Technical Article

Solution of Magnetostatic Field Problems with the Integral Equation Method

Abstract —**For the solution of magnetostatic field problems the integral equation method (IEM) lived in the shadows because of its high computational costs. However, in recent years it was shown that system matrix compression techniques can make the IEM extremely efficient. In this regard we use the fast multipole method. When complex problems have to be solved, however, compression of the system matrix only is often not sufficient. Thus, we developed approaches to speed up the assembly of the right-hand side as well as post-processing. Several non-linear problems with complex geometries are presented.**

I. Introduction

The solution of magnetostatic field problems with finite element methods suffers from some drawbacks, e.g. boundary conditions have to be defined. As on the surfaces of magnetically active parts, i. e. hard magnetic and soft magnetic material as well as coils, boundary conditions are often not known, the air region around them has to be meshed up to some distance. These additional degrees of freedom lead to unnecessarily high computational costs. By contrast, the integral equation method (IEM) requires only a discretization of the magnetically active parts themselves. This is why the IEM is very applicable for open boundary problems such as stray field analysis.

Nevertheless, the main drawback of the IEM is its fully dense system matrix whose computational costs are of $O(N^2)$ which restricts its application to small problems. Also, assembly of the right-hand side and post-processing may require large computation time when complex geometries are involved. However, in recent years, matrix compression techniques, e. g. the fast multipole method (FMM) [1] have proved to be efficient with the boundary element method [2]. Its restriction to linear problems can be overcome by using additional volume elements, which result in the IEM. Compressing the system matrix of the IEM is straightforward and it was shown that computational costs are reduced to approximately $O(N)$ [3], [4], [5], [6].

The paper is structured as follows. After a description of the used IEM formulation, acceleration with the FMM is discussed, both for matrix compression and for right-hand side assembly. A superconvergent patch recovery method improves post-processing results. Although the FMM dramatically reduces the computational costs, parallelization is necessary, especially for an efficient use of modern computers. Several parallelization approaches are shown. Finally, numerical examples show, that all methods are applicable to complex non-linear problems.

II. Integral Equation Method

A typical configuration of the considered field problems is shown in Fig. 1. Magnetic fields are excited by the magnetic field $\boldsymbol{H}_J$ of coils with free currents $\boldsymbol{J}$ in the domain Ω_J and by the magnetic field $\boldsymbol{H}_P$ of hard magnetic material in domain Ω_P with magnetization $\boldsymbol{M}_P$. $\boldsymbol{H}_J$ and $\boldsymbol{H}_P$ magnetize the soft magnetic material with permeability μ_r in domain Ω_M. The resulting total magnetic field $\boldsymbol{H}$ can be split into the sum

$$\boldsymbol{H} = \boldsymbol{H}_P + \boldsymbol{H}_J + \boldsymbol{H}_M , \tag{1}$$

where $\boldsymbol{H}_M$ is the induced field of the soft magnetic material.

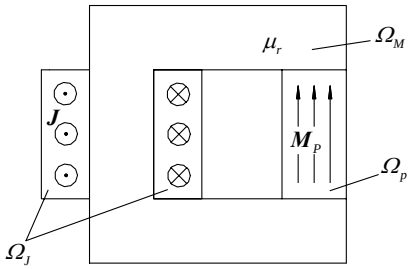

Fig. 1. Considered field problems.

Whilst $\boldsymbol{H}_J$ and $\boldsymbol{H}_P$ can be computed directly, a system of algebraic equations has to be solved in order to compute $\boldsymbol{H}_M$. However, this approach can give very inaccurate results, because $\boldsymbol{H}_M$ is often almost anti-parallel to $\boldsymbol{H}_J$ or $\boldsymbol{H}_P$ within Ω_M. This is the so-called demagnetization effect [7]. In these cases large cancellation errors occur when (1) is evaluated. This can be prevented by computing $\boldsymbol{H}$ directly [8]. To obtain a minimal number of unknowns, $\boldsymbol{H}$ can be represented by scalar source values such as, equivalent magnetic surface and volume charges, double layer distributions or the magnetic scalar potential. We use a formulation based on [9], which requires only discretization of Ω_M and which has the total scalar potential ψ and the susceptibility tensor $\boldsymbol{\chi}$ as unknowns. From (1) follows

$$\psi = \psi_P + \psi_J + \psi_M , \tag{2}$$

where ψ_P, ψ_J and ψ_M are the scalar potentials of $\boldsymbol{H}_P$, $\boldsymbol{H}_J$, and $\boldsymbol{H}_M$, respectively.

$$\psi_S(\boldsymbol{r}) = \int_{\Omega_S} \boldsymbol{M}(\boldsymbol{r}') \cdot \nabla' G(\boldsymbol{r}, \boldsymbol{r}') \mathrm{d}V' \tag{3}$$

is given in [10], where $\boldsymbol{r}$ and $\boldsymbol{r}'$ are observation and source points, respectively and G is Green's function of free space

$$G(\boldsymbol{r}, \boldsymbol{r}') = \left(4\pi\mu_0 |\boldsymbol{r} - \boldsymbol{r}'|\right)^{-1} . \tag{4}$$

With

$$\boldsymbol{M} = \mu_0 \boldsymbol{\chi} \cdot \boldsymbol{H} = -\mu_0 \boldsymbol{\chi} \cdot \nabla \psi \tag{5}$$

and by substituting (3) in (2) yields to

$$\begin{aligned} &\psi(\boldsymbol{r}) + \mu_0 \int_{\Omega_M} \left[\boldsymbol{\chi}(\boldsymbol{r}') \cdot \nabla \psi(\boldsymbol{r}')\right] \cdot \nabla' G(\boldsymbol{r}, \boldsymbol{r}') \mathrm{d}V' \\ &= \psi_J(\boldsymbol{r}) + \psi_P(\boldsymbol{r}) \end{aligned} \tag{6}$$

which is a volume integral equation for the unknown ψ.

To compute ψ_J, line integration over $\boldsymbol{H}_J$ has to be carried out

$$\psi_J(\boldsymbol{r}) = -\int_{\boldsymbol{r}_0}^{\boldsymbol{r}} \boldsymbol{H}_J(\boldsymbol{r}') \cdot \mathrm{d}\boldsymbol{r}', \tag{7}$$

where the integration path must not intersect Ω_J. $\boldsymbol{H}_J$ is obtained by Biot-Savart law

$$\boldsymbol{H}_J(\boldsymbol{r}) = \mu_0 \int_V \boldsymbol{J}(\boldsymbol{r}') \times \nabla' G(\boldsymbol{r}, \boldsymbol{r}') \mathrm{d}V. \tag{8}$$

The potential due to permanent magnets is

$$\psi_P(\boldsymbol{r}) = \int_{\Omega_P} \boldsymbol{M}_P(\boldsymbol{r}') \cdot \nabla' G(\boldsymbol{r}, \boldsymbol{r}') \mathrm{d}V'. \tag{9}$$

Discretization of (6) by means of shape functions N_j

$$\psi(\boldsymbol{r}) = \sum_{j=1}^{n} \psi_j N_j(\boldsymbol{r}) \,(17) \tag{10}$$

and with collocation method [11] yields to the system of algebraic equations

$$\begin{aligned} \psi(\boldsymbol{r}_i) + \sum_{j=1}^{n_M} \psi_j \int_{V_j} \mu_0 \left[\chi(\boldsymbol{r}') \cdot \nabla' N_j(\boldsymbol{r}') \right] \cdot \nabla' G(\boldsymbol{r}_i, \boldsymbol{r}') \mathrm{d}V' \\ = \psi_J(\boldsymbol{r}_i) + \psi_P(\boldsymbol{r}_i) \end{aligned}, \tag{11}$$

where n_M is the number of nodes within Ω_M and $\boldsymbol{r}_i$ are node coordinates.

Because χ usually depends on the magnetic field, (11) is in general non-linear and therefore has to be solved iteratively. With direct iteration (11) is solved for a linear inhomogeneous distribution of χ. New values of χ have to be calculated for each non-linear iteration step. Typical $\chi(H)$ characteristics have a maximum. If H_m is the field strength where this maximum occurs, new values of iteration step k are obtained by

$$\chi^{(k)} = \begin{cases} B\left(H^{(k)}\right) \Big/ \left(\mu_0 H^{(k)}\right) - 1 \;, H^{(k)} < H_m \\ \tilde{B}^{(k)} \Big/ \left(\mu_0 H\left(\tilde{B}^{(k)}\right)\right) - 1, H^{(k)} \geq H_m \end{cases} \tag{12}$$

with the inverse magnetization characteristic $H(B)$ and $\tilde{B}^{(k)} = \mu_0 \left(\chi^{(k-1)} + 1\right) H^{(k)}$. This procedure allows global convergence even if the material is mainly in the steep section of the $B(H)$-characteristic. To accelerate the iteration, super-relaxation

$$\chi^{(k)} = \omega \chi^{(k)} + (1 - \omega) \chi^{(k-1)} \tag{13}$$

with $\omega > 1$ is used. Iteration is stopped, when the deviations $\mathrm{mean}(\Delta\chi/\chi)$ and $\max(\Delta\chi/\chi)$ are sufficiently small.

There are some drawbacks when evaluating (11). First, the system matrix is fully dense. Computational costs for its assembly are of order $O(N^2)$. Also, at every node $\boldsymbol{r}_i$ integration over the whole domains Ω_J and Ω_P is necessary to evaluate ψ_J and ψ_P, which can make post-processing very slow. An application of the FMM to compress the system matrix and to accelerate computation of the right-hand side is therefore shown in chapter III.

Another difficulty arises when the magnetic field $\boldsymbol{H}$ needs to be computed at the nodes $\boldsymbol{r}_i$ within Ω_M, e. g. to evaluate (12) or for post-processing purposes. It can be obtained by numerical differentiation

$$\boldsymbol{H}(\boldsymbol{r}) = -\nabla \psi(\boldsymbol{r}), \tag{14}$$

but (14) is not unique at $\boldsymbol{r}_i$, since elements with C^0 continuity are used(9). A smoothing approach that allows high precision gradient computation such as in (14) is therefore presented in chapter IV.

Modern computers are equipped with multi-core CPUs and computer clusters are very popular. Hence, a parallelization is recommended. Especially complex practical problems often lead to very large systems of algebraic equations. Memory of a single computer is then insufficient and CPU times are very large in the case of many time steps. Different parallelization approaches are shown in chapter V to overcome these problems.

III. Fast Multipole Method

A. System Matrix Compression

The matrix of the linear system of equations of the IEM is fully dense, since each element interacts with each other. Fortunately, the matrix itself is not explicitly needed, if the system of linear equations is solved iteratively. Matrix-by-vector products have to be computed only. In the FMM algorithm, the matrix-by-vector product is split into a near-field part due to elements that are close to each other and a far-field part for the remaining elements

$$y = [A] \cdot \{x\} = [A_{near}] \cdot \{x\} + \{y_{far}\}. \tag{15}$$

The division of element interactions into a near-field and a far-field part is carried out by means of a hierarchical grouping scheme which is based on cubes, a so-called octree. The near-field matrix $[A_{near}]$ is a sparse matrix, which equals the total matrix of (11) with the far-field interactions removed. Elements whose distance from each other is sufficiently large are considered group-wise, where the size of the group depends on the distance. Their interactions are given in $\{y_{far}\}$. Therefore, Green's function is approximated by a truncated series expansion of order L into spherical harmonics Y_n^{-m}

$$\frac{1}{|\boldsymbol{r} - \boldsymbol{r}'|} = \sum_{n=0}^{L} \sum_{m=-n}^{n} \frac{r'^n}{r^{n+1}} Y_n^m(\theta, \varphi) Y_n^{-m}(\theta', \varphi'), r' < r. \tag{16}$$

Substituting (16) into the integral of (11) yields

$$\psi_M(\boldsymbol{r}) = \frac{1}{4\pi} \sum_{n=0}^{L} \sum_{m=-n}^{n} \left\{ \boldsymbol{M}_n^m \frac{1}{r^{n+1}} Y_n^m(\theta, \varphi) \right\}, \tag{17}$$

where $\boldsymbol{M}_n^m$ are the so-called multipole coefficients

$$\boldsymbol{M}_n^m = \boldsymbol{M}(\boldsymbol{r}') \cdot \nabla' \left[r'^n Y_n^{-m}(\theta', \varphi') \right] \mathrm{d}V'. \tag{18}$$

The multipole coefficients ${}_i\boldsymbol{M}_n^m$ of a node i needed for compression of the system matrix in (11) are obtained from (18) with (5) and (10)

$${}_i\boldsymbol{M}_n^m = \sum_{j=1}^{n_i} \psi_j \int_{V_i} \boldsymbol{\chi}(\boldsymbol{r}') \cdot \nabla' N_j(\boldsymbol{r}) \cdot \nabla' \left[r'^n Y_n^{-m}(\theta', \varphi') \right] \mathrm{d}V'. \tag{19}$$

The multipole coefficients of each cube of the octree are transformed to the so-called local coefficients $\boldsymbol{L}_n^m$ from which the magnetic field can be computed

$$\boldsymbol{H}_{far}(\boldsymbol{r}) = \frac{1}{4\pi}\sum_{n=0}^{L}\sum_{m=-n}^{n} \boldsymbol{L}_{\mathrm{n}}^{\mathrm{m}} r^{n} Y_n^m(\theta,\varphi). \tag{20}$$

Compared to BEM where only surface elements occur, it is much more important with the IEM to keep the number of near-field interactions as small as possible. This is because integrations with volume elements are in general much more time-consuming than with surface elements. In addition, elements in the near-field are often located in all three spatial directions and not only in two as in the case of BEM. The latter not only leads to more computation time but also to a lower memory compression rate. A small near-field requires higher order L of series expansions. It was shown that computational costs are reduced with modified transformations of the multipole coefficients from $O(L^4)$ to $O(L^3)$ [12]. In addition, the different sizes of surface and volume elements have to be considered for the grouping scheme [13].

B. Acceleration of the Right-Hand Side

Because permanent magnets often border the soft magnetic material directly, their discretization has to be of equal refinement on the interface. In these cases a large number of permanent magnet elements may occur. In order to give respect to the field of permanent magnets in the right-hand side of (11), at every node within Ω_M the potential ψ_P has to be computed by using (9). If the permanent magnets are meshed with n_P elements, the time complexity of this task is $n_P n_M$. If the permanent magnets are discretized with a high number of elements, assembly of the right-hand side may therefore even take longer than assembly of the system matrix. It is therefore recommendable to apply the FMM to compute (9). This can easily be done by using (17) to obtain the multipole coefficients of the field of the permanent magnets.

This approach is similar to the one for the post-processing [14]. There, the multipole coefficients are computed from all sources in the cubes, impressed sources and calculated equivalent sources. Then the FMM algorithm is executed in the common way. Finally, the local expansion is evaluated in the cubes of the finest level of the octree. The field, which is caused by elements in the same cube or its neighboring cubes, is obtained by evaluating classical nearly singular integrals.

Though exciting coils can usually be discretized with a coarse mesh, long computation times may also occur when ψ_J has to be computed on the right-hand side in (11). This is, because the line integration (7) has to be carried out for each node of the soft magnetic material. An acceleration of (8), however, is somewhat different to the one described above because the integrand in (8) has vector character. In addition, the differential operator cannot be put in front of the integral. However, the usual substitution of Green's function of its multipole expansion is still possible. Starting from this approach insertion of (16) into (8) gives

$$\boldsymbol{H}(\boldsymbol{r}) = \frac{1}{4\pi}\sum_{n=0}^{L}\sum_{m=-n}^{n} \boldsymbol{M}_{\mathrm{n}}^{\mathrm{m}} \frac{1}{r^{n+1}} Y_n^m(\theta,\varphi) \tag{21}$$

with the coefficients

$$\boldsymbol{M}_{\mathrm{n}}^{\mathrm{m}} = \int_V \boldsymbol{J}(\boldsymbol{r}') \times \nabla_{r'}\left[r'^{n} Y_n^{-m}(\theta',\varphi')\right] \mathrm{d}V'. \tag{22}$$

This approach allows an application of a standard FMM implementation, if each component of (22) is treated as a scalar potential. Threefold execution of the algorithm, one for each component gives the three components of $\boldsymbol{H}_{far}$. The contribution of the near-field $\boldsymbol{H}_{near}$ needs to be computed only once and can be performed during one of the steps to compute a far-field component.

IV. Superconvergent Patch Recovery

$\boldsymbol{H}$ needs to be computed at the nodes of Ω_M after each nonlinear iteration step to iteratively solve the non-linear problem,. Hereby, (14) is usually computed using (10)

$$\boldsymbol{H}(\boldsymbol{r}) = -\sum_{j=1}^{n} \psi_j \nabla N_j(\boldsymbol{r}). \tag{23}$$

With elements of C^0 continuity the gradients at a node of Ω_M are discontinuous. This is usually tackled by taking the mean value of the elements that surround the node. This simple procedure often leads to a slow convergence rate. A more precise approach to compute (23) is the superconvergent patch recovery method (SPR). It has been developed originally to obtain nodal values from solution values at Gaussian points [15]. Hereby a continuous approximation of the solution distribution within a local 'patch' is obtained. Patches are the elements that surround a node where the gradient has to be computed. For better graphical representation, a 2D example of a patch is shown in Fig. 2.

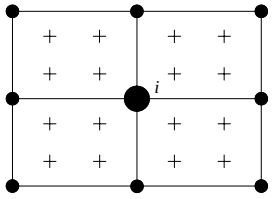

Fig. 2. Local patch with four elements around node i.
The crosses denote positions of Gaussian points.

This has already been extended for structural mechanic problems from two to three dimensions [16]. Hereby an approximated potential distribution $\tilde{\psi}(\boldsymbol{r})$ around a node i is computed by means of values ψ_{Gi} at Gaussian point coordinates $\boldsymbol{r}_{Gi}$ of the N elements that surround this node. The coefficients

$$\boldsymbol{a} = [a_{11}, a_{12}, a_{12}, a_{14}, a_{22}, a_{23}, a_{24}, a_{33}, a_{34}, a_{44}]^T \tag{24}$$

of the polynomial approximation function

$$\tilde{\psi}(x,y,z) = \boldsymbol{P}(x,y,z)\boldsymbol{a}, \tag{25}$$

$$\boldsymbol{P}(x,y,z) = \left[x^2, xy, xz, x, y^2, yz, y, z^2, z, 1\right] \tag{26}$$

are obtained by a 4D regression approach which yields to

$$\sum_{i=1}^{N} \boldsymbol{P}^T(\boldsymbol{r}_{Gi})\boldsymbol{P}(\boldsymbol{r}_{Gi})\boldsymbol{a} = \sum_{i=1}^{N} \boldsymbol{P}^T(\boldsymbol{r}_{Gi})\psi_i. \tag{27}$$

The gradient of (25) is

$$\nabla\tilde{\psi}(x,y,z) = \nabla\boldsymbol{P}(x,y,z)\boldsymbol{a} \tag{28}$$

and therefore (14) can be computed by

$$\boldsymbol{H}(\boldsymbol{r}) = -\nabla\boldsymbol{P}(x,y,z)\boldsymbol{a}. \tag{29}$$

V. Parallelization

Integral equation methods in combination with a matrix compression technique are very powerful. Nevertheless, demands of users grow, too. Furthermore, current

developments of standard computers show that clock speed of processors is not increased any longer but the number of processor cores or the number of processors of a single computer is increased. Even laptops posses a dual core processor.

Parallelization is necessary to keep up with user demands and future computers. Two well established standards for parallelization exist. The message passing interface (MPI) [17] enables communication between multiple processes, which normally run on distributed memory machines, e. g. a PC cluster. A second approach is to split time-consuming parts of a program into several threads based on the OpenMP standard [18]. OpenMP is very easily to implement and recommended on shared memory computers.

A. Parallelization with MPI

A popular parallelization technique is MPI. The program runs in multiple processes. In most cases, one process per processor is started. When the program reaches a parallel region, each process executes only a part of instructions (Fig. 3). Data between the processes along with all communication is done with MPI. Note the whole program must be parallelized for an efficient implementation. Otherwise, CPU time is wasted, since all processes execute exactly the same instructions in the serial region of the program.

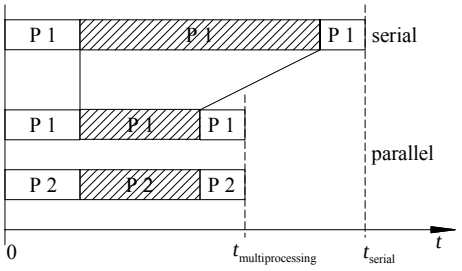

Fig. 3 Illustration of multiprocessing program execution, P *x* is the process number.

The most important advantage of a parallelization with MPI is that memory is split, too. E. g. the system matrix can be divided into multiple blocks which are stored each in one process.

Unfortunately, the software developer must define a parallelization strategy in advance. Hence, it is very difficult to find a good load balance between the processes in practice. A computer with identical nodes is required in most cases and processes must run exclusively on each node.

B. Parallelization with OpenMP

OpenMP was developed in recent years. The main advantage of OpenMP is that parallelization of a well-structured software can be relatively easily done. The program is executed in serial. When a parallel region is reached, the process is split into multiple threads (Fig. 4). The threads are managed by the operating system. Load among the threads is distributed dynamically during runtime.

It suffices to parallelize time consuming functions only. The rest of the program runs in serial on a single CPU. Since a shared memory is used, a data exchange isn't necessary. Hence, a very high speedup is obtained in practice.

This year, cluster OpenMP was introduced [19]. It is based on OpenMP but distributed memory is mapped to a virtual shared memory. Data transfer between the nodes of a cluster is done automatically in contrast to MPI.

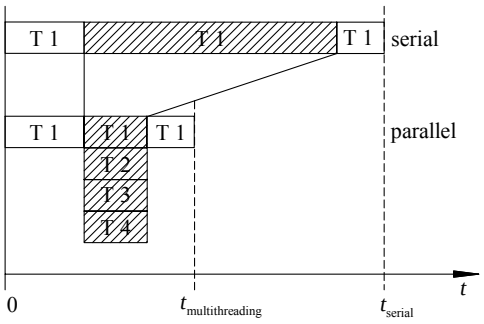

Fig. 4 Illustration of multithreading program execution, T *x* is the thread number.

C. Parallelization of the IEM and FMM

Since every modern computer is equipped with a multi-core CPU or multiple CPUs, OpenMP is always used to solve a numerical field problem.

Assembly of the system matrix is very time consuming. The near-field matrix of the FMM consists of singular and nearly singular integrals. However, parallelization of matrix assembly is very easy. Computation of integrals is distributed dynamically between the threads. Then variations in computing time of singular integrals, which depends strongly on the position of the field point with respect to the element, are absorbed.

The matrix–by-vector product of the FMM is expensive. Furthermore, the amount of transferred data is very high, too. Fortunately, a shared memory can be used and parallelization with OpenMP is very efficient. Implementation is very easily. Dependencies of FMM operations are very well structured. Only a few operations may cause memory write conflicts. These can be avoided by duplicating the very small output array of the matrix-by-vector product.

If a time varying problem is solved, time steps of a quasi-static solution are independent of each other. Hence, they can be distributed among the cluster nodes.

VI. Numerical Results

To demonstrate the applicability of the IEM, the results of three examples are presented. It is shown that the combination of the FMM with parallelized coding techniques leads to a strong reduction of computational costs. In addition, using the SPR for gradient computation improves the accuracy of magnetic field computations. All computations were run on a dual AMD Opteron 248 PC with 2.2 GHz. OpenMP was used to use both CPUs.

A. TEAM Workshop Problem No. 20

The presented IEM has been applied to the static force TEAM Workshop Problem No. 20 shown in Fig. 5 [20]. The ampere-turns are 3000. The yoke and the center pole have been discretized with second-order tetrahedrons and the coil with second-order hexahedrons. 13 non-linear iterations, i. e. 13 linear inhomogeneous problems had to be solved until the stopping criterions $\text{mean}\left(\Delta\chi/\chi\right)<1\%$ and $\max\left(\Delta\chi/\chi\right)<5\%$ were fulfilled. Computing the magnetic field by means of the SPR after a linear problem has been solved took 9s which is negligible compared to the whole

computation time of 14h 37m to solve the problem for the 23589 unknowns. Memory requirements were 591 Mbytes, which equals a compression rate of 86 %.

The magnetic force acting on the center pole has been computed for different ampere-turns with the Maxwell stress tensor method

$$\boldsymbol{F} = \oint_A \mu_0 \left[(\boldsymbol{n}\cdot\boldsymbol{H})\boldsymbol{H} - \frac{\boldsymbol{H}^2}{2}\boldsymbol{n} \right] \mathrm{d}A , \tag{30}$$

which requires integration over a surface A that encloses the center pole. The magnetic field has been computed by using (1) where the induced magnetic field $\boldsymbol{H}_M$ is

$$\boldsymbol{H}_M(\boldsymbol{r}) = \int_{\Omega_M} \frac{\chi(\boldsymbol{r}')}{4\pi} \left[\frac{\nabla\psi(\boldsymbol{r}')}{r_{SO}^3} - \frac{3\nabla\psi(\boldsymbol{r}')\cdot\boldsymbol{r}_{SO}}{r_{SO}^5}\boldsymbol{r}_{SO} \right] \mathrm{d}V , \tag{31}$$

where r_{SO} is the vector from source to observation point and $\boldsymbol{H}_J$ is obtained by (8). Evaluation of (30) took 3m04s, whereupon the surface A was discretized with 5738 nodes. The obtained forces agree well with measurements, which can be seen from Fig. 6.

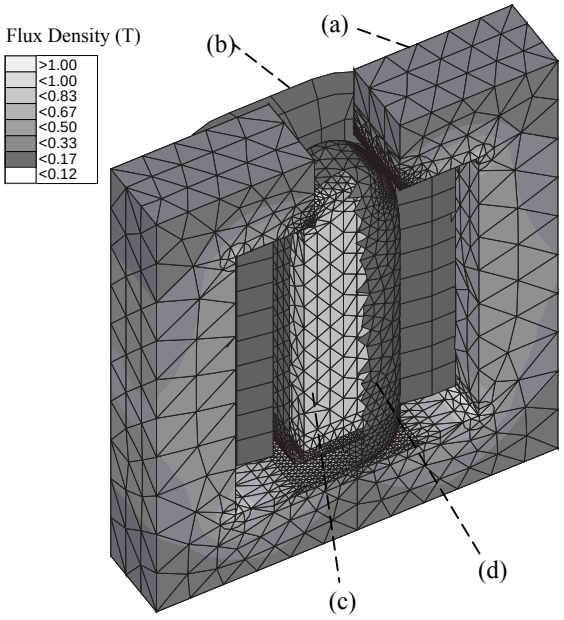

Fig. 5. Mesh of TEAM Workshop Problem No 20 with yoke (a), coil (b), center pole (c) and surface mesh for force computation with Maxwell stress tensor method (d). Some elements where omitted for better graphical representation.

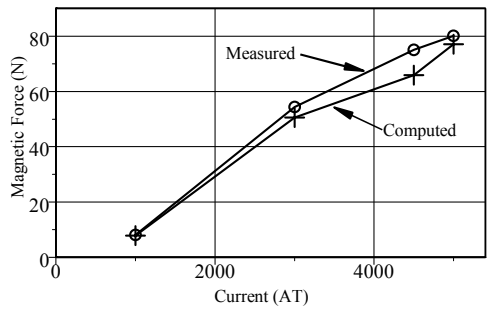

Fig. 6. Computed and measured forces of TEAM Workshop Problem No. 20.

To draw magnetic flux tubes of the setup which are shown in Fig. 7 at 5000AT, the magnetic field was computed at 120631 nodes within the air region by using (1). Instead of the previous mesh, a finer one with 81080 unknowns was used. To compute $\boldsymbol{H}$ with (1) the induced field $\boldsymbol{H}_M$ was computed with the FMM. If $\boldsymbol{H}_J$ is computed by direct evaluation of Biot-Savart law, post-processing takes 2h 33m. If $\boldsymbol{H}_J$ is obtained by means of the FMM, this was reduced to 1h 16m.

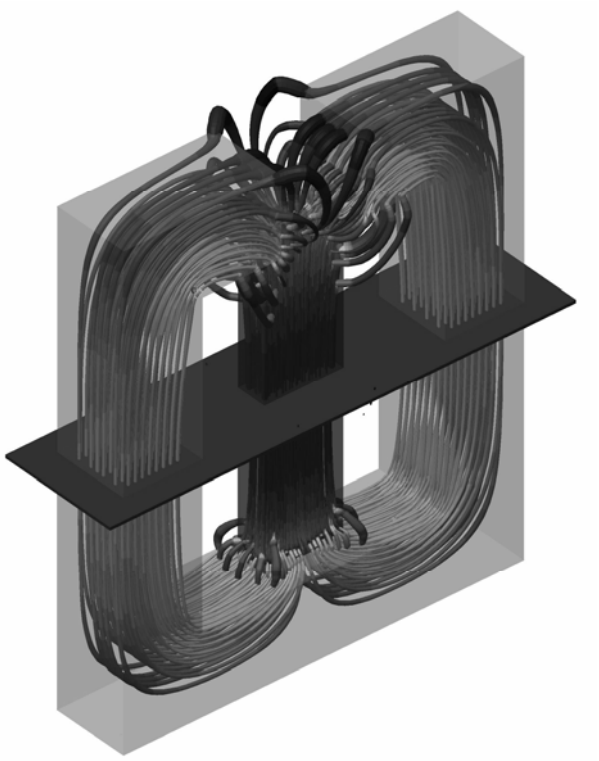

Fig. 7. Magnetic flux tubes.

B. Magnetic Gear

A novel wear-free magnetic gear shown in Fig. 8 has been developed by INDEX-Werke GmbH & Co. KG that allows for converting the low rotational speed of the rigidly coupled modulators to a high rotational speed of the rotor. It consists solely of St37 steel and permanent magnets, i.e. there are no electrical windings. The flat geometrical configuration requires a 3D simulation in order to give respect to stray fields on the front sides. A fine discretization of the permanent magnets with 14742 elements was necessary because of the small air gaps within the gear and in order to obtain compatible meshes at material interfaces. Nevertheless, because the FMM was used to compute ψ_P in (11), assembly of the right-hand side took only 7m instead of 2h 7m without the FMM. The nonlinear problem with 23529 unknowns was solved in 9h 56m and 15 nonlinear iterations. It required 595 bytes of memory. Computing the force acting on the rotor with Maxwell stress tensor method took 2m 56s.

The magnetic flux density $\boldsymbol{B}$ depicted in Fig. 9 was computed within the soft magnetic material from the total scalar potential ψ with

$$\boldsymbol{B} = \mu_0(\chi - 1)\nabla\psi . \tag{32}$$

Therein the gradient was computed with the SPR. $\boldsymbol{B}$ at the surface of a modulator is presented in Fig. 10. It has been computed with SPR and by using shape function derivatives with (23). For comparison, results of a 2D computation [21] are also shown. The field distribution with SPR is smooth and shows no erroneous mavericks.

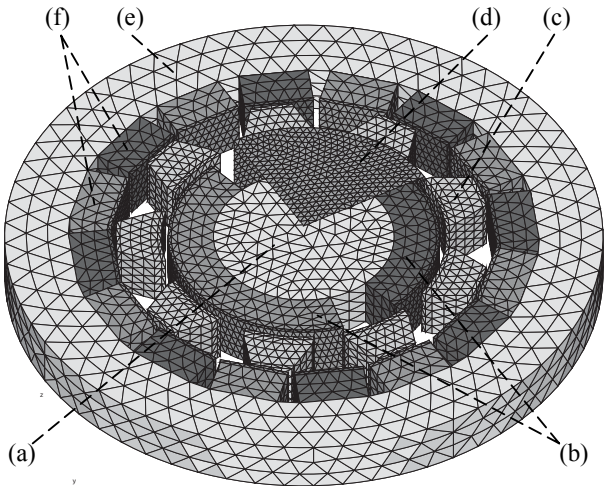

Fig. 8. Magnetic gear with rotor (a), rotor permanent magnets (b), modulators (c), one fourth of the surface for force computation (d), stator (e) and stator permanent magnets (f).

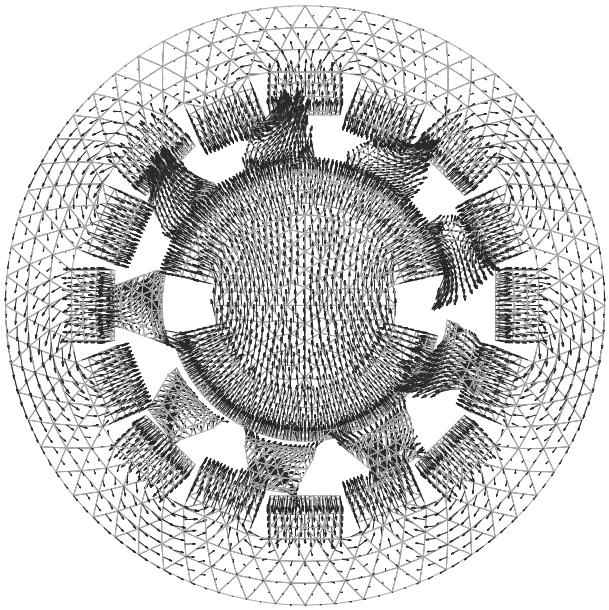

Fig. 9. Magnetic flux density within the magnetic gear.

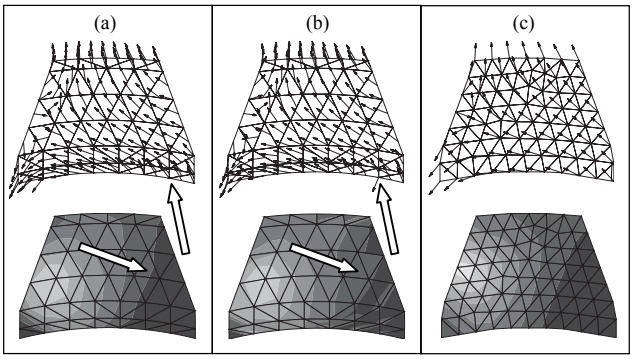

Fig. 10. Magnetic flux density at a modulator by using SPR (a), without SPR (b) and with 2D computation (c).

C. Magnetic circuits for MTXM Microscope

For the Magnetic Transmission X-ray Microscopy Project at BESSY II [22] the field distribution in the air region between the magnetic circuits shown in Fig. 11 was investigated. The current in the coils of the out-of-plane circuit was 10kAT and zero in the coils of the in-plane magnetic circuit. The in-plane pole shoes lead to a disturbance of the field homogeneity in the air gap, which had to be investigated. Solving the problem with 29980 unknowns took 11 nonlinear iterations and took 11h 50m. Memory requirements were 589Mbytes. Computation of the magnetic flux density at 5303 observation points within the air gap required only 28s because the FMM was applied. Despite the large range of element sizes, which can be seen from the blow-up of the air gap in Fig. 12, the number of linear iterations for solving the system of linear equations did not exceed 83.

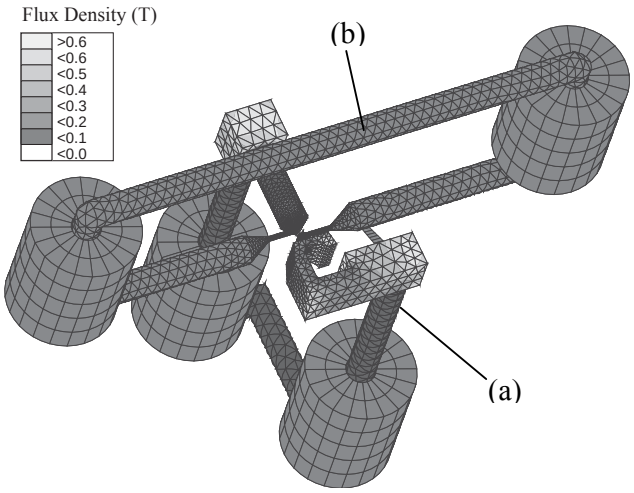

Fig. 11. MTXM setup with out-of-plane magnetic circuit (a), auxiliary mesh (b) and in-plane magnetic circuit (c).

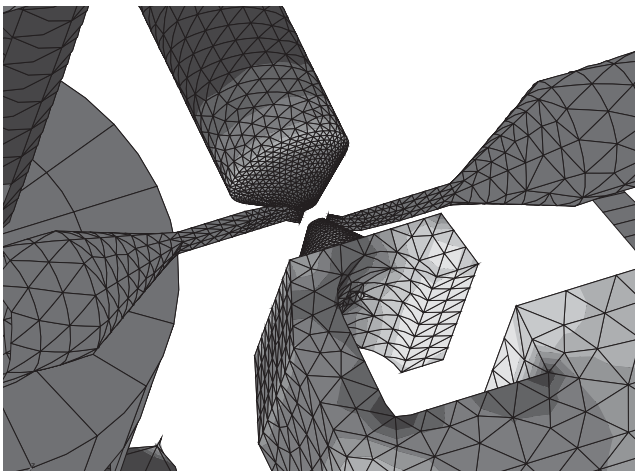

Fig. 12 Blow-up of air gap and surrounding pole-shoes.

CONCLUSIONS

An application of the IEM to non-linear magnetostatic field problems was shown. The IEM in combination with a matrix compression technique like the FMM is very efficient. Only the magnetic parts of the problem must be discretized and a relatively small number of unknowns is obtained. The FMM keeps the memory requirements of the fully dense matrix very small. The superconvergent patch recovery method improves stability of the non-linear iterative solver. Parallelization, especially with OpenMP, makes use of modern computer architectures. Hence, integral equation methods are very efficient and easy to use.

VI. REFERENCES

[1] Greengard, L., and Rokhlin, V., "The rapid evaluation of potential fields in three dimensions", *Lecture Notes in Mathematics*, 1987, pp. 121–141.

[2] Buchau, A., Rucker, W. M., Rain, O., Rischmüller, V. Kurz, S., and Rjasanow, S. "Comparison between different approaches for fast and efficient 3d BEM computations", *IEEE Trans. Magn.*, 2003, vol. 39, no. 3, pp. 1107-1110.

[3] Hafla, W., Groh, F., Buchau, A., and Rucker, W. M., "Magnetostatic Field Computations by an Integral Equation Method Using a Difference Field Concept and the Fast Multipole Method", *Proceedings of the 10 International IGTE Symposium on Numerical Field Calculation in Electrical Engineering*, 2002, pp. 262-266.

[4] Balasubramanian, S., Lalgudi, S. N., and Shanker, B., "Fast-integral-equation scheme for computing magnetostatic fields in nonlinear media", *IEEE Trans. Magn.*, Sept. 2002, vol. 38, no. 5, pp. 3426-3432.

[5] Mayergoyz, I. D., Andrei, P., and Dimian, M., "Nonlinear magnetostatic calculations based on fast multipole method", *IEEE Trans. Magn.*, May 2003, vol. 39, no. 3, pp. 1103-1106.

[6] Hafla, W., Buchau, A., Groh, F., and Rucker, W. M, "Efficient Integral Equation Method for the Solution of 3D Magnetostatic Problems", 2005, *IEEE Trans. Magn.*, vol. 41, no. 5, pp. 1408-1411.

[7] Bertotti, G., "Hysteresis in Magnetism", *Academic Press*, 1998.

[8] Mayergoyz, I. D., Chari, M. V. K., D'Angelo, J., "A new scalar potential formulation for three-dimensional magnetostatic problems", *IEEE Trans. Magn.*, Nov. 1987, vol. MAG-23, no. 6, pp. 3889-3894.

[9] Han, L., Tong, L., "Integral Equation Method Using Total Scalar Potential for the Solution of Linear or Nonlinear 3D Magnetostatic Field With Open Boundary," *IEEE Trans. Mag*, 1994, vol. 30, no. 5, pp. 2897-2900.

[10] Binns, K. J., Lawrenson, P. J., Trowbridge, C. W., "The Analytical and Numerical Solution of Electric and Magnetic Fields," *John Wiley & Sons*, 1992.

[11] Brebbia, C. A., "The boundary element method for engineers", *London: Pentech Press Ltd*, 1978.

[12] A. Buchau, W. Hafla, and W. M. Rucker, "Fast and efficient 3d boundary element method for closed domains", 2004, COMPEL, vol. 23, no. 4, pp. 859-865.

[13] Buchau, A., Hafla, W., Groh, F., Rucker, W. M., "Grouping schemes and meshing strategies for the fast multipole method", *COMPEL*, 2003, vol. 22, no. 3, pp. 495-507.

[14] A. Buchau, W. Rieger, and W. M. Rucker, "Fast Field Computations with the Fast Multipole Method**",**2001, COMPEL, vol. 20, no. 2, pp. 547-561.

[15] O. C. Zienkiewicz and J. Zhu, "The superconvergent patch recovery and a posteriori error estimates. Part I: The recovery technique", *Int. Journal for Numerical Methods in Engineering*, 1992, vol. 33, pp. 1331-1364.

[16] C. Choi, "A 3-d Adaptive Mesh Refinement Using Variable Node Solid Transition Elements", *Int. Journal for Numerical Methods in Engineering*, 1996, Vol. 39, pp. 1585-1606.

[17] Message Passing Interface Forum, *MPI: A Message-Passing Interface Standard*, http://www.mpi-forum.org, June 12 1995.

[18] OpenMP Architecture Board, *OpenMP C and C++ Application Program Interface*, Version 2.0, http://www.openmp.org, 2002.

[19] Cluster OpenMP Workshop, Intel GmbH Munich, High Performance Computing Center Stuttgart, 19th May 2006.

[20] Bíró, O., "Solution of TEAM Benchmark Problem #20 (3-D Static Force Problem)", *Proceedings of the Fourth International TEAM Workshop*, 1993, pp 23–25.

[21] Kurz, S., Fetzer, J., and Lehner, G., "A novel iterative algorithm for the nonlinear BEM-FEM coupling method", *IEEE Trans. Magn.*, 1997, vol. 33, pp. 1772-1775.

[22] Eimüller, T., "Magnetic imaging of nanostructured systems with Transmission X-ray Microscopy", *Dissertation*, *Universität Würzburg*, 2002.

Wolfgang Hafla
André Buchau
Wolfgang M. Rucker
Institute for Theory of Electrical Engineering
University of Stuttgart
Germany