Abstract—This paper deals with the efficient solution of Maxwell’s equation in the static and eddy current case using enhanced Algebraic MultiGrid methods. The magnetic vector potential is used as the field variable and Lagrange (nodal) as well as Nédélec (edge) finite elements are applied for the spatial discretization. Numerical studies for the static case (TEAM 20), the eddy current case in time domain (Magnetic Resonance Imaging scanner) and in frequency domain (electric transformer) will demonstrate the applicability of the developed Algebraic MultiGrid methods.

I. INTRODUCTION

Numerical computations of electromagnetic fields are performed for more than 20 years. Many different formulations using Lagrange (nodal) as well as Nédélec (edge) finite elements for the spatial discretization have been developed (see e.g. [4], [3]). Even many commercial codes are available for the computation of electromagnetic fields in 3D. However, the efficient solution with respect to CPU-time and memory amount for the resulting algebraic system of equations is still a challenging task, and requires ongoing research.

In this paper we will concentrate on the static and eddy current case (both in time and frequency domain) by applying enhanced Algebraic MultiGrid (AMG) solvers developed in the last 4 years. The paper is organized as follows: In Sec. II we will describe the correct formulation and Finite Element (FE) discretization of Maxwell’s equations for the eddy current case. Next, in Sec. III we will review the convergence behavior of iterative solvers, the properties of MultiGrid (MG) methods, and the motivation for AMG solvers. The main part of this paper concentrates on a detailed discussion of AMG methods and special adaption for algebraic systems of equations arising from nodal as well as edge FE-discretizations (see Sec. IV and V). Finally, we will report on the performance of the developed AMG methods for three case studies: TEAM 20, Magnetic Resonance Imaging scanner and electric transformer.

II. GOVERNING EQUATIONS AND FINITE ELEMENT DISCRETIZATIONS

The electromagnetic field is fully described by Maxwell’s equations [33]. Restricting the problem class to the quasi-static (eddy current) case, we arrive at the following partial differential equation for the magnetic vector potential $\mathbf{A}$

$$\gamma \frac{\partial \mathbf{A}}{\partial t} + \nabla \times \nu \nabla \times \mathbf{A} = \mathbf{J}_i$$

with boundary condition $\mathbf{n} \times \mathbf{A} = 0$ and $\mathbf{n}$ the outward normal vector. In (1) $\mathbf{J}_i$ denotes the impressed current density, $\nu$ the magnetic reluctivity and $\gamma$ the electric conductivity. Furtheron, the following interface conditions have to be fulfilled

$$[\mathbf{A} \times \mathbf{n}] = 0; \quad [\nu \mathbf{n} \times \nabla \times \mathbf{A}] = 0; \quad [\gamma \frac{\partial \mathbf{A}}{\partial t}] = 0$$

with $[\mathbf{Z}] = \mathbf{Z}_{\text{right}} - \mathbf{Z}_{\text{left}}$. For further discussions let $\Omega$ be a bounded single connected convex domain with boundary $\partial \Omega = \Gamma$. The function spaces $L^2(\Omega)$ and $H^1_0(\Omega)$ are defined as usual, see e.g. [1]. Therewith, the variational formulation for (1) in the function space

$$\mathbf{H}_0(\text{curl}) = \{ \mathbf{u} \in (L^2(\Omega))^3 \mid \nabla \times \mathbf{u} \in (L^2(\Omega))^3, \mathbf{u} \cdot \mathbf{n}|_{\Gamma} = 0 \}$$

reads as follows: Find $\mathbf{A} \in \mathbf{H}_0(\text{curl})$ such that

$$\int_{\Omega} \gamma A'_i \frac{\partial \mathbf{A}}{\partial t} d\Omega + \int_{\Omega} \nabla \times \mathbf{A}' \cdot \nu \nabla \times \mathbf{A} d\Omega = \int_{\Omega} \mathbf{A}' \cdot \mathbf{J}_i d\Omega$$

for any $\mathbf{A}' \in \mathbf{H}_0(\text{curl})$ is fulfilled.

It is well known, that an edge FE-discretization of (4) is $\mathbf{H}_0(\text{curl})$-conform [23]. Nevertheless, the solution of the algebraic system requires special care in order to obtain an optimal MultiGrid solver (see e.g. [2], [13]). We suggest to add a fictive electric conductivity $\gamma'$ to regions with zero electric conductivity to obtain a variational form, which is elliptic [32]. Of course, this fictive conductivity $\gamma'$ has to be chosen small as compared to the reluctivity of the material. The proof of convergence even in the case of $\gamma' \rightarrow 0$ is given in [27].

For the application of nodal finite elements, we have to perform additional steps. As shown in [9], the space $\mathbf{H}_0(\text{curl})$ can be decomposed for any convex domain $\Omega$ into

$$\mathbf{H}_0(\text{curl}) = \{ H^1_0(\Omega)^3 \oplus \nabla H^1_0(\Omega) \},$$

which corresponds to splitting the magnetic vector potential $\mathbf{A}$ as follows

$$\mathbf{A} = \mathbf{u} + \nabla \phi, \quad \nabla \cdot \mathbf{u} = 0,$$

with $\mathbf{u} \in (H^1_0(\Omega))^3$ and $\phi \in H^1_0(\Omega)$. The same decomposition is done for the test function $\mathbf{A}' = \mathbf{v} + \nabla \psi$. Since we have to guarantee $\nabla \cdot \mathbf{u} = 0$, we do so by adding the penalty term $\int_{\Omega} s \nu (\nabla \cdot \nabla \cdot \mathbf{u}) d\Omega$ to the variational formulation (4) (weighted regularization method with special function $s$, see [8], [17]). Therewith, the variational formulation can be stated as follows: Find $(\mathbf{u}, \phi) \in (H^1_0(\Omega))^3 \times H^1_0(\Omega)$ such that

$$\int_{\Omega} \nu \nabla \times \mathbf{u} \cdot \mathbf{v} + \int_{\Omega} s \nu \nabla \cdot \mathbf{u} \mathbf{v} d\Omega + \int_{\Omega} \gamma (\mathbf{v} + \nabla \psi) \frac{\partial}{\partial t} (\mathbf{u} + \nabla \phi) d\Omega = \int_{\Omega} \mathbf{J}_i \cdot \mathbf{v} d\Omega,$$

for any $(\mathbf{v}, \psi) \in (H^1_0(\Omega))^3 \times H^1_0(\Omega)$.

In both cases, nodal as well as edge finite elements, the spatial and subsequent time discretization lead to an algebraic system of equations

$$K^\nu_h A^\nu_h + \frac{\partial}{\partial t} K^\nu_h A^\nu_h = f^\nu_h, \quad K^\psi_h A^\psi_h = f^\psi_h.$$
For the rest of the paper the superscripts \(e,n\) are omitted, whenever this is possible without causing any confusion.

III. ITERATIVE AND MULTIGRID METHODS

Let us consider the algebraic system of equations of the form

\[ K_h u_h = f_h . \]

(9)

Therein \( K_h \in \mathbb{R}^{N_h \times N_h} \) denotes the system matrix, \( f_h \in \mathbb{R}^{N_h} \) the right hand side and \( u_h \in \mathbb{R}^{N_h} \) the solution vector of the unknown nodal (edge) quantity (mainly the magnetic vector potential). Additionally the entries of \( K_h \) are given by \( k_{ij} = (K_h)_{ij} \in \mathbb{R}^{p \times p} \) with \( p \) defining the number of unknowns per node (edge). The number of unknowns \( N_h \) is related to the usual discretization parameter \( h \) by the relation \( N_h = O(h^{-d}) \), with \( d = 2,3 \) the spatial dimension. The system matrix \( K_h \) is supposed to be sparse and symmetric positive definite (SPD). In general \( N_h \) is quite large and due to limited memory resources, iterative solvers have to be used instead of direct ones. However, the convergence of iterative solvers strongly depends on the condition number \( \kappa \) of the system matrix \( K_h \)

\[ \kappa(K_h) = \frac{\lambda_{\text{max}}(K_h)}{\lambda_{\text{min}}(K_h)} \]

(10)

with \( \lambda_{\text{max}} \) and \( \lambda_{\text{min}} \) the largest and the smallest eigenvalue of \( K_h \), respectively. In general, the convergence rate decreases when \( \kappa \) gets large. Since \( K_h \) stems from an FE-discretization of a second order partial differential equation (PDE), the condition number \( \kappa(K_h) \) typically behaves like \( O(h^{-2}) \). In order to cope with large condition numbers, we apply a symmetric preconditioner \( C_h \) to (9), i.e.

\[ C_h^{-1} K_h = C_h^{-1} f_h , \]

(11)

with the properties

- \( C_h^{-1} \) is an approximate inverse of \( K_h \), and
- \( C_h^{-1} \) can be applied very fast.

Consequently the condition number of the preconditioned system is much smaller than the original one. For instance, the standard preconditioner is the well known Incomplete Cholesky (IC) method (see e.g.[5]). Furthermore, the preconditioned system (11) is solved via a Krylov subspace method, i.e. Conjugate Gradient (CG) or Quasi Minimal Residual (QMR) method, see [30]. The standard preconditioners (IC, Jacobi, etc.) suffer from the fact, that the condition number still depends on \( h \). In order to get condition numbers independent of the discretization parameter \( h \), MultiGrid methods (MG) have to be used, for which it can be shown that the number of necessary iterations does not depend on the mesh parameter \( h \) (see e.g. [10]). As shown in [14], the most robust solution strategy for SPD linear equations (9) is PCG with geometric MultiGrid preconditioner. Recently, investigations have been carried out to adapt geometric MultiGrid (MG) methods for the fast solution of 3D electromagnetic field problems (e.g. [2], [13], [31], [35]).

However, geometric MG methods suffer from the inherent need of a hierarchical FE-mesh (see [10]), and thus algebraic MultiGrid (AMG) methods are of special interest, if at least one of the following cases arises:

1. The discretization provides no hierarchy of FE-meshes, which would be essential for the geometric MG method. This is the case for many FE-codes, especially commercial ones.

2. The coarsest grid of a geometric MultiGrid method is too large to be solved efficiently by a direct or classical iterative solver.

3. Classical iterative solvers are not efficient enough.

AMG methods try to mimic the geometric counterpart, but only rely on the information available on a given single grid (for the pioneer work on AMG see [29]). While within a geometric MG solver the construction of a matrix hierarchy is rather simple if a hierarchy of grids is available (see e.g. [10]), this task is not as easy if either the matrix only or the information on the finest grid are available. The classical AMG approach assumes an SPD system matrix which is additionally an M-matrix [29]. For such matrix classes a matrix hierarchy can be constructed, imitating the geometric counterpart well. It can be easily shown, that the information of an SPD system matrix is not enough in order to construct an efficient and robust AMG method. Therefore, we assume the knowledge of the underlying PDE, the FE-discretization scheme and additional information on the given FE-mesh. Therewith, such enhanced AMG methods are able to reproduce the behavior of geometric MG methods even for Maxwell’s equation, although the system matrices are not M-matrices here.

IV. ALGEBRAIC MULTIGRID METHODS

In order to outline the principles of the MultiGrid method we explain them by means of a two grid method (for an overview see [7]). For this purpose \( h \) and \( H \) describe the fine and coarse grids of an FE-discretization, respectively. The linear mappings (with \( N_h > N_H \) )

\[ R_h : \mathbb{R}^{N_h} \mapsto \mathbb{R}^{N_H} \quad \text{and} \quad P_h : \mathbb{R}^{N_H} \mapsto \mathbb{R}^{N_h} \]

(12)

are called restriction and prolongation operators. Usually we choose \( R_h = P_H^T \) for AMG. Therewith, the two grid algorithm is performed as presented in Alg. 1.

**Algorithm 1 TwoGridMethod**

\[ \begin{array}{l}
\text{Relax } \nu_1 \text{ times on the fine grid } K_h u_h = f_h \\
\text{(e.g. Gauß-Seidel forward)} \\
\text{Compute} \ \text{the defect} \ d_h = f_h - K_h u_h \\
\text{Restrict} \ \text{the defect} \ d_h \ \text{onto the coarse grid} \\
\quad d_H = R_h d_h \\
\text{Solve} \ \text{the coarse grid problem} \ K_H u_H = d_H \\
\text{Prolongate} \ \text{the coarse grid correction} \ u_H \ \text{onto the fine grid} \\
\quad u_h = P_h u_H \\
\text{Update} \ u_h \ \text{by} \ u_h, \ i.e. \ u_h = u_h + u_h \\
\text{Relax} \ \nu_2 \text{ times on the fine grid } K_h u_h = f_h \\
\text{(e.g. Gauß-Seidel backward)} \\
\end{array} \]

By replacing the exact solution of the coarse grid problem in Alg. 1 itself by a two grid approximation, we arrive
at the recursive definition of a MultiGrid cycle. The motivation for this approach comes from examining the error of the numerical solution in the frequency domain. High frequency errors, which include local variations in the solution, are well eliminated by simple iterative smoothing methods (e.g. Gauss-Seidel smoother). Once this is achieved, further fine-grid iterations would not improve significantly the convergence rate. Therefore, the solution is transferred to a coarser grid by using an appropriate restriction operator \( R_h \). On this grid, the low frequency errors of the fine grid manifest themselves as relatively high frequency errors, and are thus eliminated efficiently using again simple iterative smoothing methods. If the coarsest grid has been reached, the equation has to be solved exactly (e.g. direct solver), which can be done with little computational effort due to the small numbers of unknowns. Consequently, each grid level is responsible for eliminating a particular frequency bandwidth of the error.

The subsequent discussion on AMG follows mainly [25]. First of all, we have to perform the coarsening process to extract from the given system matrix (arising form the FE-discretization) matrices with decreasing dimension. The key point of the coarsening process is to construct an auxiliary matrix on which the coarsening is performed. Thereafter, we can always guarantee an appropriate coarsening and, in addition to be very fast. Furtheron, we have to define the smoothing operator and the restriction (prolongation) operator for the transfer of data between the different hierarchies based on the auxiliary matrix.

### A. Auxiliary Matrix

Let us assume, that the system matrix \( K_h \) stems from an FE-discretization on the FE-mesh \( \omega_h = (\omega_h^n, \omega_h^e) \), with \( \omega_h^n, |\omega_h^n| = M_h \) being the set of nodes and \( \omega_h^e \) being the set of edges (see Fig. 1). An edge is defined as a pair of indices for which the connection of the two points is a geometric edge. For instance, let \( i, j \in \omega_h^n \) be the indices of the nodes \( x_i, x_j \in \mathbb{R}^d \) then the edge is given by

\[
e_{ij} = (i, j) \in \omega_h^e
\]

and the corresponding geometric edge vector can be expressed by

\[
a_{ij} = x_i - x_j \in \mathbb{R}^d.
\]

The first task we are concerned with is the construction of an auxiliary matrix \( B_h \in \mathbb{R}^{M_h \times M_h} \) with the following properties:

\[
(B_h)_{ij} = \begin{cases} 
  b_{ij} \leq 0 & \text{if } i \neq j, \\
  1 - \sum_{j \neq i} b_{ij} \geq 0 & \text{if } i = j.
\end{cases}
\]

The entries of \( B_h \) should be defined in such a way that the distance and parameter jumps of the variational forms are reflected. The matrix pattern of \( B_h \) can be constructed via different objectives: \( B_h \) reflects the geometric FE-mesh, which is of importance for an edge element discretization, or \( B_h \) reflects the matrix pattern of the system matrix \( K_h \), which is useful for nodal FE-discretizations.

### B. Coarsening Process

The auxiliary matrix \( B_h \) is a sparse M-matrix and therefore the coarsening process for \( B_h \) is straightforward and can be done in a robust way. We know that \( B_h \) represents a virtual FE-mesh \( \omega_h = (\omega_h^n, \omega_h^e) \). Such a virtual FE-mesh can be split into two disjoint sets of nodes, i.e.,

\[
\omega_h^n = \omega_C^n \cup \omega_F^n, \quad \omega_C^n \cap \omega_F^n = \emptyset
\]

with sets of coarse grid nodes \( \omega_C^n \) and fine grid nodes \( \omega_F^n \). The splitting is usually performed such that no coarse grid nodes are connected directly and that the number of coarse grid nodes is as large as possible (see Fig. 2).

![Fig. 1. Clipping of an FE-mesh in 2D.](image)

In order to perform a coarsening algorithm, let us introduce the following sets:

\[
N_h^i = \{ j \in \omega_h^n : |b_{ij}| \neq 0, i \neq j \},
\]

\[
S_h^i = \{ j \in N_h^i : |b_{ij}| > \text{coarse} (B_h, i, j), i \neq j \},
\]

\[
S_h^{i,T} = \{ j \in N_h^i : i \in S_h^j \},
\]

where \( N_h^i \) is the set of neighbors, \( S_h^i \) denotes the set of ‘strong connections’ and \( S_h^{i,T} \) is related to the set of nodes, which have a strong connection to node \( i \), respectively. The cut-off (coarsening) function is chosen as, e.g.

\[
\text{coarse} (B_h, i, j) = \begin{cases} 
  \theta \cdot \sqrt{|b_{ii}| |b_{ij}|}, & \text{see } [29], \\
  \theta \cdot \max_{i \neq i} |b_{ii}|, & \text{see } [20], \\
  \theta, & \text{see } [34].
\end{cases}
\]

(15)

with an appropriate \( \theta \in [0, 1] \). In addition, we define the local sets

\[
\omega_C^i = \omega_C^n \cap N_h^i, \quad \omega_F^i = \omega_F^n \cap N_h^i
\]

(16)
and
\[ E_h^i = \{(i, j) \in \omega_h^i : j \in N_h^i \}. \] (17)
The coarsening algorithm is described in Alg. 2.

**Algorithm 2 Coarsening phase**

\[
\omega_C^h \leftarrow \emptyset, \quad \omega_F^h \leftarrow \emptyset
\]

**while** \( \omega_C^h \cup \omega_F^h \neq \omega_h^i \) **do**

\[
i \leftarrow \text{Pick}(\omega_h^i \setminus (\omega_C^h \cup \omega_F^h))
\]

**if** \(|S_h^{i,T}| + |S_h^{i,T} \cap \omega_F^h| = 0\) **then**

\[
\omega_F^h \leftarrow \omega_h^i \setminus \omega_C^h
\]

**else**

\[
\omega_C^h \leftarrow \omega_C^h \cup \{i\}
\]

\[
\omega_F^h \leftarrow \omega_F^h \cup (S_h^{i,T} \setminus \omega_C^h)
\]

**end if**

**end while**

Therein the function
\[
i \leftarrow \text{Pick}(\omega_h^i \setminus (\omega_C^h \cup \omega_F^h))
\]
returns a node \(i\) where the number \(|S_h^{i,T}| + |S_h^{i,T} \cap \omega_F^h|\) is maximal.

**Example:** Let us consider the FE-mesh of Fig. 3. The auxiliary matrix is defined on an FE-element \(r\) by the setting
\[
b_{ij}^r = \frac{\nu_r}{\|a_{ij}\|_2} \quad i \neq j
\]
with \(\nu_r\) the material parameter and \(a_{ij}\) the geometric edge vector (see (13)). Let us assume the following entries for row 5 of the assembled auxiliary matrix
\[
b_{51} = -202 \quad b_{54} = -400 \quad b_{57} = -100
\]
\[
b_{52} = -2 \quad b_{55} = 1009 \quad b_{58} = -101
\]
\[
b_{53} = -200 \quad b_{56} = -2 \quad b_{59} = -1
\]

Using the coarsening function of [29] (see (15)) with \(\theta = 0.25\), we obtain
\[
N_h^5 = \{1, \ldots, 4, 6, \ldots, 9\}
\]
\[
S_h^5 = \{1, 3, 4, 7, 8\}
\]
\[
S_h^{5,T} = \{4, 6, 7\}.
\]

For the construction of set \(S_h^{5,T}\) we assumed
\[
S_h^1 = \{3\} \quad S_h^8 = \{1, 2, 5, 8\}
\]
\[
S_h^2 = \{1\} \quad S_h^9 = \{4, 5, 8\}
\]
\[
S_h^3 = \{1\} \quad S_h^6 = \{7, 9\}
\]
\[
S_h^4 = \{1, 3, 5, 7, 8\} \quad S_h^7 = \{7, 8\}.
\]

A special coarsening algorithm is the agglomeration technique, where \(\theta\) is set to 0. Consequently \(N_h^i = S_h = S_h^{i,T}\) for all \(i = 1, \ldots, M_h\). Further we call \((I_h^i)_{i=1}^{M_h}\) \(|\omega_h^i| = M_H < M_h\) a ‘disjoint’ splitting for the agglomeration method if
\[
I_h^i \cap I_h^j = \emptyset, \quad \bigcup_{i=1}^{M_H} I_h^i = \omega_h^i,
\]
is valid, see Fig. 4.

**Fig. 4.** ‘Virtual’ FE-mesh with a feasible agglomeration.

If an appropriate prolongation \(Q_h\) for \(B_h\) is defined then a coarse auxiliary matrix is computed by
\[
B_H = (Q_h)^T B_h Q_h
\]
and \(B_H\) represents again a virtual FE-mesh \(\omega_H = (\omega_h^0, \omega_F^0)\), with \(\omega_h^0 = \omega_F^0\). It can be shown, that \(B_H\) is again an M-matrix if the prolongation operator \(Q_h\) fulfills certain criteria [29]. Thus the coarsening process can be applied recursively. Finally it is assumed that the degrees of freedom on the coarse grid are numbered first. For instance the nodes are reordered like
\[
\omega_h^0 = (\omega_C^0, \omega_F^0)
\]
(similarly for edges) and as a consequence the system matrix can be written as
\[
K_h = \begin{pmatrix}
K_{CC} & K_{CF} \\
K_{TC}^T & K_{FF}
\end{pmatrix}.
\]

**C. Prolongation Operators**

For a given splitting \(\omega_h^0 = \omega_C^0 \cup \omega_F^0\) the optimal prolongation operator is given by the Schur complement approach, i.e.,
\[
K_H = K_{CC} - K_{FC} K_{FF}^{-1} K_{CF}
\]
with
\[
\bar{P}_h = (I_H, -K_{CF} K_{FF}^{-1})^T.
\]
The prolongation operator \(\bar{P}_h\) can hardly be realized in practice since \(-K_{CF} K_{FF}^{-1}\) involves the inverse of \(K_{FF}\), which in turn implies a global transport of information. In addition, the coarse grid operator \(K_H\) becomes dense. The goal of an AMG method is to approximate \(\bar{P}_h\) by some prolongation operator \(P_h\) which acts only locally and therewith produces a sparse coarse grid matrix.
D. Smoother and Coarse Grid Operator

An essential point in MG methods is the smoothing operator \(S_h = \mathbb{R}^{N_h \times N_h}\) which reduces the high frequency error components. Typically, a particular smoother works for certain classes of matrices. It is shown in [6] that a point Gauss-Seidel or point Jacobi smoother is appropriate for FE-discretizations with Lagrange FE-functions for scalar elliptic PDEs of second order. Analogously, the block Gauss-Seidel and block Jacobi smoother work well for the block counterpart, e.g., discretization of Maxwell’s equation with nodal finite elements. For the edge FE-discretization we use a patch smoother.

The coarse grid operator \(K_H\) is usually constructed by Galerkin’s method, i.e.,

\[
K_H = P_H^T K_h P_H. \tag{18}
\]

After a successful setup an AMG-cycle can be performed as usual (e.g. see [10]). For instance in Alg. 3 a \(V(\nu_F, \nu_H)\)-cycle with variable pre- and post-smoothing steps is described. The variable \textsc{CoarseLevel} stores the number of levels generated by the coarsening process until the size of the system is smaller than \textsc{CoarseGrid}.

\begin{algorithm}
\textbf{Algorithm 3} \(V(\nu_F, \nu_H)\)-cycle \hspace{1cm} \textsc{AMGStep}(K, u, f, \ell) \\
\begin{align*}
K_\ell &\leftarrow K, \quad f_\ell \leftarrow f, \quad u_\ell \leftarrow u \\
\text{if } \ell &= \text{CoarseLevel then} \\
&\qquad \text{\textsc{CoarseGridSolver}(} L_\ell, L_\ell^T, f_\ell) \\
&\quad \text{RETURN} \\
\text{else} &\\
&\quad d_\ell \leftarrow 0, \quad w_{\ell+1} \leftarrow 0 \\
&\quad u_\ell \leftarrow S^{\nu_F}_{\ell+1}(u_{\ell+1}, f_\ell) \\
&\quad d_\ell \leftarrow f_\ell - K_{\ell} u_\ell \\
&\quad d_{\ell+1} \leftarrow (P_\ell)^T d_\ell \\
&\quad \textsc{AMGStep}(K_{\ell+1}, u_{\ell+1}, f_{\ell+1}, \ell + 1) \\
&\quad u_{\ell} \leftarrow P_{\ell} u_{\ell+1} \\
&\quad u_\ell \leftarrow u_\ell + u_{\ell} \\
&\quad w_\ell \leftarrow S^{\nu_F}\ell(u_\ell, f_\ell) \\
\end{align*}
\end{algorithm}

V. REALIZATIONS OF AMG

In the following subsections we specialize the abstract algorithms, define the components for nodal and edge FE-discretization and additionally propose a method for complex symmetric systems. Prior to that we mention that static, transient and nonlinear analysis of a given problem results in the solution of linear systems (9). Therefore we restrict ourself to the linear analysis. Other applications can be found in [15], [16], [18], [19], [26].

A. AMG for nodal elements

First we consider equation (7) and use nodal elements for discretization. Note that for the following approach the scalar case \((p = 1)\) is included (e.g. scalar potential equation).

Construction of ‘virtual’ FE-meshes: The definition of the auxiliary matrix \(B_h\) plays an important role for this problem class. The classical approach uses

\[ (B_h)_{ij} = -||k_{ij}||_\infty \quad \text{for} \quad i \neq j. \]

with \(|||\cdot||_\infty\) the maximum norm. The diagonal entries of \(B_h\) are computed according to (14). Now the degrees of freedom per node of the system matrix have to be related to an entry in the auxiliary matrix, which in turn implies that the matrix pattern of \(K_h\) and of \(B_h\) has to be equal, i.e.,

\[ ||k_{ij}||_\infty \neq 0 \Leftrightarrow |b_{ij}| \neq 0. \]

Construction of coarse FE-spaces: The simplest prolongation operator is given by

\[ (P_h)_{ij} = \begin{cases} I_p & \text{if } i = j \in \omega_h^n, \\ \frac{1}{|S_h^{ij} \cap \omega_h^n|} I_p & \text{if } i \in \omega_h^n, j \in S_h^{ij} \cap \omega_h^n, \\ 0 & \text{else,} \end{cases} \]

with \(I_p \in \mathbb{R}^{p \times p}\) the p-dimensional identity matrix. The AMG method shows a better convergence behavior as compared to (19) with the subsequent discrete harmonic extension, i.e.,

\[ (P_h)_{ij} = \begin{cases} I_p & \text{if } i = j \in \omega_h^n, \\ -k_{ij}^{-1}(k_{ij} + c_{ij}) & \text{if } i \in \omega_h^n, j \in \omega_h^n, \\ 0 & \text{else,} \end{cases} \]

with

\[ c_{ij} = \sum_{p \in \omega_h^n} \left( \sum_{q \in \omega_h^n} k_{pq} \right)^{-1} k_{ip} k_{pj}. \]

However, the increasing memory requirement and the slower application compared to the prolongation (19) is the major drawback of the discrete harmonic extension. Note, that the entries of the prolongation operators are matrix valued, e.g. \((P_h)_{ij} \in \mathbb{R}^{p \times p}\), like the entries of the system matrix \(K_h\).

Smoothing operator: We use a block Gauss-Seidel method as smoothing operator, e.g. [6], or a patch-block Gauss-Seidel method, e.g. [21]. The latter should be used for anisotropic problems.

B. AMG for edge elements

The second class originates from an FE-discretization with edge FE-functions of the variational form (4).

Construction of ‘virtual’ FE-meshes: It was motivated by R. Hiptmair in [13] for geometric MG methods that for this class of problems a refinement of the FE-mesh can be performed on the nodes of an FE-mesh as it is usually done for Lagrange FE-functions. We use this fact and base our coarsening on an auxiliary matrix \(B_h\) which is constructed for instance by the FE-element wise setting

\[ b_{ij}^e = -\frac{\nu_r}{||a_{ij}||} \quad \text{for } i \neq j \text{ and } (i,j) \in \omega_h^n. \]

with \(\nu_r\) the reluctivity of the material. Again the diagonal elements are computed via (14).

Example: Let us consider the FE-mesh of Fig. 3 and choose element \(r = 1\). We get the following element matrix

\[ B_h^1 = 100 \cdot \begin{pmatrix} 2.5 & -0.5 & -1 & 0 \\ -0.5 & 2.5 & 0 & -1 \\ -1 & 0 & 2.5 & -0.5 \\ 0 & -1 & -0.5 & 2.5 \end{pmatrix}. \]
The entries \((B_h^1)_{14}, (B_h^1)_{23}, (B_h^1)_{41}\) and \((B_h^1)_{32}\) are zero, i.e., there is no diagonal edge in the virtual FE-mesh.

Let us recall that an FE-mesh is represented by

\[
\omega_h = (\omega_h^n, \omega_h^s),
\]

i.e., the set of nodes \(\omega_h^n\) and the set of edges \(\omega_h^s\). The coarse grid is defined by identifying each coarse grid node \(j \in \omega_h^s\) with an index \(k \in \omega_H^s\). This is expressed by the index map \(\text{ind}()\) as

\[
\omega_h^n = \text{ind}(\omega_C^n).
\]

A ‘useful’ set of coarse grid edges \(\omega_H^s\) can be constructed if we invest in a special prolongation operator \(Q_h\) for the auxiliary matrix \(B_h\). The prolongation operator \(Q_h\) is constructed such that each fine grid node prolongates exactly from one coarse grid node, so that one arrives at a partition of \(\omega_h^n\) into clusters, each of them being represented by a coarse grid variable. We extend the index map \(\text{ind} : \omega_h^n \rightarrow \omega_H^s\) defined above onto the whole fine set \(\omega_h^n\) by assigning to all fine grid nodes of a cluster the coarse grid index of the representative:

\[
\text{ind} : \omega_h^n \rightarrow \omega_H^s.
\]

A consequence is that \(\text{ind}(i) = \text{ind}(j)\) iff \(i, j \in \omega_h^n\) prolongate from the same coarse grid variable. We define an agglomerator (cluster) \(I_h^i\) of a grid point \(i \in \omega_h^n\) by (see Fig. 5)

\[
I_h^i = \{j \in \omega_h^n | \text{ind}(j) = \text{ind}(i)\} \subset N_h^i,
\]

and hence the set of coarse grid nodes can be written as

\[
\omega_H^n = \{\text{ind}(i) | i \in \omega_h^n\}.
\]

The prolongation operator \(Q_h\) has only 0 and 1 entries

\[
(Q_h)_{ij} = \begin{cases} 1 & i \in \omega_h^n, j = \text{ind}(i) \\ 0 & \text{otherwise} \end{cases}.
\]

Now, a coarse grid edge only exists if there is at least one fine edge connecting the agglomerates \(I_h^i\) and \(I_h^j\) with \(i \neq k\) (see Fig. 5), i.e.,

\[
\exists r \in I_h^i, \exists s \in I_h^j \text{ such that } (r, s) \in \omega_h^s.
\]

Note that a decrease of the number of edges in the coarsening process is not proofed in general, but a decrease is heuristically given, if the average number of nonzero entries of \(B_h\) does not grow too fast.

**Construction of coarse FE-spaces:** The construction of the prolongation operator \(P_h : \mathbb{R}^{N_h} \rightarrow \mathbb{R}^{N_h}\), is delicate because of the kernel of the curl-operator, i.e., all gradient fields. \(P_h\) is defined for \(i = (i_1, i_2) \in \omega_h^s, j = (j_1, j_2) \in \omega_H^s\) as

\[
(P_h)_{ij} = \begin{cases} 1 & j = \text{ind}(i_1), \text{ind}(i_2) \\ -1 & j = \text{ind}(i_2), \text{ind}(i_1) \\ 0 & \text{otherwise} \end{cases},
\]

by assuming a positive orientation of an edge \(j = (j_1, j_2)\) from \(j_1\) to \(j_2\) if \(j_1 < j_2\) holds. The constructed prolongation operator \(P_h\) has full rank, because the coarse grid edges prolongate to \(N_h\) distinct fine grid edges by construction. For a detailed discussion see [27].

**Smoothing operator:** To complete the components for an AMG method for edge element FE-discretizations, we need an appropriate smoother. We consider two different types of smoothers for \(K_h\). The first one was suggested by D. Arnold, R. Falk and R. Winther in [2]. This is a block Gauss-Seidel smoother where all edges that belong to \(E_h^i\) (see (17)) are smoothed simultaneously for all \(i \in \omega_h^n\) (see Fig. 6).

Another kind of smoother was suggested by R. Hiptmair in [13]. A mathematically equivalent formulation is outlined in Alg. 4. Therein the vector \(g_h^{r,i} \in \mathbb{R}^{N_h}\) is defined by

![Fig. 5. Virtual FE-mesh with a feasible agglomeration and coarse grid edges](image)

**Algorithm 4 Hybrid smoother of Hiptmair**

\[
u_h \leftarrow \text{GAUSSSEIDEL}(K_h, I_h, u_h)
\]

for all \(i \in \omega_h^n\) do

\[
u_h \leftarrow \nu_h + \frac{(d_h - K_hu_h)g_h^{r,i}}{(K_hg_h^{r,i} \cdot g_h^{r,i})} \cdot g_h^{r,i}
\]

end for

by construction, i.e.,

\[
\omega_h^s \rightarrow \omega_H^s.
\]
with a vector \( g_{h}^{\alpha,i} \in \mathbb{R}^{M_{h}} \), \((g_{h}^{\alpha,i})_{j} = \delta_{ij}\).

It should be noted, that the proposed AMG solver has been adopted by Mifune et. al. and successfully implemented [22].

C. AMG for Time Harmonic Case

In the harmonic case the time derivative of the magnetic vector potential is substituted by

\[
\frac{\partial A}{\partial t} \rightarrow j \omega A
\]

with \( j \) the complex number, \( \omega \) the angular frequency and \( A \) the complex magnetic vector potential. Therewith, we have to apply the AMG method to a complex valued and symmetric algebraic system of equations with system matrix

\[
K_{h} = K_{h}^{re} + j K_{h}^{im}.
\]  

(23)

In (23) \( K_{h}^{re} \) denotes the real part and \( K_{h}^{im} \) the complex part of the system matrix. The application to scalar potential equations has been presented in [28], and adaption to the magnetic vector potential formulation is straight forward as shown below.

Formation of ‘virtual’ FE-meshes: The auxiliary matrix is defined to be real valued. This means, that we setup \( B_{h} \) for an edge element discretization as defined in Sec. V-B. For a nodal element discretization we can use the procedure described in Sec. V-A for \( K_{h}^{re} \).

Construction of coarse FE-spaces: For the construction of a coarse grid operator \( K_{h} \) we define the system prolongation to be real valued and computed as defined in Sec. V-A as well as Sec. V-B. Therefore we get

\[
K_{h} = P_{h}^{T} K_{h} P_{h} = P_{h}^{T} K_{h}^{re} P_{h} + j P_{h}^{T} K_{h}^{im} P_{h} = K_{h}^{re} + j K_{h}^{im}.
\]

The prolongation \( Q_{h} \) is also taken from the real valued realizations correspondingly.

Smoothing operator: In the case of an algebraic system of equations arising from a nodal FE-discretization we apply a block Jacobi or Gauss-Seidel smoother in the complex variant. The complex version of the smoother proposed by D. Arnold, R. Falk and R. Winther in [2] is used for an edge FE-discretization.

VI. CASE STUDIES

In order to gain robustness and efficiency, the proposed AMG methods were used as a preconditioner in the conjugate gradient (CG) method for the static and eddy current case (in time domain) and the Quasi Minimal Residual (QMR) method for the time harmonic case. The iteration was stopped as soon as an error reduction in the preconditioner energy norm has been reduced by a factor \( 10^{-6} \) for the PCG method. In the time harmonic case (QMR solver) we use

\[
\|f_{h} - K_{h} w_{h}\|_{2} \leq 10^{-6} \|f_{h}\|_{2}.
\]

For all calculations a \( V(2,2) \)-cycle has been applied and the coarsest matrix equation is solved by a Cholesky factorization (degrees of freedom \( \leq 500 \)). All computations were done on a PC Pentium 1.7 GHz.

A good measure for the speed of coarsening is the so called grid complexity, which is given by

\[
GC(K_{h}) = \frac{1}{M_{1}} \sum_{i=1}^{L} M_{i}.
\]

(24)

with \( L \) the number of levels and \( M_{i} \) the number of nodes (edges) for level \( i \). This number is close to 1, if the reduction of unknowns is done very fast. If the number is very large then the coarsening is usually very slow. A second measure which is more related to the memory consumption and arithmetic costs is the operator complexity, i.e.,

\[
OC(K_{h}) = \frac{1}{N_{1}} \sum_{i=1}^{L} N M E_{i} \cdot N_{i}.
\]

(25)

where \( N M E_{i} \) denotes the average number of non-zero entries on level \( i \) and \( N_{i} \) the number of unknowns on this level. This number gives an idea of how much memory is used with respect to the finest grid. The same applies for the arithmetic costs. Again this number is close to 1 if only a small amount of memory is required. The abbreviation MB denotes the amount of MegaByte used. The computations with Lagrange and Nédélec FE-functions were always done on the same FE-mesh. We want to emphasize, that \( N_{h} = p |\omega_{h}| \) for node (static case: \( p = 3 \); eddy current case: \( p = 4 \) ) and \( N_{h} = |\omega_{h}| \) for edge FE-discretization.

A. Static Analysis

For the computational domain we consider the geometry of TEAM 20 (see Fig. 7), which has been discretized by tetrahedra elements.

Table I displays the evaluated grid complexity \( GC \) and operator complexity \( OC \) as well as required memory for the nodal (defined by \( n \)) and edge (defined by \( e \)) case. \( N_{h} \) defines the number of unknowns. It can be clearly seen, that the required memory scales optimal with the number of unknowns and the values for \( OC \) and \( GC \) are close to 1. The number of iterations as well as elapsed CPU-times are shown in Tab. II. The short time for performing the setup makes the AMG solvers very attractive for any nonlinear problem.

![Fig. 7. FE-mesh of TEAM 20 (without air region)](image-url)
### TABLE I
TEAM 20: Complexities and memory requirement

<table>
<thead>
<tr>
<th>$N_h$</th>
<th>Setup [s]</th>
<th>Solve [s]</th>
<th>Iter</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.263</td>
<td>2.253</td>
<td>1.4</td>
<td>1.07</td>
</tr>
<tr>
<td>8.022</td>
<td>16.217</td>
<td>1.3</td>
<td>1.06</td>
</tr>
<tr>
<td>56.673</td>
<td>122.762</td>
<td>1.3</td>
<td>1.07</td>
</tr>
</tbody>
</table>

### TABLE II
TEAM 20: CPU times and number of iterations

<table>
<thead>
<tr>
<th>$N_h$</th>
<th>Setup [s]</th>
<th>Solve [s]</th>
<th>Iter</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.263</td>
<td>2.253</td>
<td>0.2</td>
<td>0.2</td>
</tr>
<tr>
<td>8.022</td>
<td>16.217</td>
<td>0.4</td>
<td>0.5</td>
</tr>
<tr>
<td>56.673</td>
<td>122.762</td>
<td>1.7</td>
<td>2.9</td>
</tr>
</tbody>
</table>

### TABLE III
MRI-scanner: Complexities and memory requirement

<table>
<thead>
<tr>
<th>$N_h$</th>
<th>Setup [s]</th>
<th>Solve [s]</th>
<th>MB</th>
</tr>
</thead>
<tbody>
<tr>
<td>27.834</td>
<td>61.342</td>
<td>2.5</td>
<td>15</td>
</tr>
<tr>
<td>88.053</td>
<td>197.375</td>
<td>7.7</td>
<td>31.2</td>
</tr>
<tr>
<td>162.882</td>
<td>368.131</td>
<td>14.6</td>
<td>75.2</td>
</tr>
</tbody>
</table>

### B. Transient Analysis

In order to show the performance of the proposed enhanced AMG methods for an eddy current problem, we present results of 3D magnetic field computations for a simplified MRI scanner with $z$-gradient coil, as shown in Fig. 8 [24]. Here, gradient and magnet coils are assumed as smeared cylindrical coils. Furthermore, only the three inner cryostat cylinders are modeled. Table III displays the values for the grid complexity $GC$, the operator complexity $OC$ and the required memory. Again a very good performance with optimal memory requirement can be found.

### TABLE IV
MRI-scanner: CPU times and number of iterations

<table>
<thead>
<tr>
<th>$N_h$</th>
<th>Setup [s]</th>
<th>Solve [s]</th>
<th>Iter</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.786</td>
<td>1.2</td>
<td>1.02</td>
<td>9</td>
</tr>
<tr>
<td>37.390</td>
<td>1.2</td>
<td>1.02</td>
<td>72</td>
</tr>
<tr>
<td>296.064</td>
<td>1.2</td>
<td>1.03</td>
<td>580</td>
</tr>
</tbody>
</table>

### C. Time Harmonic Analysis

Finally we show the performance of the AMG method for the harmonic analysis. We present results of 3D magnetic field computations for an electric transformer as shown in Fig. 9. In Tab. V the values for the grid complexity $GC$, the operator complexity $OC$, the required memory, number of iterations and the CPU-times can be found. For this example we only used a edge discretization with tetrahedral elements.

### TABLE V
Electric Transformer: Performance for different FE-meshes

<table>
<thead>
<tr>
<th>$N_h$</th>
<th>Setup [s]</th>
<th>Solve [s]</th>
<th>Iter</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.022</td>
<td>16.217</td>
<td>1.3</td>
<td>1.06</td>
</tr>
<tr>
<td>22.5</td>
<td>51.9</td>
<td>16</td>
<td></td>
</tr>
</tbody>
</table>

### VII. Conclusion

The presented AMG solvers are well suited for the efficient solution – both concerning CPU time and memory requirements – of algebraic systems of equations arising from nodal as well as edge FE discretizations of Maxwell’s equation. Especially the presented algorithms for the coarsening strategy make the solvers very attractive also...
for nonlinear electromagnetic field problems, since the setup time can be kept very small.

Further research will concentrate on improvements of the prolongation operators to obtain even better convergence rates. One possible method was proposed in [34] (so called smoothed aggregation), which could be applied for our problem classes.

If even more speedup is required for practical applications, then the presented AMG methods can be parallelized on distributed memory computers. First promising results can be found in [11], [12].

VIII. Acknowledgment

This work was supported by ‘Deutsche Forschungsgemeinschaft, Sonderforschungsbereich 603 and the Austrian Science Foundation - ‘Fonds zur Förderung der wissenschaftlichen Forschung (FWF)’ - under the grant P14953 ‘Robust Algebraic MultiGrid Methods and their Parallelization’.

REFERENCES


Manfred Kaltenbacher
Department of Sensor Technology
University of Erlangen, Germany
manfred@ise.e-technik.uni-erlangen.de

Stefan Reitzinger
Institute of Computational Mathematics
University of Linz, Austria
reitz@numa.uni-linz.ac.at