbf{curl}\ \vec{\mathrm{A}} - \textbf{grad}\ \nu_0\ \mathrm{div}\ \vec{\mathrm{A}} &= \tilde{j}_E, \\ \mathrm{div}\ \tilde{j}_E &= 0 \ , \end{aligned}$$

where $\vec{\mathrm{A}}$ is the magnetic vector potential, ν the field dependent reluctivity and $\tilde{j}_E$ the eddy current density

$$\tilde{j}_E = -\sigma(\partial_t \vec{\mathrm{A}} + \textbf{grad}\ \phi) \ .$$

The electrical conductivity is denoted by σ and the electric scalar potential by ϕ. For $x' = Qx$ the potentials satisfy the condition

$$\begin{aligned} &\vec{\mathrm{A}}(x') = Q\vec{\mathrm{A}}(x), \\ &\phi(x') = \phi(x) \ . \end{aligned}$$

Hence, the symmetry condition

$$\begin{pmatrix} \vec{\mathrm{A}} \\ \phi \end{pmatrix}(x') = \begin{pmatrix} Q & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \vec{\mathrm{A}} \\ \phi \end{pmatrix}(x)$$

should be taken into account when setting up the FEM equation system and reconstructing the global solution.

3.2 BE domain

In this subsection we consider most simple case when the system matrix A can be renumbered and partitioned into $m \times m$ blocks structure having blocks of size exactly $n = N/m$. This is the case for a piecewise constant discretization scheme, where N is the number of unknowns, m is the size of the symmetry group and n is the number of unknowns in each symmetry cell. Let

$$\Gamma_h = \bigcup_{j=1}^{N_{El}} \Gamma_j$$

be a union of boundary elements Γ_j approximating a given surface $\Gamma = \partial\Omega$ and

$$\{\phi_j,\ j = 1, \ldots, N\} \tag{2}$$

a system of ansatz functions having a compact support

$$\mathrm{supp}\ \phi_j \subset \Gamma_h \ .$$

The usual form of the entries of the collocation BEM matrix is as follows

$$a_{ij} = \int_{\Gamma_h} K(x, y_i)\phi_j(x)\mathrm{d}F_x, \quad i,j = 1, \ldots, N \ . \tag{3}$$

Thus the full BEM matrix requires N^2 memory units. Using the symmetry of the problem is one of the obvious possibilities for reducing computer memory consumption. Several symmetry conditions should be fulfilled to yield a really effective procedure.

Besides the geometrical symmetry of Ω we require the geometrical symmetry of its boundary Γ, i.e. for the symmetry mappings Q from above holds

$$x' = Qx \in \Gamma, \quad \forall\, x \in \Gamma \ . \tag{4}$$

For smooth surfaces Γ_h the condition (4) implies the following connection of unit normal vectors to Γ_h at x and $x' = Qx$

$$Qn_x = n_{Qx} = n_{x'}, \quad \forall\, x \in \Gamma_h \ . \tag{5}$$

The system of boundary elements, collocation points and the ansatz functions (2) have discretization symmetry if there is a permutation $\{\sigma_1, \ldots, \sigma_N\}$ of the index set $\{1, \ldots, N\}$ with the following properties

$$Qy_i = y_{\sigma_i}, \quad i = 1, \ldots, N \ , \tag{6}$$

$$Q(\mathrm{supp}\ \phi_j) = \mathrm{supp}\ \phi_{\sigma_j}, \quad j = 1, \ldots, N \ , \tag{7}$$

$$\begin{aligned} &\phi_j(x) = \phi_j(Q^Tx') = \phi_{\sigma_j}(x') = \phi_{\sigma_j}(Qx), \\ &\forall\, x \in \mathrm{supp}\ \phi_j,\ \ j = 1, \ldots, N \ . \end{aligned} \tag{8}$$

The permutation $\{\sigma_1, \ldots, \sigma_N\}$ offers the possibility for renumbering unknowns corresponding to the symmetry of the problem:

$$\{1, 2, \ldots, N/2, \sigma_1, \sigma_2, \ldots, \sigma_{N/2}\} \ , \tag{9}$$

where the pairs (i, σ_i) are arbitrarily chosen from the set

$$\{1, \ldots, N\} \backslash \{1, 2, \ldots, i-1, \sigma_1, \sigma_2, \ldots, \sigma_{i-1}\} \ .$$

The problem has the symmetry of the kernel if the function $K(x, y)$ features the following property

$$K(x, y) = K((x - y, x - y), (x - y, n_x)) \ . \tag{10}$$

Lemma 1 *The symmetrical BEM discretization* (6)–(8) *of the geometrically symmetrical problem* (5) *having symmetrical kernel (10) leads, by the numbering of unknowns corresponding to* (9)*, to the following property of the matrix A in* (3)*:*

$$a_{ij} = a_{\sigma_i\sigma_j}, \quad \forall\, i,j \ . \tag{11}$$

Proof. Definition (3) leads after substitution (4) to

$$\begin{aligned} a_{ij} &= \int_{\mathrm{supp}\ \phi_j} K(x, y_i)\phi_j(x)\mathrm{d}F_x \\ &= \int_{Q(\mathrm{supp}\ \phi_j)} K((Q^Tx' - y_i, Q^Tx' - y_i), (Q^Tx' - y_i, n_x)) \\ &\qquad \times \phi_j(Q^Tx')\mathrm{d}F_{x'} \\ &= \int_{\mathrm{supp}\ \phi_{\sigma_j}} K((x' - Qy_i, x' - Qy_i), (x' - Qy_i, Qn_x)) \\ &\qquad \times \phi_{\sigma_j}(x')\mathrm{d}F_{x'} \end{aligned}$$

$$= \int_{\text{supp}\,\phi_{\sigma_j}} K((x'-y_{\sigma_i}, x'-y_{\sigma_i}), (x'-y_{\sigma_i}, n_{x'})) \times \phi_{\sigma_j}(x')\mathrm{d}F_{x'} = a_{\sigma_i\sigma_j} \ ,$$

where the properties (5)–(8) have been used.

Since $\sigma_{\sigma_i} = i$ for all i the property (11) implies

$a_{i\sigma_j} = a_{\sigma_i j}, \quad \forall\ i,j$.

Thus the system of linear equations of the symmetrical BEM takes the following block-circulant form

$$\begin{pmatrix} A_{11} & A_{12} \\ A_{12} & A_{11} \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \ , \tag{12}$$

where the vectors u_1, u_2, b_1 and b_2 have half of the initial dimension.

If the problem possesses more than one symmetry then the same procedure can be applied for each block of the matrix (12) and the system obtains the more general block-circulant form

$$\begin{pmatrix} A_1 & A_2 & \dots & A_m \\ A_m & A_1 & \dots & A_{m-1} \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ A_2 & A_3 & \dots & A_1 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ \dots \\ \dots \\ u_m \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \dots \\ \dots \\ b_m \end{pmatrix} \ . \tag{13}$$

Thus only the basis matrices $A_1, A_2, \dots, A_m$ should be generated and stored. The amount of numerical work and of memory will therefore be reduced from N^2 to N^2/m. This factor can be very useful for practical computations. The numerical solution of the system of linear equations having a block-circulant matrix can also be implemented much more efficiently than a straightforward direct elimination method which would lead to $O(N^3)$ arithmetical operations [16]. The main property of the circulant matrices

$$A = \begin{pmatrix} a_1 & a_2 & a_3 & \dots & a_{m-1} & a_m \\ a_m & a_1 & a_2 & a_3 & \dots & a_{m-1} \\ a_{m-1} & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & a_2 \\ a_2 & a_3 & \dots & a_{m-1} & a_m & a_1 \end{pmatrix} \in \mathbb{C}^{m\times m}$$

is that all of them are simultaneously diagonalized by the matrix of the discrete Fourier transform F_m:

$$A = \frac{1}{m} F_m \Lambda F_m^*, \quad \Lambda = \mathrm{diag}(F_m a), \qquad f_{k,l} = \omega_m^{(k-1)(l-1)} = \mathrm{e}^{i\frac{2\pi}{m}(k-1)(l-1)} \ . \tag{14}$$

The most simple nontrivial circulant matrix

$$J = \begin{pmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \\ \dots & \dots & \dots & \dots & \\ 0 & 0 & 0 & \dots & 1 \\ 1 & 0 & 0 & \dots & 0 \end{pmatrix}$$

has the following eigenvalues

$\Lambda = \mathrm{diag}\big(\omega_m^{l-1},\ l = 1, \dots, m\big)$.

Using the Kronecker product $\otimes$ of matrices we rewrite the block-circulant matrix A of the system (13) in the form (cf. (14))

$$A = \sum_{k=1}^{m} J^{k-1} \otimes A_k = \frac{1}{m} \sum_{k=1}^{m} (F_m \Lambda^{k-1} F_m^*) \otimes A_k \ ,$$

where the dimension of the matrices A_k is now $n = N/m$. Since

$F_m F_m^* = F_m^* F_m = I_m$

and using the known property of the Kronecker product

$(A \otimes B)(C \otimes D) = (AC) \otimes (BD)$

we obtain

$$(F_m^* \otimes I_n) A (F_m \otimes I_n) = \sum_{k=1}^{m} J^{k-1} \otimes A_k = m \sum_{k=1}^{m} \Lambda^{k-1} \otimes A_k \ .$$

The system (13) can now be rewritten in the block-diagonal form

$$\begin{pmatrix} D_1 & 0 & \dots & 0 \\ 0 & D_2 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & D_m \end{pmatrix} \begin{pmatrix} \tilde{u}_1 \\ \tilde{u}_2 \\ \dots \\ \dots \\ \tilde{u}_m \end{pmatrix} = \begin{pmatrix} \tilde{b}_1 \\ \tilde{b}_2 \\ \dots \\ \dots \\ \tilde{b}_m \end{pmatrix} \ , \tag{15}$$

where

$$D_l = \sum_{k=1}^{m} \omega_m^{(l-1)(k-1)} A_k \in \mathbb{C}^{n\times n} \tag{16}$$

and

$\tilde{u} = (F_m^* \otimes I_n) u, \ \tilde{b} = (F_m^* \otimes I_n) b$.

Thus the following algorithm has been derived (similar to proposed in [1])

1. Compute all basis matrices $A_k,\ k = 1, \dots, m$
2. Compute

   $\tilde{b} = (F_m^* \otimes I_n) b$

   using n Fast Fourier Transforms (FFT).
3. For $l = 1, \dots, m$

3.1 Generate the matrix

$$D_l = \sum_{k=1}^{m} \omega_m^{(l-1)(k-1)} A_k$$

3.2 Solve the system

$$D_l \tilde{u}_l = \tilde{b}_l$$

4. Compute

$$u = \frac{1}{m} (F_m \otimes I_n) \tilde{u}$$

using n FFT's.

The straightforward implementation of this algorithm leads to $O(mn^2)$ operations and memory units in `Step 1`, $O(n \log(m))$ operations in `Step 2`, $O(mn^2)$ operations and memory units in `Step 3.1`, $O(mn^3)$ operations for solving all systems in `Step 3.2` and finally $O(nm \log(m))$ operations in the last `Step 4`. Thus `Step 3.2` is the most expensive and defines the final amount of numerical work for the whole algorithm $O(mn^3) = O(N^3/m^2)$. This amount remains of the same capital order of $O(N^3)$, but it is reduced by a remarkable factor m^2 which can be 64 for a domain which is symmetrical with respect to the reflections over all three coordinate planes – as we will consider in the next section.

3.3
Symmetry of excitation and symmetry-exploiting ACA

In the BE domain the equation

$$\Delta \vec{A} = -\mu_0 \tilde{j}_S$$

is to be solved, where $\vec{A}$ is the Coulomb gauged magnetic vector potential and $\tilde{j}_S$ is an impressed source current density. This vector equation decouples into three scalar Laplace equations for the Cartesian components of $\vec{A}$, so that we are left with the equation system (13).

Electromagnetic devices often possess the symmetry of excitation, which means, that the symmetry mappings Q fulfill

$$\tilde{j}_S(Qx) = Q\tilde{j}_S(x), \quad \forall\, x \in C \ . \tag{17}$$

In case of an excitation symmetry we don't perform the Fourier transform as described in the previous section but simplify the equation system (13) in a different way. As a consequence from (17) we obtain a linear dependency of the components of the r.h.s. in (13)

$$b_2 = \alpha_2 b_1, \ldots, b_m = \alpha_m b_1 \quad \text{with } \alpha_2, \ldots, \alpha_m \in \mathbb{R} \ .$$

Thus the equation system can be reduced for each potential component (A_x, A_y and A_z) to one subsystem of dimension N/m, where N is the total number of unknowns per potential component. Since the excitation symmetry can differ for different potential components, there are up to three subsystems to be solved. Let D_x, D_y and D_z be the corresponding system matrices, which can be calculated as linear combination of basis matrices $A_1, \ldots, A_m$. Thus the exploitation of the excitation symmetry leads to reduction of computational costs from N^2 to at most $3N^2/m^2$.

Instead of assembly of D_x, D_y and D_z from $A_1, \ldots, A_m$ during the matrix computation we could compute and store the basis matrices $A_1, \ldots, A_m$ and perform the assembly during the matrix-vector product. This is of course not reasonable in full computation of matrices. However, when applying the ACA method the individual approximation and storage of all m basis matrices is reasonable, because most of them can be approximated very efficiently. The assembly of the system matrices by means of linear combination is carried out in the matrix-vector multiplication. Due to the almost linear complexity of the ACA algorithm [2] the size of a compressed matrix A_j is approximately $c_j N/m$, where c_j is some constant depending on geometry and accuracy of the approximation. Thus the memory requirements for the entire system matrix consisting of $A_1, \ldots, A_m$ as well as the costs of matrix-vector multiplication amount to approximately cN with $c = \sum_{j=1}^{m} c_j/m$. Numerical results presented in the next section show the almost linear behaviour of the method w.r.t. the number of unknowns N.

4
Numerical results

Electromagnetic devices can be analysed by the coupled BEM–FEM method, where the conducting and magnetic parts are discretized by finite elements. In contrast, the surrounding space is described with the boundary element method. This discretization scheme is well suited for problems including moving parts and has been described in detail elsewhere [10–12].

For numerical tests we consider TEAM workshop problem 10 [14] (TEAM = Testing Electromagnetic Analysis Methods). An exciting coil is set between two steel channels, and a steel plate is inserted between the channels. The geometry is symmetrical with respect to all three coordinate planes. In order to examine the behaviour of the ACA algorithm and the Full BEM method when exploiting symmetries, we consider along with the full mesh three further meshes exploiting one, two and all

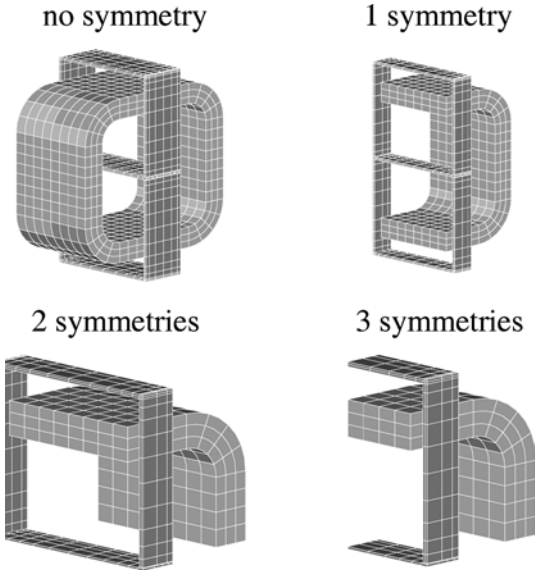

Fig. 3. TEAM problem 10. An exciting coil is set between two steel channels, and a steel plate is inserted between the channels. This geometry is symmetrical with respect to all coordinate planes

Table 1. Resources needed for the ACA and the Full BEM. Both methods have been compared on three different mesh sequences. The ACA accuracy has been set at 10^{-4}

Mesh	No symmetry $m = 1$	1 symmetry $m = 2$	2 symmetries $m = 4$	3 symmetries $m = 8$
Coarse	$n = 4230$	$n = 2287$	$n = 1164$	$n = 596$
ACA	59.5 Mb	38.1 Mb	18.8 Mb	9.2 Mb
	847 s	447 s	254 s	117 s
Full BEM	273.0 Mb	159.6 Mb	62.0 Mb	16.2 Mb
	567 s	213 s	72 s	25 s
Medium	$n = 8142$	$n = 4399$	$n = 2234$	$n = 1131$
ACA	157.3 Mb	94.1 Mb	46.1 Mb	22.2 Mb
	1648 s	903 s	468 s	251 s
Full BEM	1011.6 Mb	590.6 Mb	228.5 Mb	58.5 Mb
	3087 s	1148 s	315 s	82 s
Fine	$n = 16902$	$n = 8795$	$n = 4438$	$n = 2246$
ACA	416.1 Mb	231.5 Mb	112.1 Mb	54.0 Mb
	3770 s	1902 s	1024 s	508 s
Full BEM	4359.1 Mb	2360.6 Mb	901.7 Mb	230.9 Mb
	–	–	1893 s	411 s

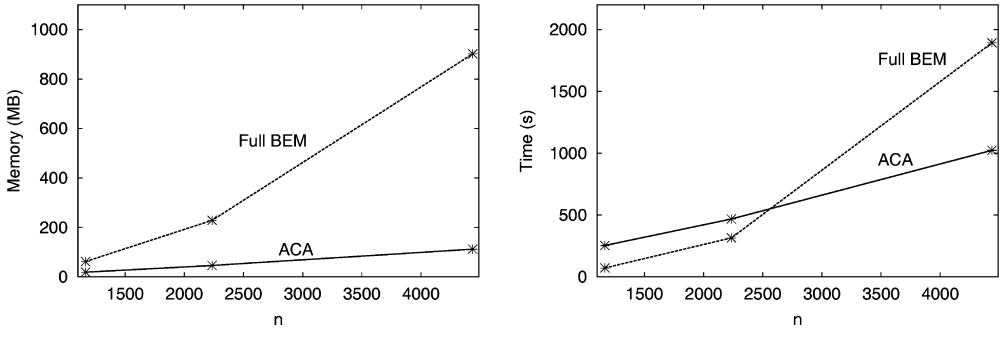

Fig. 4. Memory requirements (left) and CPU times (right) versus problem size for fixed $m(m = 4)$ and variable mesh

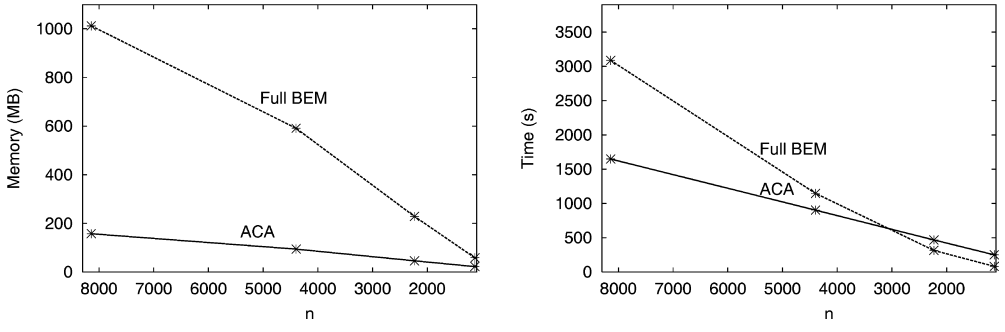

Fig. 5. Memory requirements (left) and CPU times (right) with respect to the symmetry for variable m and fixed mesh (medium discretization)

three symmetries respectively (Fig. 3). Additionally, for each mesh of this mesh sequence we gradually perform two mesh refinements to show the linear behaviour of the ACA algorithm with respect to the problem size. Thus we obtain three mesh sequences with altogether 12 meshes. Hexahedral second order FEM elements (20 nodes) are used in connection with rectangular second order BEM elements (8 nodes).

In the case when there are some fixed collocation points (e.g. points on a symmetry face in case of a mirror symmetry), the size of each subblock in (13) is close to, but not exactly equal to, $n = N/m$ and the matrices D_x, D_y, D_z for the single layer potential become singular. However, the global system has a unique solution [1]. There are several methods to handle the subsystems via regularization or via projections proposed in [1]. Since in our solver no inversion of approximated matrices takes place, also singular matrices can be handled and the unique global solution can still be reconstructed without further difficulties.

TEAM problem 10 is treated as a magnetostatic problem (for details see [15]). In all computations we set the ACA accuracy $\varepsilon = 10^{-4}$. The problem is solved using both the ACA algorithm and the Full BEM method. Table 1 shows for both algorithms the memory requirement of BEM matrices as well as the CPU time needed for the solution. All values refer to a 450-MHz Sun Ultra workstation. We compared the average flux density in the center of the inner steel plate ($\bar{B}_z = 1.663$ T) with measurements ($\bar{B}_z = 1.654$ T) [15] and found good agreement. The difference of the computed flux densities with and without ACA is neglectable ($\Delta\bar{B}_z \approx 3 \cdot 10^{-4}$ T).

One can observe for any kind of symmetry that the growth of the problem size due to the mesh refinements

Table 2. Relative size of BEM matrices coming from the single layer potential for the medium mesh sequence. The percentage gives the compression rate obtained by the ACA algorithm for each individual submatrix using the approximation accuracy $\epsilon = 10^{-4}$. Submatrices which involve transformed nodes show a very good compression

Matrix	No symmetry $m = 1$ $n = 8142$	1 symmetry $m = 2$ $n = 4399$	2 symmetries $m = 4$ $n = 2234$	3 symmetries $m = 8$ $n = 1131$
A_1	12.5% (63.2 Mb)	15.4% (22.7 Mb)	20.5% (7.8 Mb)	32.8% (3.2 Mb)
A_2	–	10.1% (14.9 Mb)	12.9% (4.9 Mb)	18.4% (1.8 Mb)
A_3	–	–	8.7% (3.3 Mb)	12.3% (1.2 Mb)
A_4	–	–	6.3% (2.4 Mb)	8.2% (0.8 Mb)
A_5	–	–	–	6.1% (0.6 Mb)
A_6	–	–	–	5.1% (0.5 Mb)
A_7	–	–	–	5.1% (0.5 Mb)
A_8	–	–	–	3.1% (0.3 Mb)
Total memory	63.2 Mb	37.6 Mb	18.4 Mb	8.9 Mb

results in a linear growth of the memory consumption and the CPU time for the ACA algorithm. Figure 4 shows the comparison between the ACA and the Full BEM for one kind of symmetry. Although the ACA algorithm is slower for coarse meshes, its linear complexity makes it superior for large n.

Now we examine the effect of the symmetry exploitation. It is clear that the profit using Full BEM should be of order $O(n^2)$ whereas the memory requirement and CPU time reduction using ACA is expected to be linear. Figure 5 shows the behaviour of the memory usage and CPU time for the medium mesh sequence. As mentioned above in the case of ACA the individual approximation and storage of all m basis matrices will be performed. The relative size of the basis matrices resulting from the single layer potential is shown in Table 2. The assembly of the system matrices by means of linear combination is carried out in the matrix-vector multiplication.

The full BEM method performs the assembly of D_x, D_y, D_z during the matrix computation. The number of different matrices depends on the kind of geometrical and excitational symmetry. For the TEAM 10 example it holds that $D_x = D_y = D_z$ in the case without symmetry, $D_x \neq D_y = D_z$ in the case of one symmetry, and three different matrices arise in case of two or three symmetries. For this reason the curve corresponding to the total memory requirements of the Full BEM method in Fig. 5 does not actually decrease like $O(n^2)$ but the memory requirements for each single matrix do.

The numerical example considered here exhibits the property of excitation symmetry. Note that in the general case of non-symmetric excitation the memory requirements would decrease linearly w.r.t the size m of the symmetry group, as can be seen from the Eq. (15), and therefore like $O(n)$.

5 Conclusions

The memory requirement as well as the computational time for the solution of electromagnetic problems can be reduced by application of the ACA algorithm or by exploitation of symmetry. Whereas the latter method still possesses the complexity $O(N^2)$, the ACA algorithm shows almost linear behaviour with respect to the number of unknowns N. The combination of both methods gives the possibility for the asymptotically optimal and practically feasible solution of electromagnetic problems.

References

1. **Allgower EL, Georg K, Miranda R, Tausch J** (1998) Numerical exploitation of equivariance. Z. Angew. Math. Mech. 78: 795–806
2. **Bebendorf M** (2000) Approximation of boundary element matrices. Numerische Mathematik 86(4): 565–589
3. **Bebendorf M, Rjasanow S** (2000) Matrix compression for the radiation heat transfer in exhaust pipes. In: Multifield Problems, Sändig A-M, Schiehlen W, Wendland WL (eds), pp. 183–191. Springer-Verlag, Berlin
4. **Bonnet M** (2003) Exploiting partial or complete geometrical symmetry in 3D symmetric Galerkin indirect BEM formulations. Int. J. Numer. Meth. Eng. 57: 1053–1083
5. **Bossavit A** (1986) Symmetry, groups and boundary value problems: a progressive introduction to noncommutative harmonic analysis of partial differential equations in domains with geometrical symmetry. Comput. Meth. Appl. Mech. Eng. 56: 167–215
6. **Cheng H, Greengard L, Rokhlin V** (1999) A fast adaptive multipole algorithm in three dimensions. J. Comput. Phys. 155(2): 468–498
7. **Goreinov SA, Tyrtyshnikov EE, Zamarashkin NL** (1997) A theory of pseudoskeleton approximations. Linear Algebra Appl. 261: 1–21
8. **Hackbusch W** (1999) A sparse matrix arithmetic based on H-matrices. Part I. Computing 62(2): 89–108
9. **Hackbusch W, Nowak ZP** (1989) On the fast matrix multiplication in the boundary element method by panel clustering. Numer. Math. 54(4): 463–491
10. **Kurz S, Becker U, Maisch H** (2001) Dynamic simulation of electromechanical systems – from Maxwell's theory to common rail Diesel injection. Naturwissenschaften 88(5): 183–192
11. **Kurz S, Fetzer J, Lehner G, Rucker WM** (1998) A novel formulation for 3D eddy current problems with moving bodies using a Lagrangian description and BEM-FEM coupling. IEEE Trans. Magnetics 34(5): 3068–3073
12. **Kurz S, Fetzer J, Lehner G, Rucker WM** (1999) Numerical analysis of 3D eddy current problems with moving bodies using BEM-FEM coupling. Surveys on Mathematics for Industry 9: 131–150
13. **Kurz S, Rain O, Rjasanow S** (2002) The adaptive cross approximation technique for the 3D boundary element method. IEEE Trans. Magnetics 38(2): 421–424
14. **Nakata T, Takahashi N, Fujiwara K** (1992) Summary of results for benchmark problem 10 (steel plates around a coil). COMPEL 11: 335–344
15. **Preis K, et al** (1991) Numerical analysis of 3D magnetostatic fields. IEEE Trans. Magnetics 27(5): 3798–3803
16. **Rjasanow S** (1994) Effective Algorithms with Block Circulant Matrices. Linear Alg. Appl. 202: 55–69