1 SCIENTIFIC COMPUTING TOOLS FOR 3D MAGNETIC FIELD PROBLEMS

Michael Kuhn, Ulrich Langer and Joachim Schöberl

SFB "Numerical and Symbolic Scientific Computing",
Johannes Kepler University Linz, A–4040 Linz, Austria

ABSTRACT

3D magnetic field problems are challenging not only because of interesting applications in the industry but also from the mathematical point of view. In order to handle technical magnetic field problems efficiently, it is not sufficient to have a fast Maxwell solver, but also a geometry modeller, an advanced 3D mesh generator, mesh handling and refinement methods, parallelization tools, and postprocessing tools including advanced visualization techniques are required. We will present such pre- and postprocessing tools, especially adapted to adaptive multilevel methods used in the solver. In the magnetostatic case, the Maxwell solver is based on special mixed variational formulations of the Maxwell equations. In combination with an adaptive nested multilevel preconditioned iteration approach, we obtain an optimal solver with respect to the complexity. This is confirmed by the results of numerical experiments presented. We also propose a concept for coupling finite elements with boundary elements. Coupled finite and boundary element schemes are especially suited for problems where it is necessary to compute the exterior magnetic field.

Key words. Maxwell's equations, finite element method, boundary element method, multigrid preconditioners, domain decomposition, scientific computing tools, geometry modelling, mesh generation, visualization, parallelization

1.1 INTRODUCTION

Usually, technical 3D electromagnetic field problems are characterized by complicated interface geometries with possibly moving parts (e.g. rotating parts), large jumps in the coefficients, nonlinearities, singularities, and the necessity of calculating the exterior magnetic field. In practice, the aim of the simulations is often the optimization of the magnetic device we are concerned with. Coupled field problems in which the electromagnetic fields interact with other fields, e.g. with mechanical fields, lead to problems of high computational complexity. In order to handle such kind of technical electromagnetic field problems, it is not sufficient to have a fast Maxwell solver and optimizer, but also a geometry modeller, an advanced 3D mesh generator, mesh handling and refinement methods, parallelization tools, and postprocessing tools including advanced visualization techniques are required.

We will present such pre- and postprocessing tools, especially adapted to adaptive multilevel methods used in the solver and the optimizer. Of course, the heart of our object oriented 3D field problem solver environment FEPP is an adaptive multilevel solver. In the magnetostatic case, the Maxwell solver is based on special mixed variational formulations of the Maxwell equations in $\mathrm{H}_0(\mathbf{curl}) \times \mathrm{H}_0^1(\Omega)$ and their discretization by the lowest order Nédélec and Lagrange finite elements, respectively. This mixed finite element scheme can be reduced to a symmetric and positive definite (spd) scheme for defining the vector potential introduced for the B-field. A clever combination of a multigrid regularisator with an adaptive nested multilevel preconditioned iteration approach for the spd system leads to an optimal solver with respect to the complexity. This is confirmed by the results of numerical experiments for academical problems and real life applications as well. Let us mention that multilevel preconditioners for Maxwell's equations have also been studied in [1, 15]. In [8], h-p–adaptive finite element methods were studied for Maxwell's equations. Similar to the 2D case [7, 16, 21], we also propose a domain decomposition concept for coupling finite elements with boundary elements that leads to a symmetric, but indefinite scheme. Coupled finite and boundary element schemes are especially suited for problems where it is necessary to take into account the exterior magnetic field. However, as in 2D, it may be advantageous to use the boundary element technique also in other air subdomains, e.g. in order to separate moving parts. In the finite element subdomains we introduce some vector potential for the B-field whereas scalar potentials for the H-field are used in the boundary element subdomains. Another approach to the coupled finite and boundary element approximation of the magnetostatic Maxwell equations was studied in [2].

For the parallelization, two different strategies have been developed. The first approach uses a thread-based implementation that is especially suited for shared memory parallel computers such as the ORIGIN 2000. It is highly efficient if small numbers of processors are used. The second concept is based on distributed data algorithms and has been developed for massively parallel computers and workstation (PC) clusters.

The rest of the paper is organized as follows. In Section 2, we recall the Maxwell equations on which the numerical magnetic field simulations are based. Section 3 is devoted to the numerical solution of 3D magnetostatic field problems. In Section 4, we present the scientific computing tools that are necessary for treating more advanced 3D field problems. Section 5 contains some results of our numerical experiments. Finally, we add some concluding remarks in Section 6.

More information on the problem solver environment FEPP and on the presented results can be obtained from the homepage http://www.sfb013.uni-linz.ac.at of the special research project (SFB) "Numerical and Symbolic Scientific Computing" that is supported by the Austrian Science Foundation (FWF) under the grant SFB F013.

1.2 THE MAXWELL EQUATIONS

The Maxwell equations

$$\mathrm{curl}(H) = J + \frac{\partial D}{\partial t}, \;\; \mathrm{div}(B) = 0, \;\; \mathrm{curl}(E) = -\frac{\partial B}{\partial t}, \;\; \mathrm{div}(D) = \varrho \qquad (1.1)$$

couple the magnetic fields H (magnetic field intensity) and B (magnetic flux density) with the electric fields E (electric field intensity), D (electric flux density), J (electric current density) and ϱ (electric charge density). The partial differential equations (1.1) must be completed by constitutive relationships, initial and boundary conditions. In the case of isotropic media, the constitutive relationships have the form

$$J = \sigma E, \;\; B = \mu H, \;\; D = \varepsilon E, \qquad (1.2)$$

where σ, μ, ε denote the conductivity, the permeability, and the permittivity, respectively. In more general cases including anisotropic and inhomogeneous materials, these coefficients are tensors, where the single components depend on the locus (inhomogeneous materials) as well as on certain field quantities (nonlinear case). If moving bodies are involved, one has to add a term of the form $\mathrm{curl}(v \times B)$ to the right hand side of the third equation in (1.1), where v is the velocity [19, 23]. We come back to this case at the end of this paper in connection with coupled magnetomechanical problems. In the following, we restrict our considerations to the linear stationary case. Then, the equations describing the electric and magnetic fields can be separated. The magnetostatic equations, in which we are interested from now on, can be rewritten as

$$\mathrm{curl}(H) = J, \;\; H = \nu B, \;\; \mathrm{div}(B) = 0, \qquad (1.3)$$

where $\nu = 1/\mu$ and the current density J is supposed to be given. If permanent magnets are present, J has to be replaced by $J + \mathrm{curl}(H_0)$, where H_0 is the initial (permanent) magnetic field intensity. Furthermore, we note that the current density J is divergence-free by physical reasons, i.e.,

$$\mathrm{div}(J) = 0. \qquad (1.4)$$

Theoretically, the computational domain Ω coincides with the space R^3 in any case. The behaviour of the magnetic field at infinity is described by radiation conditions. In practice, one may often simplify the problem by considering a bounded, simply connected computational domain $\Omega \subset R^3$ with Lipschitz boundary $\Gamma = \partial\Omega$ and by replacing the radiation condition by the boundary condition

$$B \cdot n = 0 \quad \text{on } \partial\Omega, \qquad (1.5)$$

where n stands for the unit outward normal with respect to $\partial\Omega$. Of course, this modification causes some modelling error. However, if the sources are completely surrounded by iron or, more generally, if Ω is chosen large enough, this error can be neglected. In Subsection 1.3.4, we will use the Boundary Element Method in order to avoid the restriction of the problem to bounded domains and thus the definition of artificial boundary

conditions. Instead, the physical radiation conditions are modelled exactly.
Introducing some vector potential u for the B-field $B = \mathrm{curl}(u)$ [9], we can easily derive the canonical variational formulation

$$\text{Find } u \in H_0(\mathrm{curl}, \Omega) : \int_\Omega \nu \, \mathrm{curl}(u) \cdot \mathrm{curl}(v) \, dx = \int_\Omega J \cdot v \, dx \quad \forall\, v \in H_0(\mathrm{curl}, \Omega) \tag{1.6}$$

from the magnetostatic equations (1.3) in Ω and the boundary conditions (1.5) on Γ, where $H_0(\mathrm{curl}, \Omega) := \{v \in L^2(\Omega)^3 | \ \mathrm{curl}(v) \in L^2(\Omega)^3, \ v \times n = 0 \text{ on } \Gamma\}$. Condition (1.4) ensures the solvability of the variational problem (1.6). However, the solution is not unique. Indeed, if one adds to some solution u of (1.6) a gradient-field $\nabla\varphi \in L^2(\Omega)^3$ with $\nabla\varphi \times n = 0$, one obtains again a solution.
From now one we allow the coefficient $\nu = 1/\mu$ to be piecewise constant. We mention that on the interfaces the normal component of the B-field and the tangential component of the H-field are continuous.

1.3 3D MAGNETOSTATIC FIELD PROBLEMS

1.3.1 Mixed FE Discretization

In the previous section we observed that the solution of the canonical variational formulation (1.6) is not unique. In order to obtain some variational formulation that has a unique solution, we must take into account some gauging condition for the vector potential u. The Coulomb gauging

$$\mathrm{div}(u) = 0. \tag{1.7}$$

aims at divergence-free vector potentials. Treating the weak formulation of the Coulomb gauging (1.7) as an equality constraint, we arrive at the following mixed variational formulation that is fundamental for our approach to the numerical solution of the magnetostatic Maxwell equations (1.3): Find $(u, p) \in X \times M := H_0\,(\mathrm{curl}, \Omega) \times H_0^1(\Omega)$ such that

$$\begin{aligned} a(u,v) + b(v,p) &= \langle f, v\rangle \quad \forall\, v \in H_0\,(\mathrm{curl}, \Omega), && (1.8)\\ b(u,q) &= 0 \qquad\quad \forall\, q \in H_0^1(\Omega), && (1.9) \end{aligned}$$

where $a(u,v) := \int_\Omega \nu \, \mathrm{curl}(u) \cdot \mathrm{curl}(v) \, dx$, $b(v,p) := \int_\Omega v \cdot \nabla p \, dx$, and $\langle f, v\rangle := \int_\Omega J \cdot v \, dx$. Now it is not difficult to conclude from the Brezzi-Babuška theory that the mixed variational problem (1.8) - (1.9) has an unique solution. Moreover, choosing $v = \nabla p \in H_0\,(\mathrm{curl}, \Omega)$ in (1.8), we immediately observe from (1.4) and the Friedrichs inequality that

$$p = 0. \tag{1.10}$$

This simple observation is crucial for our approach. Indeed, adding to the second equation of our mixed variational problem (1.8) - (1.9) an arbitrary symmetric and positive definite (spd) bilinear form $c(\cdot,\cdot) : M \times M \to R^1$, we arrive at the equivalent mixed variational problem: Find $(u, p) \in H_0\,(\mathrm{curl}, \Omega) \times H_0^1(\Omega)$ such that

$$\begin{aligned} a(u,v) + b(v,p) &= \langle f, v\rangle \quad \forall\, v \in H_0\,(\mathrm{curl}, \Omega), && (1.11)\\ b(u,q) - c(p,q) &= 0 \qquad\quad \forall\, q \in H_0^1(\Omega). && (1.12) \end{aligned}$$

Let now be $X_h := \mathcal{N}_h^1 \subset X$ and $M_h := \mathcal{S}_h^1 \subset X$ the lowest order edge element space (see [25]) and the space of piecewise linear nodal elements on a shape-regular tetrahedral triangularization of Ω, respectively [6], where h is some discretization parameter. Then the mixed finite element approximation to the regularized mixed variational problem (1.11) - (1.12) reads as follows: Find $(u_h, p_h) \in X_h \times M_h$ such that

$$\begin{aligned} a(u_h, v_h) + b(v_h, p_h) &= \langle f, v_h \rangle \quad \forall\, v_h \in X_h, && (1.13)\\ b(u_h, q_h) - c(p_h, q_h) &= 0 \qquad\quad\; \forall\, q_h \in M_h. && (1.14) \end{aligned}$$

Again, using the Brezzi-Babuška theory, we conclude that the mixed finite element scheme (1.13) - (1.14) has an unique solution. From the same theory, we can derive the standard a–priori error estimates (see, e.g., [4]). Similarly to the continuous case, we observe that $p_h = 0$ since $q_h \in M_h$ implies $\nabla q_h \in X_h$.
Let us choose the standard finite element basis in the spaces X_h and M_h. Then the mixed finite element scheme (1.13) - (1.14) is equivalent to the regular, symmetric, but indefinite system

$$\begin{pmatrix} A & B^T \\ B & -C \end{pmatrix} \begin{pmatrix} \underline{u}_h \\ \underline{p}_h \end{pmatrix} = \begin{pmatrix} \underline{f}_h \\ 0 \end{pmatrix} \tag{1.15}$$

of linear finite element equations, where the matrices A, B, C and the first component $\underline{f}_h$ of the right-hand side are derived from the bilinear forms $a(\cdot,\cdot)$, $b(\cdot,\cdot)$, $c(\cdot,\cdot)$, and the linear form $\langle f, \cdot \rangle$, respectively.

1.3.2 SPD Reduction

Eliminating $\underline{p}_h = C^{-1} B \underline{u}_h$ from the second equation in (1.15) and inserting it into the first equation, we obtain the spd Schur complement system

$$G \underline{u}_h := (A + B^T C^{-1} B) \underline{u}_h = \underline{f}_h. \tag{1.16}$$

Let $\tilde{C}$ be some spd matrix that is spectrally equivalent to C (briefly, $\tilde{C} \approx C$), i.e. there are some positive constants c_1 and c_2, independent of h, such that

$$c_1 (\tilde{C} \underline{q}_h, \underline{q}_h) \le (C \underline{q}_h, \underline{q}_h) \le c_2 (\tilde{C} \underline{q}_h, \underline{q}_h) \quad \forall \underline{q}_h \in R^{m_h}, \tag{1.17}$$

where $m_h := dim(M_h)$. Then the original Schur complement system (1.16) is equivalent to the modified Schur complement system

$$\tilde{G} \underline{u}_h := (A + B^T \tilde{C}^{-1} B) \underline{u}_h = \underline{f}_h. \tag{1.18}$$

Instead of solving the symmetric, but indefinite system (1.15), we solve the spd modified Schur complement system (1.18). We recall that the second component $\underline{p}_h$ is equal to 0. Furthermore, we can play around with the regularisator $\tilde{C}$ in order to improve the the properties of the system matrix $\tilde{G}$ with respect to the solver we are going to construct now.

1.3.3 Solver

Let us choose the spd bilinear form

$$c(p,q) := \frac{1}{\nu_{\min}} \int\limits_{\Omega} \nabla p \, \nabla q \, dx, \tag{1.19}$$

corresponding to the Laplace operator scaled by $1/\nu_{\min}$ (other choices for the $c(\cdot,\cdot)$ are discussed in [20]), and let us consider a spd preconditioner C_H for the spd matrix $H := A + M$, i.e. there are some positive constants c_3 and c_4, independent of h, such that

$$c_3(C_H \underline{v}_h, \underline{v}_h) \le (H\underline{v}_h, \underline{v}_h) \le c_4(C_H \underline{v}_h, \underline{v}_h) \quad \forall \underline{v}_h \in R^{n_h}, \tag{1.20}$$

where $n_h := dim(X_h)$ and M is the mass matrix in X_h defined by

$$(M\underline{u}_h, \underline{v}_h) := \nu_{\min} \int\limits_{\Omega} u_h \, v_h \, dx.$$

The discrete LBB–condition and the spectral equivalence inequalities (1.20) imply that $C_H \approx H \approx G \approx \tilde{G}$ (see [20] for the detailed proof). Once a good preconditioner C_H and an appropriate regularisator $\tilde{C}$ is available, we can solve the modified Schur complement system (1.18) by the Preconditioned Conjugate Gradient (PCG) method. In practice, we choose the multigrid preconditioner

$$C_H := H(I - M_H)^{-1} \tag{1.21}$$

and the multigrid regularisator

$$\tilde{C} := C_C := C(I - M_C)^{-1}, \tag{1.22}$$

where M_H and M_C are the corresponding multigrid iteration operators with respect to H and C. Choosing appropriate symmetric multigrid cycles, we can now conclude from the results of [1, 12, 15] that the PCG method is asymptotically optimal with respect to the operation count and to the memory demand [17]. This is confirmed by the numerical results presented in Section 1.5.

1.3.4 Symmetric Coupling of FEM and BEM

If one is interested in the exterior magnetic field, then the coupling of the FEM with the BEM is certainly the natural technique to handle this problem. For simplicity of the presentation, let us consider the case where the magnetic sources and the ferromagnetic materials are located in some bounded and simply connected Lipschitz domain Ω_F where we will use the FEM for approximating the magnetic field. Thus, we suppose that in the exterior BEM subdomain $\Omega_B := (\bar{\Omega}_F)^c$ the electric current density vanishes, i.e., $J = 0$, and $\nu = \nu_B = 1/\mu_0$ (air). We can again introduce the vector potential u for the B–field $B = \mathrm{curl}(u)$ in Ω_F. Furthermore, we now assume that Ω_F is chosen such that Ω_B is simply connected too. Then in the exterior domain Ω_B, the H–field can be represented as a gradient field of some scalar potential φ, i.e. $H = \mathrm{grad}(\varphi)$ in Ω_B.

Therefore, in the exterior subdomain Ω_B, the magnetostatic Maxwell equations (1.3) are essentially reduced to the scaled Laplace equation for the scalar potential φ. The Cauchy data for the solution of this equation are related by Calderon's integral equation

$$\begin{pmatrix}\varphi\\ \lambda\end{pmatrix} = \mathcal{C}\begin{pmatrix}\varphi\\ \lambda\end{pmatrix} := \begin{pmatrix} \frac{1}{2}\mathcal{I}+\mathcal{K} & -\nu\mathcal{V} \\ -\frac{1}{\nu}\mathcal{D} & \frac{1}{2}\mathcal{I}-\mathcal{K}^* \end{pmatrix}\begin{pmatrix}\varphi\\ \lambda\end{pmatrix} \tag{1.23}$$

on the interface $\Gamma := \partial\Omega_F = \partial\Omega_B$, where φ denotes the trace of the scalar potential on Γ, $\lambda = \frac{1}{\nu}\frac{\partial\varphi}{\partial n} = B\cdot n$ on Γ, $n :=$ outer unit normal to Ω_F, $\mathcal{V} :=$ single layer potential operator on Γ, $\mathcal{K} :=$ double layer potential operator on Γ, $\mathcal{D} :=$ hypersingular operator on Γ.

Using now Coulumb's gauging condition (1.7) in Ω_F and Cauchy's representation formula (1.23) of the Cauchy data together with the interface condition predicting the continuity of the tangential part $H \times n$ of the H–field and of the normal component $B \cdot n$ of the B–field, we arrive at the mixed coupled FE-BE variational formulation: Find $(u, \varphi, p) \in V := U \times \Phi \times P$ such that:

$$\begin{aligned} a(u,\varphi;v,\psi) + b(v,p) &= \langle f, v\rangle \quad \forall\,(v,\psi) \in U\times\Phi, &(1.24)\\ b(u,q) - c(p,q) &= 0 \qquad\quad \forall\, q \in P, &(1.25)\end{aligned}$$

where $U := H(\mathrm{curl}, \Omega_F)$, $\Phi := H_\star^{1/2}(\Gamma)$, and $P := H_\star^1(\Omega_F)$. The bilinear forms are defined by the identities

$$\begin{aligned} a(u,\varphi;v,\psi) &:= \int_{\Omega_F} \nu\, \mathrm{curl}(u)\cdot\mathrm{curl}(v)\,dx \\ &\quad + \langle \nu_B \mathcal{V}(\mathrm{curl}(u)\cdot n), \mathrm{curl}(v)\cdot n\rangle_\Gamma - \langle (\frac{1}{2}\mathcal{I}+\mathcal{K})\varphi, \mathrm{curl}(v)\cdot n\rangle_\Gamma \\ &\quad + \langle \frac{1}{\nu_B}\mathcal{D}\varphi, \psi\rangle_\Gamma + \langle (\frac{1}{2}\mathcal{I}+\mathcal{K}^*)(\mathrm{curl}(u)\cdot n), \psi\rangle_\Gamma, \end{aligned}$$

$$b(u,q) := \int_{\Omega_F} u\cdot\nabla q\,dx, \quad c(p,q) := \int_{\Omega_F} \nabla p\cdot\nabla q\,dx, \quad \langle f, v\rangle := \int_{\Omega_F} J\cdot v\,dx.$$

The subscribe "$\star$" means that the function of the corresponding space should be L_2–orthogonal to the constant functions. We mention that $H_\star^1(\Omega_F)$ can be changed to $H^1(\Omega_F)$ if the above bilinear form $c(\cdot,\cdot)$ is replaced by the bilinear form

$$c(p,q) := \int_\Omega (\alpha\nabla p\,\nabla q + \beta\, p\, q)\,dx, \tag{1.26}$$

that may be easier to handle in practice, where α and β are appropriately chosen, positive scaling constants. Again, in both cases, one can show existence and uniqueness of the solution. Moreover, $p = 0$ if $\int_{\Omega_F} J\cdot\nabla q\,dx = 0 \quad \forall\, q \in H_\star^1(\Omega_F)$ (see [20] for the proof).

Formally, the mixed domain and boundary integral variational formulation (1.24)-

(1.25) is equivalent the problem

$$
\begin{aligned}
\operatorname{curl}(\nu \operatorname{curl}(u)) &= J \quad \text{in} \quad \Omega_F \quad \text{(PDE)},\\
\operatorname{div}(u) &= 0 \quad \text{in} \quad \Omega_F \quad \text{(gauging condition)},\\
\operatorname{div}(\mu_B \nabla\varphi) &= 0 \quad \text{in} \quad \Omega_B \quad \text{(PDE)},\\
\varphi &= O(1/|x|) \quad \text{for} \quad |x| \to \infty \quad \text{(radiation condition)},\\
\nabla\varphi \times n &= \nu_F(\operatorname{curl}(u) \times n) \quad \text{(interface condition: } H \times n \text{ continuous)},\\
\lambda := \mu_B(\nabla\varphi \cdot n) &= \operatorname{curl}(u) \cdot n \qquad \text{(interface condition: } B \cdot n \text{ continuous)},
\end{aligned}
$$

that corresponds to the magnetostatic Maxwell equations in R^3.

Choosing the finite (boundary) element subspaces $U_h := \mathcal{N}_h^1 \subset U$, $\Phi_h := \mathcal{S}_h^1 \subset \Phi$ and $P_h := \mathcal{S}_h^1 \subset P$ in analogy to Subsection 1.3.1, we derive from (1.24) the symmetric coupled FE-BE Galerkin scheme: Find $(u_h, \varphi_h, p_h) \in V_h := U_h \times \Phi_h \times P_h$ such that

$$
\begin{aligned}
a(u_h, \varphi_h; v_h, \psi_h) + b(v_h, p_h) &= \langle f, v_h\rangle \quad \forall\, (v_h, \psi_h) \in U_h \times \Phi_h, &(1.27)\\
b(u_h, q_h) - c(p_h, q_h) &= 0 \qquad\quad \forall\, q_h \in P_h, &(1.28)
\end{aligned}
$$

that is again equivalent to the following symmetric, but indefinite system of coupled FE-BE equations

$$
\begin{pmatrix} A_h & K_h^T & B_h^T \\ K_h & -D_h & 0 \\ B_h & 0 & -C_h \end{pmatrix}
\begin{pmatrix} \underline{u}_h \\ \underline{\varphi}_h \\ \underline{p}_h \end{pmatrix}
=
\begin{pmatrix} \underline{f}_h \\ 0 \\ 0 \end{pmatrix}, \tag{1.29}
$$

where $A_h = A_h^F + A_h^B$ consists of the contributions from the first two terms of the bilinear form $a(\cdot,\cdot)$. Eliminating again $\underline{p}_h = C^{-1}B\underline{u}_h$ from the third equation in (1.29) and inserting it into the first equation, we obtain the Schur complement system

$$
\begin{pmatrix} \tilde{A}_h & K_h^T \\ K_h & -D_h \end{pmatrix}
\begin{pmatrix} \underline{u}_h \\ \underline{\varphi}_h \end{pmatrix}
=
\begin{pmatrix} \underline{f}_h \\ 0 \end{pmatrix},
$$

where $\tilde{A}_h = A_h^F + A_h^B + B_h^T C_h^{-1} B_h$. In contrast to the finite element case considered in Subsection 1.3.2, here the Schur complement system remains symmetric and indefinite. Similar to the 2D case discussed in [5, 21, 24], we can now construct efficient solvers on the basis of the Bramble-Pasciak transformation [3, 30].

1.4 SCIENTIFIC COMPUTING TOOLS

In this section we discuss the scientific computing tools that are required for the solution of general field problems in 3D. We apply these tools to the simulation of Maxwell's equations introduced in Section 1.3.

The interplay of these computing tools is illustrated in Figure 1.1. Clearly, the central module is the simulation code. Usually, this is the most CPU-time consuming part of the whole process, and it is a topic of intensive research. The input to the simulation module is the description of the problem which is to be solved. This includes the computational

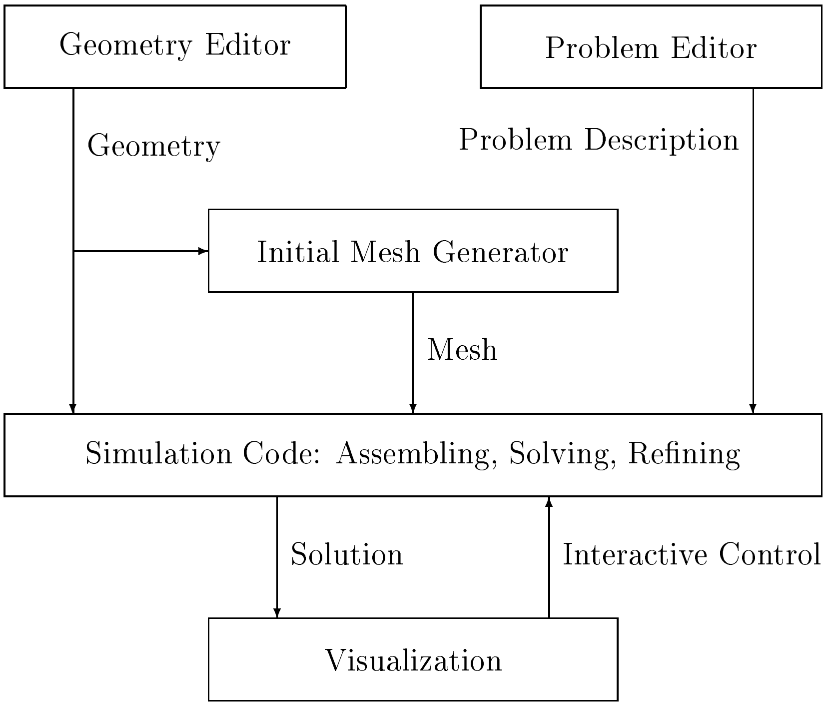

Figure 1.1: Interplay of Scientific Computing Tools.

domain, the underlying equations, the boundary conditions etc. These input data have to be described by proper data structures. Therefore, the user should have access to tools for an interactive generation of the geometry as well as the problem description. These tools are surely different for scientific computing and for commercial applications. For scientific processes, the tool may be just based on a text editor. Beside the small amount of data for geometry and equations, a mesh is needed for the simulation. Since the mesh data are much more complex, an automatic tool for mesh generation is needed. The simulation produces a nearly arbitrarily large amount of data. For the output of the information, advanced visualization tools are necessary.

We will describe the tools developed within the research project "Numerical and Symbolic Scientific Computing", namely the mesh generator NETGEN, the simulation code FEPP and the visualization module VIPP. As well, the data structures for the geometry representation and the problem description are described by means of a specific example.

1.4.1 Geometry Description

There are many different techniques for describing the geometry of domains. Their advantages and disadvantages depend on the specific application we are dealing with. For example, a surface representation is useful for visualization of virtual reality environments, but for simulation purposes solid representations are necessary. Surfaces can be described by parametric classes, like a cylinder with two points on the axis and a radius, or by approximation with spline patches. While the spline approximation is very flexible

```
solid kernel =
               orthobrick (-8, -6, -1; 8, 6, 1)
       and not orthobrick (-6, -4, -2; -1, 4, 2)
       and not orthobrick ( 1, -4, -2;  6, 4, 2);

solid coil1 =
                cylinder (-7, -3, 0;-7, 3, 0; 3)
        and not cylinder (-7, -3, 0;-7, 3, 0; 2)
        and plane (0, -3, 0; 0, -1, 0)
        and plane (0,  3, 0; 0,  1, 0);

solid coil2 =
                cylinder ( 0, -3, 0; 0, 3, 0; 3)
        and not cylinder ( 0, -3, 0; 0, 3, 0; 2)
        and plane (0, -3, 0; 0, -1, 0)
        and plane (0,  3, 0; 0,  1, 0);

solid coil3 =
                cylinder ( 7, -3, 0; 7, 3, 0; 3)
        and not cylinder ( 7, -3, 0; 7, 3, 0; 2)
        and plane (0, -3, 0; 0, -1, 0)
        and plane (0,  3, 0; 0,  1, 0);

solid box = orthobrick (-12, -8, -5; 12, 8, 5);

solid air = box and not (kernel or coil1 or coil2 or coil3);
```

Figure 1.2: Geometry description for the transformer.

for describing complicated geometries like car bodies, human bodies or geodetic data, it requires special tools for the generation of the geometry description itself.

Many realistic geometries like the transformer shown below can be described by a Constructive Solid Geometry (CSG) model very easily. An open number of parametric classes forms primitives. These primitives are combined by the Eulerian operations *union*, *intersection* and *complement* to form more complicated solids. It is of moderate effort to define such geometries by means of a text editor.

Figure 1.2 contains the whole description of the geometry of the transformer. It contains five subdomains, namely the kernel, three coils and the air domain around these parts. The outer boundary is given by an iron casing of high conductivity that motivates the boundary condition (1.5). The file is more or less self-explaining. Primitives are, e.g., cylinders of infinite length defined by two points on the axis and the radius, or a brick in normal direction defined by a minimal and a maximal point. For example, one coil is defined by the outer cylinder, from which an inner cylinder is cut off, and by two planes which restrict the infinite cylinder to a finite length. The geometry is drawn in Figure 1.3. Fast geometry visualization is done by NETGEN on the basis of (nonconforming) triangle approximations.

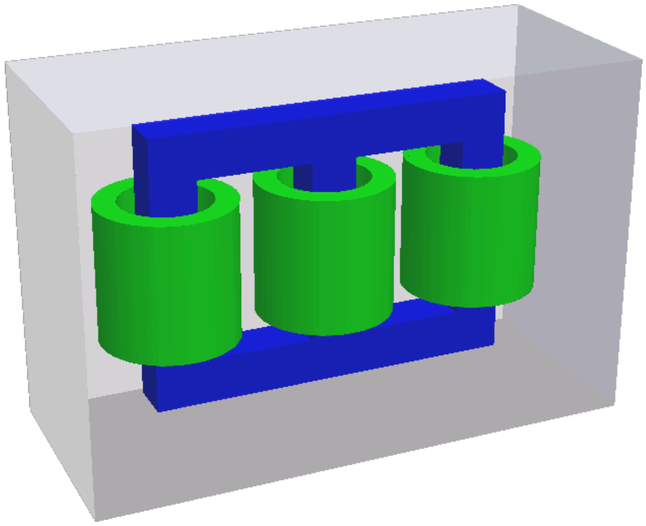

Figure 1.3: Geometry visualization of the transformer.

1.4.2 The 3D Mesh Generator NETGEN

A mesh generator is required to generate a triangulation based on the geometry description. We now explain the principal parts of the fully automatic tetrahedral mesher NETGEN [27]. Figure 1.4 shows the sequence of steps performed by NETGEN. The principle is bottom up, from 0-dimensional objects (points) to 3-dimensional objects (tetrahedra). First, one has to detect all corner points (and other special points) of the geometry. These points are defined by nonlinear equations. The algorithm used here is a combination of a globally convergent bisection method and a fast, locally convergent Newton method. Starting from the corner points, the edges are discovered by curve following methods.

After handling the edges, the surfaces are meshed. This is done by a version of the advancing front method. Starting from the boundary of a surface in 3D, triangle by triangle is cut off to reduce the remaining part of the surface. The principle is shown in Figure 1.5. The rules defining how to generate triangles for a given boundary constellation are described by data structures instead of hard-coded instructions. This design decision keeps the code relatively short and maintainable.

The last part is the volume mesh generation. Originally, NETGEN used the same advancing front method for volume meshing as for surface meshing. Together with a back-tracking strategy the algorithm is very robust. The drawback of this approach is that it is very time consuming. Now, we use a combination of the Delaunay algorithm with the rule based advancing front method. The Delaunay algorithm is very fast, but often it fails to generate the whole volume mesh compatible to the boundary mesh. Then, a small number of local zones are left. These are filled out by the original, robust rule based advancing front method.

From the geometry description of the transformer given in Figure 1.2, NETGEN generated a coarse mesh with 2465 tetrahedra as shown in Figure 1.6.

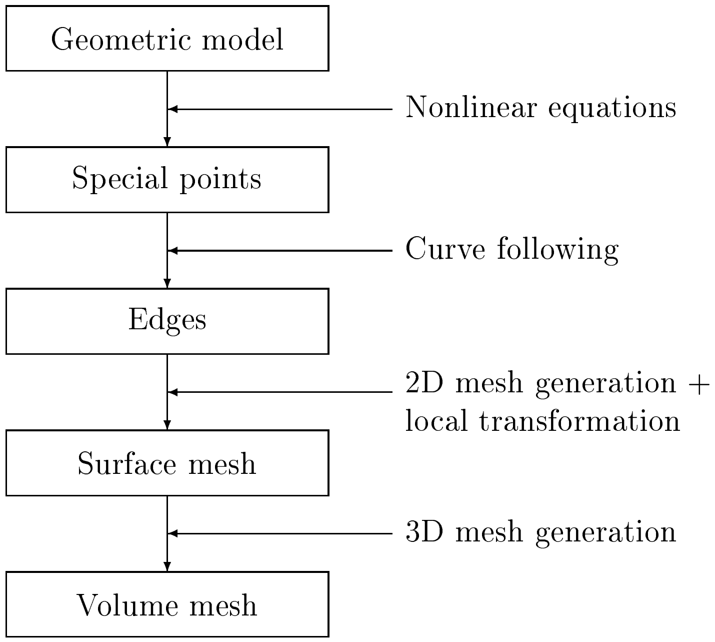

Figure 1.4: Subproblems in mesh generation.

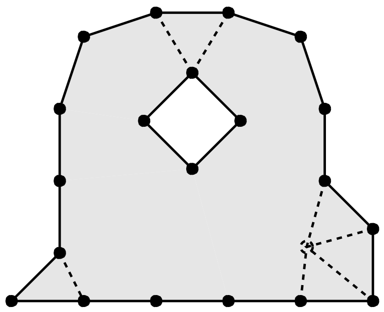

Figure 1.5: Advancing Front Method: cutting off triangles.

1.4.3 Problem and Solver Description

The simulation module FEPP is designed to handle different types of equations with one code - even within one run. This is achieved by a problem description language (see Figure 1.7 for our model problem). It enables us to define an arbitrary number of finite element spaces, gridfunctions, bilinear forms or linear forms. By certain flags the objects can be adapted to the problem. For Maxwell equations we use edge elements. This is indicated by the `-edge` flag in the definition of the finite element space called `-vedge`.

The bilinear form is built up from a number of predefined forms, e.g., the token `rotrotedge` defines the form

$$\int_\Omega \nu(x)\,\mathrm{curl}(u) \cdot \mathrm{curl}(v)\,dx.$$

The coefficient $\nu(x)$ can be given as constant, as piecewise constant, by a generic formula

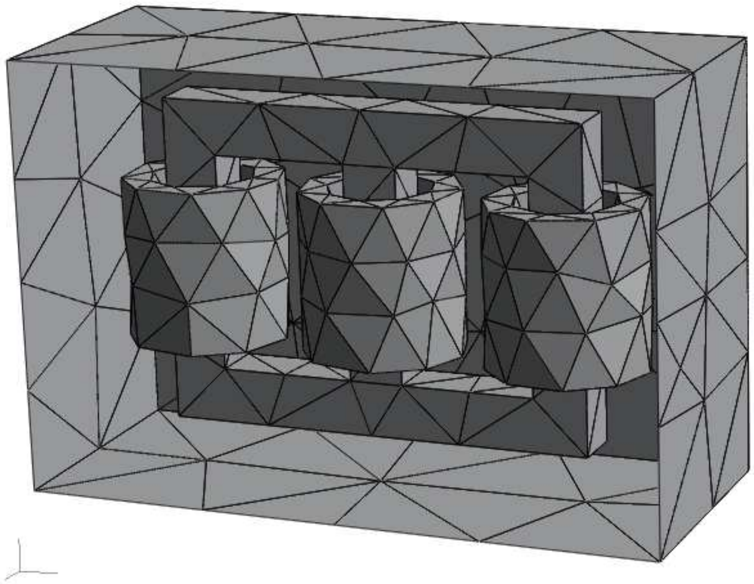

Figure 1.6: Initial mesh of the transformer.

or as grid function being the solution of an other problem. The linear form is similarly defined. In our application the matrix H corresponds to the bilinear form (ah) whereas the modified Schur complement matrix $\tilde{G}$ corresponds to the bilinear form (ae) which is basically realized by some special form denoted by the token `rotrotedgestab`.

Next, we have to define a preconditioner. We want to use the multigrid preconditioner described in Section 1.5. The preconditioner needs the bilinear form to work with. It can be tuned for the problem by a list of parameters. Clearly, the type of smoothing and prolongation operators have to be chosen to meet the properties of the underlying problem. The multigrid cycle, the number of smoothing steps and the coarse grid solver can be adapted to the specific example.

The problem to be solved is a boundary value problem. This needs the bilinear form, the linear form, and the gridfunction for the solution. These objects are internally represented by matrices and vectors. The solver for symmetric problems is a preconditioned conjugate gradient method. We use the multigrid preconditioner (ce) defined in Figure 1.7 (cf. also Section 1.3.3).

Next, the type of post-processing is specified. The function `calcflux` computes the natural fluxes for the problem at hand. For the Maxwell equations it computes the induction field B. Finally, the error estimator and element selection strategy for the refinement of the grid is defined. Since the fluxes are already computed, the Zienkiewicz-Zhu error estimator `estzz` is very cheap to compute (see, e.g., [29]).

```
loadgrid  trafo

// edge element space:
define space vedge -edge -domains=[ 1, 2, 3, 4, 5 ]

// define forms and gridfunctions:
define bilinearform ae -spc=vedge
define bilinearform ah -spc=vedge
define linearform fe -spc=vedge
define gridfunction ue -spc=vedge
define gridfunction b -dim=3 -elconst

// build up forms:
add ah rotrotedge ( ([ 1e-3, 1, 1, 1, 1 ])) -quad=3
add ah massedge (1e-3) -quad=2
add ah robin (1e8) -quad=5 -bound=1
add ae rotrotedgestab ( ([ 1e-3, 1, 1, 1, 1 ]), (0), (1), (1000) ) -quad=3
add ae robin (1e8) -quad=5 -bound=1

add fe sourceedge1 ( (x3), 0, (-x1-7) ) -dom=2 -quad=1
add fe sourceedge1 ((-x3), 0,    (x1) ) -dom=3 -quad=1
add fe sourceedge1 ( (x3), 0, (-x1+7) ) -dom=4 -quad=1

// define multigrid preconditioner:
define preconditioner ce -name=mg -mat=ah -prol=21 -smoother=3 -cycle=1

// solve boundary value problem:
define problem bvp -mat=ae -rhs=fe -sol=ue -pre=ce

// define prostprocessing (fluxes):
define problem calcflux -mat=ae -sol=ue -flux=b

// ZZ - error estimator:
define gridfunction elerr -elconst
define problem estzz -mat=ae -flux=b -sol=ue  -elerr=elerr
define problem badelementmarker -elerr=elerr -fac=0.1 -minlevel=1
```

Figure 1.7: Problem description for the transformer.

1.4.4 The Adaptive Multilevel (Parallel) Solver FEPP

The numerical simulations are carried out using the object oriented C++ code *FEPP* [28]. The required input including the mesh, geometry and problem description has been described already. Starting with a coarse mesh, a hierarchy of refined meshes is produced based on error estimators until a desired accuracy is reached. The hierarchy of those meshes is essential for the application of efficient multigrid and multilevel preconditioners which guarantee optimal complexity of the algorithms.

After starting *FEPP* the problem description file is processed starting with loading the mesh. Then, the objects defined in the file are created. The command

```
>assemble
```

causes the assembling of all objects on the current grid. In our case, this includes the allocation of the solution vector, the allocation of the vectors and matrices corresponding to linear and bilinear forms, followed by the assembling of the integrators given by the `add` operation. The assembling of the problem `bvp` involves immediately the iterative solution of the finite element equations. The post-processing includes the calculation of fluxes according to the components of the bilinear form as well as the error estimator giving the refinement information, i.e., elements with large local error are marked for refinement.

The user can proceed interactively with the command

```
>refine
```

which causes the refinement of the grid and the assembling of all objects on the new grid. After 5 refinement steps, the grid shown in Figure 1.8 is obtained. Alternatively, the command

```
>draw3d
```

allows an interactive visualization of the current data including meshes and data of grid functions. The concept of the visualization is described in the next subsection.

1.4.5 Scientific Visualization

The development of our own visualization tool *VIPP* [22] was driven by the goal of interactive visualization during adaptive multilevel computations and by the requirements caused by very large data sets. In particular, since modern numerical algorithms as described in this paper allow us to execute 3D computations with several 10^5 unknowns within minutes even on notebook computers, the performance of interactive visualization including rotating objects and zooming is an important issue. So, any system based on file transfers is prohibitive for online visualization.

Instead, interactive visualization has to be considered as a part of the simulation process. In our case, a shared data concept was chosen. Namely, *VIPP* accesses the data owned by *FEPP* directly. The interface between *VIPP* and *FEPP* is realized by general *FeppDataAccess*–functions. Moreover, the visualization can make use of the hierarchical data structures supplied by *FEPP*. That means, during CPU intensive visualization tasks, e.g., slicing through objects, smaller data sets corresponding to coarser meshes are used.

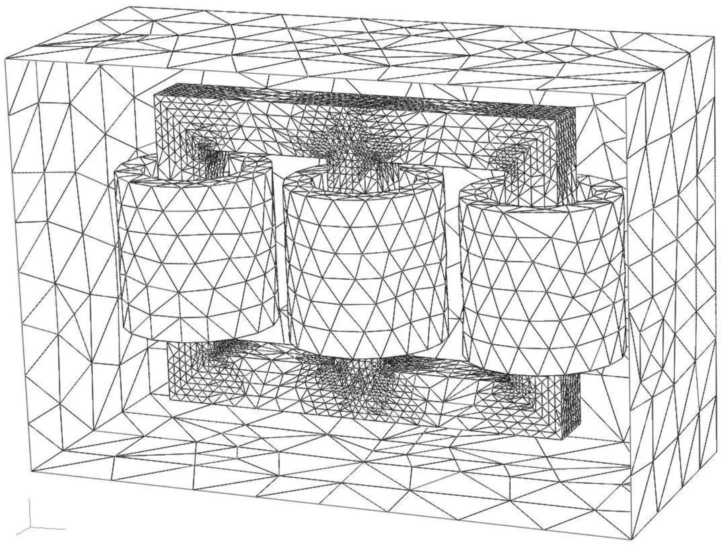

Figure 1.8: Level 6 mesh of the transformer.

The finest resolution (corresponding to the access to data of the finest grid) is restored as soon as the user stops interactive tasks.

The tool *VIPP* [22] allows us to visualize surface and volume meshes, including shrinked meshes. Slicing planes can be defined and are drawn continuously according to a given point and a normal direction. Scalar gridfunctions (solutions) can be visualized by colors, iso-lines or iso-surfaces.

A 3D vector valued solution (B–field) is shown in Figure 1.9 as a vector plot. The modulus of B is represented by the colors of the arrows.

1.5 NUMERICAL RESULTS

In this section, we present numerical results for our model problem. Both, geometry and problem data have already been given. The magnetic field which is to be computed is generated by tangential currents within the three coils. The currents are now normalized to 1 in contrast to the description file given above. The iron core has a permeabilty of 1000.

As proposed we solve the modified Schur complement system (1.18). The matrix $\tilde{C}$ is implicitly defined by (1.22). The action of $\tilde{C}^{-1}$ corresponds to a multigrid V–cycle with 2 Gauss–Seidel pre- and post-smoothing steps. As a solver for system (1.18) we apply the PCG method with the preconditioner C_H as given in (1.21). The action of C_H^{-1} corresponds to a multigrid V–cycle using a block–Gauss–Seidel smoother with 1 pre- and post-smoothing step [1]. In particular those edges adjacent to one vertex belong to a common block.

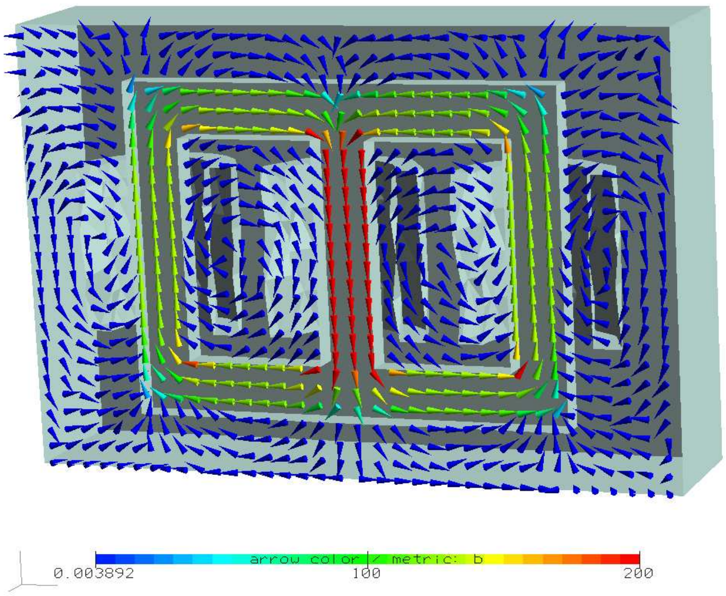

Figure 1.9: Solution (B–field) in a vector plot.

We start the calculations on the coarsest mesh with 2805 unknowns (one unknown per edge). Finer meshes are obtained by adaptive refinement leading to 258867 unknowns after 5 refinement steps. The solution (B–field) is shown in Figure 1.9. As expected, areas of rapid changes in the B-field along edges correspond to areas of mesh refinement (cf. Figure 1.8) according to the error estimator.

Table 1.1 shows the results obtained on a SGI Octane workstation with a RS 10000 processor, 250 MHz and 2GB memory. The CPU times are given in seconds. Hereby, column 4 (acc. time) shows the overall (accumulated) time including mesh refinement, assembling and solution, whereas the last column shows partial times (per level) required for the assembling of linear and bilinear forms (assemble) and the PCG solver (solve). The number of iterations and times are given for a relative accuracy of 10^{-6} required by the PCG. The constant numbers indicate that the preconditioners are robust with respect to

Level	Unknowns	Iter.	acc. time [sec]	level times (assemble/solve)
1	2805	7	5.7	1.4/4.0
2	6211	8	12.8	3.1/2.2
3	10995	9	27.3	5.2/5.9
4	32865	10	73.2	15.4/19.7
5	81614	10	193.4	38.4/54.5
6	258867	11	614.4	135.5/194.7

Table 1.1: Numerical results for the transformer.

the discretization parameter h. Since the CPU time grows from level to level with almost the same factor as the number of unknowns, the algorithm presented in this paper is of asymptotically optimal complexity as was theoretically predicted.

1.6 CONCLUDING REMARKS

In the second part of this paper we have presented scientific computing tools that are necessary for studying numerically 3D field problems of technical complexity by means of advanced adaptive multilevel algorithms. We have just demonstrated the interplay of these tools for the simulation of 3D linear magnetostatic field problems. In the first part, we have developed a robust, asymptotically optimal, multigrid preconditioned CG solver for regularized finite element schemes approximating the 3D magnetostatic Maxwell equations. These numerical techniques can be extended to more advanced electromagnetical field problems including nonlinear problems [13, 14], time-harmonic problems, transient problems and coupled magnetomechanical problems. For instance, in [26], these adaptive multigrid techniques have been successfully extended to 3D transient coupled magnetomechanical problems. The numerical simulation of the acceleration of a magnetic excited aluminum plate was in quite good agreement with the corresponding measured quantities. Moreover, our multigrid solver was 30 times faster than some commercial ICCG solver.

Furthermore, we have proposed a concept for constructing and solving symmetric coupled FE-BE schemes approximating the magnetostatic Maxwell equations in the whole R^3. For the sake of simplicity, we have only considered the case of two subdomains where the BEM cares for a precise approximation of the magnetic field in the exterior air subdomain and the FEM is used in the bounded complement. As in 2D, this technique can be used to construct nonoverlapping domain decomposition methods. In dependence of the data, one can use the BEM in some subdomains whereas the FEM is applied in the other subdomains. We mention that on the interfaces between two BEM subdomains the additional unknown $\lambda := B \cdot n$ occurs (see [20] for details). These domain decomposition schemes are especially suited for parallel computing. Following the parallelization concepts proposed in [24], [21] and [10], we have developed a parallel version of our multigrid package FEPP under MPI. For this purpose, at least, one further scientific computing tool that handles some automatic domain decomposition and data distribution is required. We use a modified recursive spectral bisection technique that was before developed in 2D [11, 18, 20].

REFERENCES

[1] D. Arnold, R. Falk and R. Winther (1999). Multigrid in H(div) and H(curl). *Numer. Math.* (to appear).

[2] A. Bossavit (1991). Mixed methods and the marriage between mixed finite elements and boundarys elements. *Numerical Methods for Partial Differential Equations*, 7:347–362.

[3] J. H. Bramble and J. E. Pasciak (1988). A preconditioning technique for indefinite

systems resulting from mixed approximations of elliptic problems. *Mathematics of Computation*, 50(181):1–17.

[4] F. Brezzi and M. Fortin (1991). *Mixed and Hybrid Finite Element Methods*, volume 15 of *Springer Series in Computational Mathematics*. Springer-Verlag.

[5] C. Carstensen, M. Kuhn and U. Langer (1998). Fast parallel solvers for symmetric boundary element domain decomposition equations. *Numer. Math.*, 79:321–347.

[6] P. Ciarlet (1978). *The Finite Element Method for Elliptic Problems.* North-Holland Publishing Company, Amsterdam-New York-Oxford.

[7] M. Costabel (1987). Symmetric methods for the coupling of finite elements and boundary elements. In C. A. Brebbia, W. L. Wendland and G. Kuhn, editors, *Boundary Elements IX*, pages 411–420. Springer–Verlag.

[8] L. Demkowicz and L. Vardaptyan (1998). Modeling of electromagnetic absorption/scattering using hp-adaptive finite elements. *Comput. Methods Appl. Mech. Engrg.*, 152:103–124.

[9] V. Girault and P.A. Raviart (1986). *Finite Element Methods for Navier-Stokes Equations.* Springer-Verlag.

[10] G. Haase (1999). *Parallelisierung numerischer Algorithmen für partielle Differentialgleichung.* Teubner Verlag, Stuttgart.

[11] G. Haase and M. Kuhn (1999). Preprocessing for 2d FE-BE domain decomposition methods. *Computing and Visualization in Science.* (to appear).

[12] W. Hackbusch (1985). *Multi–Grid Methods and Applications*, volume 4 of *Springer Series in Computational Mathematics.* Springer–Verlag, Berlin.

[13] B. Heise (1994). Analysis of a fully discrete finite element method for a nonlinear magnetic field problem. *SIAM J. Numer. Anal.*, 31(3):745–759.

[14] B. Heise, M. Kuhn and U. Langer (1997). A mixed variational formulation for 3D magnetostatics in the space $H(\mathrm{rot})\cap H(\mathrm{div})$. Technical Report 97-1, Johannes Kepler University Linz, Institute of Mathematics.

[15] R. Hiptmair (1999). Multigrid methods for Maxwell's equations. *SIAM J. Numer. Anal.*, 36:204–225.

[16] G. C. Hsiao and W. Wendland (1991). Domain decomposition in boundary element methods. In *Proc. of IV Int. Symposium on Domain Decomposition Methods, (R. Glowinski, Y. A. Kuznetsov, G. Meurant, J. Périaux, O. B. Widlund eds.), Moscow, May 1990*, pages 41–49. SIAM Publ., Philadelphia.

[17] M. Jung, U. Langer, A. Meyer, W. Queck and M. Schneider (1989). Multigrid preconditioners and their applications. In G. Telschow, editor, *Third Multigrid Seminar, Biesenthal 1988*, pages 11–52. Karl–Weierstrass–Institut, Berlin. Report R–MATH–03/89.

[18] R. Koppler, G. Kurka and J. Volkert (1997). Exdasy - an user-friendly and extentable data distribution system. *Lecture Notes in Computer Science*, 1300:118–127.

[19] A. Kost (1994). *Numerische Methoden in der Berechnung elektromagnetischer Felder.* Springer–Lehrbuch. Springer Verlag, Berlin, Heidelberg.

[20] M. Kuhn (1998). *Efficient Parallel Numerical Simulation of Magnetic Field Problem.* Ph.D. thesis, Johannes Kepler University Linz.

[21] M. Kuhn and U. Langer (1997). Adaptive domain decomposition methods in FEM and BEM. In J.R. Whiteman, editor, *The Mathematics of Finite Elements and Applications*, pages 103–122. John Wiley & Sons.

[22] G. Kurka (1998). A fast visualization system for adaptive grids. SFB Report 98-11, Johannes Kepler University Linz, SFB "Numerical and Symbolic Scientific Computing".

[23] St. Kurz, J. Fetzer, G. Lehner and W.M. Rucker (1998). A novel formulation for 3d eddy current problems with moving bodies using a Lagrangian description and BEM-FEM coupling. *IEEE Transaction on Magnetics*, 34:3068–3073.

[24] U. Langer (1994). Parallel iterative solution of symmetric coupled BE/FE–equations via domain decomposition. *Contemp. Math.*, 157:335–344.

[25] J. Nédélec (1986). A new family of mixed finite elements in R^3. *Numer. Math.*, 50:57–81.

[26] M. Schinnerl, J. Schöberl, M. Kaltenbacher, U. Langer and R. Lerch (1999). Multigrid methods for the fast numerical simulation of coupled magnetomechanical systems. *ZAMM.* (to appear).

[27] J. Schöberl (1997). NETGEN - An advancing front 2D/3D-mesh generator based on abstract rules. *Computing and Visualization in Science*, 1:41–52.

[28] J. Schöberl (1997). Object-oriented Finite Element Code *FEPP.* See also: http://www.sfb013.uni-linz.ac.at.

[29] R. Verfürth (1996). *A Review of A Posteriori Error Estimation and Adaptive Mesh–Refinement Techniques.* Wiley-Teubner Series in Advances in Numerical Mathematics. Wiley and Teubner, Chichester, Stuttgart.

[30] W. Zulehner (1998). Analysis of iterative methods for saddle point problems: A unified approach. Institutsbericht 538, Johannes Kepler University Linz, Department of Numerical Analysis.