IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 3 September 2024, accepted 8 October 2024, date of publication 11 October 2024, date of current version 24 October 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3478338

RESEARCH ARTICLE

Linear-Time Solution of 3D Magnetostatics With Aggregation-Based Algebraic Multigrid Solvers

FEDERICO MORO[1], (Senior Member, IEEE), LORENZO CODECASA[2], (Member, IEEE), AND ARTEM NAPOV[3]

[1]Dipartimento di Ingegneria Industriale, Università degli Studi di Padova, 35131 Padua, Italy
[2]Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, 20133 Milan, Italy
[3]Service de Métrologie Nucléaire, Université Libre de Bruxelles, 1050 Bruxelles, Belgium

Corresponding author: Federico Moro (federico.moro@unipd.it)

This work was supported by the University of Padua.

ABSTRACT Two different approaches for the linear-time solution of three-dimensional magnetostatic field problems encompassing large-scale linear systems with millions of degrees of freedom are presented. The φ–method is formulated in terms of nodal variables, i.e., magnetic scalar potentials, and involves the solution of a curl-curl linear system for pre-processing and of a div-grad linear system. The A–method is formulated in terms of edge variables, i.e., line integrals of the magnetic vector potential, and involves the solution of a curl-curl linear system only. It is shown that these linear systems can be solved by flexible conjugate gradient in combination with the aggregation-based multigrid preconditioner (AGMG), tailored for div-grad problems, and with the aggregation-based (auxiliary-space) multigrid preconditioner (AGMG_CC), tailored for curl-curl problems. The robustness and efficiency of these solvers is illustrated using magnetostatic problems of practical interest, with linear or nonlinear media and having parts with complex geometry. Numerical results show that both AGMG and AGMG_CC are able to attain linear solution time in all cases considered, while being both faster and more robust than state-of-the-art algebraic multigrid solvers and preconditioned Krylov subspace solvers, typically adopted in commercial finite element software for electromagnetic analysis.

INDEX TERMS Finite element method, cell method, computer-aided engineering, iterative solver, algebraic multigrid, auxiliary space preconditioner, linear time, nonlinear, magnetostatics, large-scale linear system.

I. INTRODUCTION

The numerical discretization of real-life electromagnetic field problems typically leads to large-scale sparse linear systems encompassing millions of degrees of freedom (DOFs). This is in particular the case when three-dimensional (3D), non-linear, time-dependent problems are considered. Since this often repetitive system solution is commonly the most time-consuming step in electromagnetic (EM) simulation, its efficiency and robustness are of crucial importance.

In particular, finite element method (FEM) or cell method (CM) discretizations of magnetostatic field problems lead to large sparse, symmetric and positive or semipositive definite linear systems [1], [2]. These systems are commonly solved by a Krylov solver such as the preconditioned conjugate gradient (PCG) method [3], [4]. For optimal performance the latter must be combined with an effective preconditioner, whose choice depends on the type of associated linear system.

For 3D magnetostatics, discretization approaches based on the use of the magnetic scalar potential lead to a div-grad linear system, for which it is desirable to use PCG with a standard multigrid preconditioner [5], [6]. In fact, the resulting iterative solver typically exhibits a linear time, i.e., proportional to the number of DOFs. In particular, this occurs when the number of iterations of the iterative solver is more or less independent of the number of DOFs and run time per iteration increases at most linearly with the number of DOFs. Multigrid preconditioners (or solvers) are subdivided into two different classes, namely *geometric* and *algebraic*. Geometric multigrid requires a hierarchy of

The associate editor coordinating the review of this manuscript and approving it for publication was Ladislau Matekovits.

nested discretization grids, where the finest grid corresponds to the linear system to be solved and the coarsest grid is sufficiently small [7]. This can be a disadvantage if some parts of the model have parts with complex shapes and/or large aspect ratios. Unlike geometric multigrid, algebraic multigrid (AMG) only requires information from the system matrix itself and therefore has broader applicability [8], [9]. In particular, aggregation-based algebraic multigrid methods like the one implemented in the AGMG software are known to be more efficient and robust than standard AMG implementations [10], [11]. For example, numerical tests conducted in [12] have shown that AGMG performs better than state-of-the-art algebraic multigrid solvers such as BoomerAMG [13].

Another approach for the numerical modeling of 3D magnetostatics is based on the use of magnetic vector potential, and leads to a curl-curl linear system, for which standard AMG cannot be effectively used (see, e.g., [14], [15]). The use of auxiliary space (AS) preconditioner has been proposed in the literature to overcome the difficulties of AMG [16], [17], [18]. One step of this preconditioner amounts to an application of three approximate solutions of auxiliary linear systems for which standard AMG can be used. The state-of-the-art implementation of an AS preconditioner is the Auxiliary-space Maxwell Solver (AMS) preconditioner of the HYPRE library [13], which relies on BoomerAMG. However, as mentioned above, solvers like AGMG are more efficient than BoomerAMG, and therefore would likely improve the performance of the auxiliary space preconditioners. On the other hand, as illustrated with numerical experiments in Section V, the AMS occasionally suffers from a lack of robustness and exhibits superlinear time when applied to linear systems arising from the discretization of 3D magnetostatic problems of practical interest. For these reasons, an auxiliary space preconditioner implementation based on AGMG was considered here and successfully tested on magnetostatic problems with linear and nonlinear media and having parts with complex geometry. This implementation is referred to as AGgregation-based MultiGrid software for Curl-Curl problems (AGMG_CC).

It has to be noted that the efficient solution of nonlinear magnetostatics is particularly important since most of electrical devices have ferromagnetic parts with nonlinear behavior. The assumption of linear media in these cases would lead to unrealistic estimates of electrical parameters of the device. For an accurate simulation, a more elaborated solution strategy is needed: i) first, a set of nonlinear algebraic equations is generated from partial differential equations governing the field problem using a numerical discretization technique, ii) these nonlinear equations are then solved using an iterative technique such as the fixed-point (FP) iteration method or the Newton-Raphson (NR) method which in turn involve solving at each step of a large-scale sparse linear system. Two FP algorithms are proposed here, to exploit both AGMG and AGMG_CC also for the solution of nonlinear magnetostatics.

The remainder of this paper is organized as follows. The nonlinear magnetostatic field problem is described in the continuous setting in Section II, where the classical φ–formulation and A–formulation are presented. Their discretization with the cell method is discussed in Section III, together with solution algorithms adopted for nonlinear magnetostatics. It is shown that both φ–method and A–method lead to the solution of div-grad and curl-curl linear systems that can be treated in linear time by flexible conjugate gradient solver with AGMG and AGMG_CC preconditioners, respectively, described in Section IV. Numerical results in Section V, which include the solution of realistic magnetostatic problems with linear and nonlinear materials, show the effectiveness and robustness of both AGMG and AGMG_CC.

II. MAGNETOSTATIC PROBLEM

In a magnetostatic field problem the computational domain Ω is typically partitioned into the following subdomains: Ω_m, that contains linear or nonlinear magnetic media, such as ferromagnetic materials, and in which no currents are present; Ω_0, that contains magnetic field sources such as coils carrying direct current; $\Omega \setminus (\Omega_m \cup \Omega_0)$, which is the rest of the domain, and is made up of non-magnetic materials such as air, with magnetic permeability equal to μ_0 (the permeability of vacuum). Ω is assumed to be bounded, with boundary $\Gamma = \partial\Omega$.

Maxwell's equations for magnetostatics in Ω are [19]:

$$\nabla \times \mathbf{H} = \mathbf{J}_0, \tag{1}$$

$$\nabla \cdot \mathbf{B} = 0. \tag{2}$$

where $\nabla\times$ indicates the curl operator, $\nabla\cdot$ indicates the divergence operator, and vectors $\mathbf{H}$, $\mathbf{J}_0$, $\mathbf{B}$ are the magnetic field, the source current density, and the magnetic flux density, respectively. In a magnetostatic problem $\mathbf{J}_0$ is assigned within coils and must fulfill the following constraints:

$$\nabla \cdot \mathbf{J}_0 = \mathbf{0}, \quad \text{in } \Omega_0, \tag{3}$$

$$\mathbf{J}_0 \cdot \mathbf{n} = 0, \quad \text{on } \partial\Omega_0, \tag{4}$$

where the dot symbol in (4) indicates the inner product and $\mathbf{n}$ is the exterior unit normal. Note that (3) is a direct consequence of (1) and (4) is written for coils without terminals.

Maxwell's equations are complemented by constitutive relationships. For nonlinear isotropic media, the magnetic permeability is defined as a nonlinear scalar function [20]:

$$\mu(H) = \frac{B(H)}{H}, \tag{5}$$

which is the ratio between the field strengths of $\mathbf{B}$ and $\mathbf{H}$. This function is obtained by interpolating the B-H curve, experimentally evaluated, and is bounded above and below as $\mu_0 \leq \mu \leq \mu_{\max}$. The magnetic constitutive relationship for nonlinear isotropic media thus becomes:

$$\mathbf{B} = \mu(H)\,\mathbf{H}. \tag{6}$$

It has to be noted that the behavior of linear materials can also be described by (6), assuming the magnetic permeability to be piecewise constant and independent of the field strength H.

To obtain a unique solution to the magnetostatic field problem, it is necessary to impose appropriate boundary conditions (BCs). For the sake of simplicity, the following homogeneous boundary condition is considered in this work:

$$\mathbf{B} \cdot \mathbf{n} = 0, \quad \text{on } \Gamma, \tag{7}$$

which is suitable to model in practical applications either symmetry conditions or magnetic wall BCs (e.g., tangent field at the boundary of the computational domain).

Finally, at each interface Σ between subdomains with different material properties, the continuity of the tangent component of the magnetic field and the continuity of the normal component of magnetic flux density must also be satisfied:

$$[\mathbf{H}] \times \mathbf{n} = \mathbf{0}, \quad \text{on } \Sigma, \tag{8}$$

$$[\mathbf{B}] \cdot \mathbf{n} = 0, \quad \text{on } \Sigma, \tag{9}$$

where the cross symbol indicates the cross product and the square brackets indicate the field jump across Σ.

The solution of magnetostatics typically involves the magnetic scalar potential φ or the magnetic vector potential $\mathbf{A}$, to reduce the number of unknowns and obtain a single partial differential equation. This leads to two equivalent formulations, briefly recalled below [21], [22].

A. φ–FORMULATION

Assuming a simply connected domain Ω, as usual in practical magnetostatic problems, the magnetic scalar potential can be introduced by means of the following decomposition:

$$\mathbf{H} = \mathbf{H}_0 - \nabla\varphi, \quad \text{in } \Omega, \tag{10}$$

which splits the magnetic field into a curl-free component $\nabla\varphi$ (i.e., the gradient of the magnetic scalar potential) and the so-called *source field* $\mathbf{H}_0$ satisfying:

$$\nabla \times \mathbf{H}_0 = \mathbf{J}_0, \quad \text{in } \Omega. \tag{11}$$

The source field, defined in the whole domain Ω, does not have to satisfy any BCs and can be obtained in several ways. For instance, it can be the magnetic field generated by coils in the absence of magnetic media in Ω_m (i.e., calculated from Biot-Savart's law or from the numerical solution of a magnetostatic problem), or it can be calculated (at the discrete level) using numerical procedures based on graph theory or functional analysis, without any physical interpretation (see, e.g., [23] for a thorough discussion). The use of Biot-Savart's law is typically time-consuming because it requires the evaluation of integrals with Gaussian quadrature. To ensure a fast calculation of $\mathbf{H}_0$, a purely algebraic procedure, based on AGMG_CC, is proposed in Section III-B.

By plugging (6) in (2) and using (10), the following div-grad partial differential equation is obtained:

$$\nabla \cdot \mu(H)\, \nabla\varphi = \nabla \cdot \mu(H)\, \mathbf{H}_0, \quad \text{in } \Omega. \tag{12}$$

To get a unique potential, up to a constant, (12) must be solved together with (7). The latter rewritten in terms of φ becomes:

$$\mu(H)\, \nabla\varphi \cdot \mathbf{n} = \mu(H)\, \mathbf{H}_0 \cdot \mathbf{n}, \quad \text{on } \Gamma. \tag{13}$$

Once (12) has been solved in terms of magnetic scalar potential, the magnetic field is obtained using (10).

B. A–FORMULATION

Assuming this time a contractible domain Ω, as usual in practical magnetostatic problems, the magnetic vector potential is introduced by means of an equivalent form of Gauss' law (2):

$$\mathbf{B} = \nabla \times \mathbf{A}, \quad \text{in } \Omega. \tag{14}$$

By inverting the constitutive relationship (6), one obtains:

$$\mathbf{H} = \nu(H)\, \mathbf{B}, \tag{15}$$

where $\nu = \mu^{-1}$ is the magnetic reluctivity.

By plugging (14) in (15) and using (1), the following curl-curl partial differential equation is obtained:

$$\nabla \times \nu(H) \nabla \times \mathbf{A} = \mathbf{J}_0, \quad \text{in } \Omega. \tag{16}$$

To complete the field problem formulation, (16) must be solved with (7). As a consequence of Stokes' theorem, the latter can be expressed in terms of the vector potential as:

$$\mathbf{A} \times \mathbf{n} = \mathbf{0}, \quad \text{on } \Gamma. \tag{17}$$

Once (16) has been solved in terms of magnetic vector potential, the magnetic flux density is obtained by (14); then, the magnetic field is calculated from $\mathbf{B}$ using (15).

III. CELL METHOD DISCRETIZATION

Formulations described in Section II can be discretized using the FEM (see, e.g., [22]), which however requires Gaussian quadrature for the evaluation of the mass matrix coefficients. In this work, an alternative approach is adopted, namely the CM, which does not use Gaussian quadrature. The numerical approaches resulting from the CM discretization of the φ–formulation and the A–formulation, discussed in Section II, will be called hereinafter *φ–method* and *A–method*, respectively.

The discretization framework of the CM, which is detailed in [24], is briefly summarized as follows. The computational domain is first meshed into a tetrahedral mesh (the so-called *primal grid* $\mathcal{G}_\Omega$), made up of different geometric entities: nodes, edges, faces and volumes. Then, the corresponding *dual grid* $\widetilde{\mathcal{G}}_\Omega$ is obtained by joining the centers of tetrahedrons and those of its triangular faces. By using the approach proposed in [25] to properly incorporate BCs in the CM, an *augmented dual grid* is constructed from $\mathcal{G}_\Omega$ as the union of $\widetilde{\mathcal{G}}_\Omega$ and the dual grid $\widetilde{\mathcal{G}}_\Gamma$, the latter being obtained

from the boundary mesh. In this way there is a one-to-one correspondence between the geometric entities of the primal and dual grids. Any physical field in the CM can be interpolated by piecewise constant edge or face functions, such as those defined in [26], so that vector fields are represented on $\mathcal{G}_\Omega$ as arrays of DOFs. Differential operators are transformed into incidence matrices using Stokes' theorem. Finally, by means of the so-called *energy approach*, magnetic material properties are transformed into positive-definite mass matrices [26]. In this way partial differential equations governing the field problem can be written directly in algebraic fashion. Discrete topological and material laws are assembled together to form a system of equations to be solved in terms of DOFs. In the post-processing, locally approximated fields can be obtained from the arrays of DOFs using piecewise constant functions.

A. SOURCE CURRENTS IN COILS

The calculation of source currents in coils is a pre-processing step common to both φ–method and A–method. A novel procedure for the calculation of the source current in current-driven coils is here proposed, based on the use of very efficient and robust AGMG and AGMG_CC solvers. This procedure extends the one already proposed in [27], based on less efficient iterative solvers for rectangular systems such as Least Squares Minimal Residual (LSMR) or, equivalently, Least Squares with QR-factorization (LSQR).

For the sake of simplicity, a single coil without terminals is considered, but the procedure can easily be extended to the case of multiple coils with terminals. According to [27], the source current density $\mathbf{J}_0$ can be decomposed as:

$$\mathbf{J}_0 = \nabla \times \mathbf{T}_0 + I_0\, \mathbf{K}_0, \tag{18}$$

where I_0 is the current of the coil, $\mathbf{T}_0$ is an electric vector potential in Ω_0 such that $\mathbf{T}_0 \times \mathbf{n} = \mathbf{0}$ on $\partial\Omega_0$. The topological field $\mathbf{K}_0$ in (18) is a div-free field that is not a curl of any vector field, such that:

$$\int_{\Sigma_0} \mathbf{K}_0 \cdot \mathbf{n}\, d\Sigma = 1, \tag{19}$$

where Σ_0 is an arbitrary cut surface of Ω_0, on which I_0 is enforced. Integrating (18) over each primal face of $\mathcal{G}_{\Omega_0}$ and using Stokes' theorem, the following decomposition of source currents is obtained:

$$\mathbf{j}_{0,\Omega_0} = \mathbf{C}_{\Omega_0} \mathbf{t}_{0,\Omega_0} + I_0\, \mathbf{k}_{0,\Omega_0}, \tag{20}$$

where $\mathbf{j}_{0,\Omega_0} = (j_{0,f})_{f\in\mathcal{G}_{\Omega_0}}$ is the array of fluxes of $\mathbf{J}_0$ through any primal face f of $\mathcal{G}_{\Omega_0}$, i.e., $j_f = \int_f \mathbf{J}_0 \cdot d\mathbf{s}$, $\mathbf{C}_{\Omega_0}$ is the discrete curl operator defined on $\mathcal{G}_{\Omega_0}$ (i.e., the faces-to-edges incidence matrix with 0, ±1 coefficients), $\mathbf{t}_{\Omega_0} = (t_e)_{e\in\mathcal{G}_{\Omega_0}}$ is the array of line integrals of $\mathbf{T}_0$ along each primal edge e of $\mathcal{G}_{\Omega_0}$, i.e., $t_e = \int_e \mathbf{T}_0 \cdot d\mathbf{l}$. Similarly to the array of source currents, $\mathbf{k}_{0,\Omega_0} = (k_{0,f})_{f\in\mathcal{G}_{\Omega_0}}$ is the array of fluxes of $\mathbf{K}_0$ through any primal face f of $\mathcal{G}_{\Omega_0}$, i.e., $k_f = \int_f \mathbf{K}_0 \cdot d\mathbf{s}$.

Integrating the charge conservation law (3) over the primal volumes of $\mathcal{G}_{\Omega_0}$ and using the divergence theorem, Kirchhoff's current law is obtained. According to (4), source currents on boundary faces $\mathbf{j}_{0,\partial\Omega_0}$ must be null. Therefore:

$$\mathbf{D}_{\Omega_0} \mathbf{j}_{0,\Omega_0} = \mathbf{0}, \tag{21}$$

$$\mathbf{j}_{0,\partial\Omega_0} = \mathbf{0}, \tag{22}$$

where $\mathbf{D}_{\Omega_0}$ is the discrete divergence operator of $\mathcal{G}_{\Omega_0}$ (i.e., the volume-to-faces incidence matrix with 0, ±1 coefficients). Substituting (20) in (21) and exploiting the identity $\mathbf{D}_{\Omega_0}\mathbf{C}_{\Omega_0} = \mathbf{0}$, the following rectangular system is obtained:

$$\mathbf{D}_{\Omega_0} \mathbf{k}_{0,\Omega_0} = \mathbf{0}. \tag{23}$$

A non-trivial solution of (23) can be obtained by enforcing: i) null fluxes through all the faces of $\mathcal{G}_{\partial\Omega_0}$, ii) a unit flux on an arbitrary face of $\mathcal{G}_{\Sigma_0}$, i.e., the discretized cut surface on which the source current is imposed, and iii) null fluxes on the remaining faces of $\mathcal{G}_{\Sigma_0}$. Let $\mathbf{k}_{0,\partial\Omega_0\cup\Sigma_0}$ be the array of constrained variables. The following splitting then results:

$$\mathbf{k}_{0,\Omega_0} = \mathbf{k}_{0,\partial\Omega_0\cup\Sigma_0} + \mathbf{Q}_{\Omega_0}^{\mathrm{T}}\, \widehat{\mathbf{k}}_{0,\Omega_0}, \tag{24}$$

where $\mathbf{Q}_{\Omega_0}$ is a selection operator made of {0, 1} coefficients that extracts free variables $\widehat{\mathbf{k}}_{0,\Omega_0}$ from $\mathbf{k}_{0,\Omega_0}$ and the operator $(\cdot)^{\mathrm{T}}$ indicates the matrix transpose. Substituting (24) in (23), the following reduced rectangular linear system is obtained:

$$\widehat{\mathbf{D}}_{\Omega_0} \widehat{\mathbf{k}}_{0,\Omega_0} = \widehat{\mathbf{g}}_{\Omega_0}, \tag{25}$$

where $\widehat{\mathbf{D}}_{\Omega_0} = \mathbf{D}_{\Omega_0}\mathbf{Q}_{\Omega_0}^{\mathrm{T}}$ and $\widehat{\mathbf{g}}_{\Omega_0} = -\mathbf{D}_{\Omega_0}\mathbf{k}_{0,\partial\Omega_0\cup\Sigma_0}$.

By letting $\widehat{\mathbf{k}}_{0,\Omega_0} = \widehat{\mathbf{D}}_{\Omega_0}^{\mathrm{T}} \widehat{\boldsymbol{\psi}}_{\Omega_0}$, one admissible solution of (25) can be found by solving the div-grad linear system:

$$\widehat{\mathbf{D}}_{\Omega_0} \widehat{\mathbf{D}}_{\Omega_0}^{\mathrm{T}}\, \widehat{\boldsymbol{\psi}}_{\Omega_0} = \widehat{\mathbf{g}}_{\Omega_0}, \tag{26}$$

which can be regarded as a div-grad system due to the following physical interpretation: it is related to an optimal-flow problem (e.g., an electric circuit formed by resistors with unit resistance) defined on the dual graph of the mesh in Ω_0, the latter being obtained by joining the centers of cells.

The vector $\mathbf{t}_{0,\Omega_0}$ in (20) can be obtained by solving an electrokinetic problem inside Ω_0 under the BCs given in (22):

$$\widetilde{\mathbf{C}}_{\Omega_0} \widetilde{\mathbf{e}}_{\Omega_0} = \mathbf{0}, \tag{27}$$

$$\widetilde{\mathbf{e}}_{\Omega_0} = \mathbf{M}_{\rho,\Omega}\, \mathbf{j}_{0,\Omega_0}, \tag{28}$$

where the dual curl operator is defined as $\widetilde{\mathbf{C}}_{\Omega_0} = \mathbf{C}_{\Omega_0}^{\mathrm{T}}$ and $\widetilde{\mathbf{e}}_{\Omega_0} = (\widetilde{e}_{\widetilde{e}})_{\widetilde{e}\in\widetilde{\mathcal{G}}_{\Omega_0}}$ is the array of electromotive forces (emfs) defined on dual edges $\widetilde{e}$ of $\widetilde{\mathcal{G}}_{\Omega_0}$, i.e., $\widetilde{e}_{\widetilde{e}} = \int_{\widetilde{e}} \mathbf{E} \cdot d\mathbf{l}$. The coefficients of resistance matrix $\mathbf{M}_{\rho,\Omega} = (m_{\rho,ff'})_{f,f'\in\mathcal{G}_{\Omega_0}}$ are:

$$m_{\rho,ff'} = \int_\Omega \rho(x)\, \mathbf{w}_f(x) \cdot \mathbf{w}_{f'}(x)\, dx, \tag{29}$$

where ρ is the resistivity of the coil (i.e., it can be set to 1) and $\mathbf{w}_f$ are the piecewise constant face basis functions defined in [26]. By inserting (28) in (27) and using (20), the following curl-curl linear system is obtained:

$$\mathbf{C}_{\Omega_0}^{\mathrm{T}} \mathbf{M}_{\rho,\Omega_0} \mathbf{C}_{\Omega_0}\, \mathbf{t}_{0,\Omega_0} = -I_0\, \mathbf{C}_{\Omega_0}^{\mathrm{T}} \mathbf{M}_{\rho,\Omega_0} \mathbf{k}_{0,\Omega_0}. \tag{30}$$

Once topological discrete fields $\mathbf{k}_{0,\Omega_0}$ and $\mathbf{t}_{0,\Omega_0}$ have been found, div-free source currents can be obtained from (20).

B. φ METHOD

Integrating (10) along each primal edge of $\mathcal{G}_\Omega$ and using Stokes' theorem gives the corresponding discrete field decomposition:

$$\mathbf{h}_\Omega = \mathbf{h}_{0,\Omega} - \mathbf{G}_\Omega \boldsymbol{\varphi}_\Omega, \tag{31}$$

where $\mathbf{h}_\Omega = (h_e)_{e\in\mathcal{G}_\Omega}$ is the array of magnetomotive forces (mmfs) $h_e = \int_e \mathbf{H}\cdot d\mathbf{l}$, i.e., line integrals of $\mathbf{H}$ along each primal edge e of $\mathcal{G}_\Omega$, $\mathbf{h}_{0,\Omega}$ is the array of mmfs related to the source field and $\boldsymbol{\varphi}_\Omega = (\varphi_n)_{n\in\mathcal{G}_\Omega}$ is the array of nodal values of the magnetic scalar potential. The matrix $\mathbf{G}_\Omega$ is the discrete gradient operator of $\mathcal{G}_\Omega$ (i.e., the edges-to-nodes incidence matrix with 0, ±1 coefficients).

Unlike traditional approaches for computing the source magnetic field, based on the time-consuming calculation of Biot-Savart's integrals [19] or spanning tree techniques [20], [21], the fully algebraic procedure proposed in [28] is adopted here. The key advantage is that the final linear system for the computation of source mmfs $\mathbf{h}_{0,\Omega}$ is amenable to linear time solution using algebraic multigrid.

Integrating (11) over each primal face of $\mathcal{G}_\Omega$ and using Stokes' theorem yields:

$$\mathbf{C}_\Omega\,\mathbf{h}_{0,\Omega} = \mathbf{j}_{0,\Omega}, \tag{32}$$

where $\mathbf{C}_\Omega$ is the discrete curl operator of $\mathcal{G}_\Omega$ and $\mathbf{j}_{0,\Omega} = (j_{0,f})_{f\in\mathcal{G}_\Omega}$ is the array of source currents defined on $\mathcal{G}_\Omega$, where $j_{0,f} = 0$ for any $f \notin \mathcal{G}_{\Omega_0}$. As proven in [28], the rectangular linear system (32) is equivalent to the curl-curl linear system:

$$\mathbf{C}_\Omega^{\mathrm{T}}\mathbf{C}_\Omega\,\mathbf{h}_{0,\Omega} = \mathbf{C}_\Omega^{\mathrm{T}}\,\mathbf{j}_{0,\Omega}. \tag{33}$$

The local constitutive relationship (6) is discretized using the energy approach described in [26] and assembled into the corresponding global relationship defined on $\mathcal{G}_\Omega$, yielding:

$$\widetilde{\mathbf{b}}_\Omega = \mathbf{M}_{\mu(H),\Omega}\,\mathbf{h}_\Omega, \tag{34}$$

where $\widetilde{\mathbf{b}}_\Omega = (b_{\widetilde{f}})_{\widetilde{f}\in\widetilde{\mathcal{G}}_\Omega}$ is the array of magnetic fluxes $b_{\widetilde{f}} = \int_{\widetilde{f}} \mathbf{B}\cdot d\mathbf{s}$ through any dual face $\widetilde{f}$ of $\widetilde{\mathcal{G}}_\Omega$. In presence of nonlinear media, the coefficients of the permeance matrix $\mathbf{M}_{\mu,\Omega} = (m_{\mu,ee'})_{e,e'\in\mathcal{G}_\Omega}$ depend on the magnetic field strength and are defined as:

$$m_{\mu,ee'} = \int_\Omega \mu(H(x))\,\mathbf{w}_e(x)\cdot\mathbf{w}_{e'}(x)\,dx, \tag{35}$$

where $\mathbf{w}_e$ are the piecewise constant edge basis functions defined in [26]. Note that $\mathbf{M}_{\mu,\Omega}$ is positive-definite since $\mu(H) > 0$ and the matrix assembly does not require Gaussian quadrature since μ is assumed to be piecewise constant.

Integrating (2) over each dual cell of $\widetilde{\mathcal{G}}_\Omega$ (a polyhedron) and using the divergence theorem gives the global Gauss' law:

$$\widetilde{\mathbf{D}}_\Omega\,\widetilde{\mathbf{b}}_\Omega + \widetilde{\mathbf{D}}_{\Omega\Gamma}\,\widetilde{\mathbf{b}}_\Gamma = \mathbf{0}, \tag{36}$$

where $\widetilde{\mathbf{D}}_\Omega = -\mathbf{G}_\Omega^{\mathrm{T}}$ is the dual divergence operator (or dual volume-to-face incidence matrix), $\widetilde{\mathbf{D}}_{\Omega\Gamma}$ is the surface divergence operator and $\widetilde{\mathbf{b}}_\Gamma$ is the array of magnetic fluxes through the dual faces of $\widetilde{\mathcal{G}}_\Gamma$. $\widetilde{\mathbf{D}}_{\Omega\Gamma}$ is constructed by noting that its transpose is a selection matrix, which extracts the primal nodes of $\mathcal{G}_\Gamma$ from those of $\mathcal{G}_\Omega$. It has to be noted that, by letting $\widetilde{\mathbf{b}}_\Gamma = \mathbf{0}$ in (36), i.e., null boundary fluxes, the boundary condition (7) is automatically enforced.

The final system of nonlinear equations is obtained using the same scheme adopted at the continuous level: (34) is inserted in (36) and (31) is used to express mmfs in terms of scalar potentials. The final div-grad system thus becomes:

$$\mathbf{G}_\Omega^{\mathrm{T}}\,\mathbf{M}_{\mu(H),\Omega}\,\mathbf{G}_\Omega\,\boldsymbol{\varphi}_\Omega = \mathbf{G}_\Omega^{\mathrm{T}}\,\mathbf{M}_{\mu(H),\Omega}\,\mathbf{h}_{0,\Omega}, \tag{37}$$

where the right-hand side (RHS) is found after solving (33).

The nonlinear system (37) is solved using the fixed-point iteration method, which has some remarkable advantages: it is proven to be more robust than the Newton-Raphson method, i.e., less prone to convergence problems, and does not require careful choice of the trial solution, which is typically a very problem-dependent and time-consuming operation [29].

The main steps of the FP method to solve nonlinear magnetostatics with the φ–method are summarized in Algorithm 1.

At line 3, the initial magnetic permeability in any element of the mesh is set to the maximum permeability value $\mu_{\max}$, which is given by the B-H curve of the magnetic material.

At line 5, the solution process continues if the error $e^{(k)}$ is greater than a fixed threshold $\tau < 1$ and less than a fixed number of iterations $k_{\max}$.

At line 6, the coefficients of the permeance matrix are calculated with (35) for a given magnetic permeability value $\mu^{(k)}$, which is estimated for any cell at each iteration. Since $\mu^{(k)} > 0$ the permeance matrix is positive definite and the related linear system (37) is suitable for the AGMG solver.

At line 7, the coefficient matrix $\mathbf{K}_\Omega^{(k)}$ and the RHS $\mathbf{f}_\Omega^{(k)}$ of (37) are updated with the new permeance matrix $\mathbf{M}_{\mu^{(k)},\Omega}$.

At line 8, the residual at $k+1$th step is calculated using the Euclidean norm from the solution at kth step.

At line 10, the relative error $e^{(k+1)}$ is calculated as the ratio of residuals at $k+1$th step and at the first one. When the error is below the threshold, solution is assumed to have converged.

At line 15, the div-grad linear system (37) is iteratively solved using AGMG, starting from $\boldsymbol{\varphi}_\Omega^{(k)}$ (i.e., last computed solution) as a trial solution to accelerate convergence.

At line 17, the magnetic field is calculated at the center of each mesh cell from $\mathbf{h}_\Omega^{(k+1)}$ using piecewise constant edge functions, hence its strength $H^{(k+1)}$ is calculated with the Euclidean norm. By using piecewise constant edge functions, it can be proven that the interpolated magnetic field inside Ω_m (i.e., no source currents) is constant element-by-element since $\mathbf{C}_\Omega\,\mathbf{h}_\Omega^{(k+1)} = \mathbf{j}_\Omega$, with $j_{0,f} = 0$ for any $f \notin \mathcal{G}_{\Omega_0}$.

At line 18, for each cell, the magnetic flux density strength $B^{(k+1)}$ is calculated from $H^{(k+1)}$ by means of the B-H curve.

Finally, at line 19, the magnetic permeability value $\mu^{(k+1)}$ is updated for each cell as the ratio of $B^{(k+1)}$ and $H^{(k+1)}$.

C. A METHOD

Integrating (14) over each primal face of $\mathcal{G}_\Omega$ and using Stokes' theorem magnetic fluxes can be expressed in terms

Algorithm 1 Fixed-Point Iterations (φ–method)

$k = 0$
$e^{(0)} = 1$
$\mu^{(0)} = \mu_{\max}$ (for all mesh elements)
$\boldsymbol{\varphi}_\Omega^{(0)} = \mathbf{0}$
while $e^{(k)} > \tau \wedge k < k_{\max}$ **do**
  Assembly of permeance matrix $\mathbf{M}_{\mu^{(k)},\Omega}$
  Compute matrix and RHS of (37): $\mathbf{K}_\Omega^{(k)}$, $\mathbf{f}_\Omega^{(k)}$
  $r^{(k+1)} = \|\mathbf{K}_\Omega^{(k)} \boldsymbol{\varphi}_\Omega^{(k)} - \mathbf{f}_\Omega^{(k)}\|$
  if $k > 1$ **then**
    $e^{(k+1)} = r^{(k+1)}/r^{(1)}$
  else
    $e^{(1)} = 1$
  end if
  Use $\boldsymbol{\varphi}_\Omega^{(k)}$ as a trial solution for AGMG
  Find $\boldsymbol{\varphi}_\Omega^{(k+1)}$ by solving (37) with AGMG
  $\mathbf{h}_\Omega^{(k+1)}$ from $\boldsymbol{\varphi}_\Omega^{(k+1)}$ using (31)
  $H^{(k+1)}$ from $\mathbf{h}_\Omega^{(k+1)}$ by interpolation
  $B^{(k+1)}$ from $H^{(k+1)}$ with B-H curve
  $\mu^{(k+1)} = B^{(k+1)}/H^{(k+1)}$
  $\boldsymbol{\varphi}_\Omega^{(k)} \leftarrow \boldsymbol{\varphi}_\Omega^{(k+1)}$
  $k \leftarrow k + 1$
end while

of line integrals of the magnetic vector potential:

$$\mathbf{b}_\Omega = \mathbf{C}_\Omega \, \mathbf{a}_\Omega, \tag{38}$$

where $\mathbf{b}_\Omega = (b_f)_{f \in \mathcal{G}_\Omega}$ is the array of fluxes of $\mathbf{B}$ through any primal face f of $\mathcal{G}_\Omega$, i.e., $b_f = \int_f \mathbf{B} \cdot d\mathbf{s}$ and $\mathbf{a}_\Omega = (a_e)_{e \in \mathcal{G}_\Omega}$ is the array of line integrals of the magnetic vector potential along each primal edge e of $\mathcal{G}_\Omega$, that is $a_e = \int_e \mathbf{A} \cdot d\mathbf{l}$. Note that magnetic fluxes in (38) are defined on the primal grid instead of the dual grid as in the φ–method.

Using the energy approach the local magnetic constitutive relationship (15) can be discretized into the corresponding global relationship defined on $\mathcal{G}_\Omega$:

$$\widetilde{\mathbf{h}}_\Omega = \mathbf{M}_{\nu(H),\Omega} \, \mathbf{b}_\Omega, \tag{39}$$

where $\widetilde{\mathbf{h}}_\Omega$ is the array of mmfs along the dual edges of $\widetilde{\mathcal{G}}_\Omega$ and the coefficients of the reluctance matrix $\mathbf{M}_{\nu(H),\Omega}$ are:

$$m_{\nu,ff'} = \int_\Omega \nu(H(x)) \, \mathbf{w}_f(x) \cdot \mathbf{w}_{f'}(x) \, dx. \tag{40}$$

Note that $\mathbf{M}_{\nu,\Omega}$ is positive-definite since $\nu(H) > 0$ and the matrix assembly does not require Gaussian quadrature since the reluctivity is assumed to be piecewise constant.

Integrating (1) over the dual faces of $\widetilde{\mathcal{G}}_\Omega$ and using Stokes' theorem yields the discrete Ampère's law:

$$\widetilde{\mathbf{C}}_\Omega \, \widetilde{\mathbf{h}}_\Omega + \widetilde{\mathbf{C}}_{\Omega\Gamma} \, \widetilde{\mathbf{h}}_\Gamma = \widetilde{\mathbf{j}}_{0,\Omega}, \tag{41}$$

where $\widetilde{\mathbf{C}}_\Omega = \mathbf{C}_\Omega^{\mathrm{T}}$ is the dual curl operator (or dual face-to-edge incidence matrix) of $\widetilde{\mathcal{G}}_\Omega$, $\widetilde{\mathbf{C}}_{\Omega\Gamma}$ is the dual surface curl operator of $\widetilde{\mathcal{G}}_\Gamma$, and $\widetilde{\mathbf{h}}_\Gamma$ is the array of mmfs defined on the dual edges of $\widetilde{\mathcal{G}}_\Omega$. $\widetilde{\mathbf{C}}_{\Omega\Gamma}$ is constructed noting that its transpose is a selection matrix which extracts the primal edges of $\mathcal{G}_\Gamma$ from those of $\mathcal{G}_\Omega$. Finally, $\widetilde{\mathbf{j}}_{0,\Omega} = (j_{0,\widetilde{f}})_{\widetilde{f} \in \widetilde{\mathcal{G}}_\Omega}$ is the array of source currents $j_{0,\widetilde{f}} = \int_{\widetilde{f}} \mathbf{J}_0 \cdot d\mathbf{s}$ through dual faces $\widetilde{f}$ of $\widetilde{\mathcal{G}}_\Omega$, where $j_{0,\widetilde{f}} = 0$ for any $\widetilde{f} \notin \widetilde{\mathcal{G}}_{\Omega_0}$.

The RHS of (41) is obtained as follows. Source currents must comply with Kirchhoff's current law in the coil region:

$$\widetilde{\mathbf{D}}_{\Omega_0} \, \widetilde{\mathbf{j}}_{0,\Omega_0} + \widetilde{\mathbf{D}}_{\Omega_0\Gamma_0} \, \widetilde{\mathbf{j}}_{0,\Gamma_0} = \mathbf{0}, \tag{42}$$

which is obtained (similarly to (36)) by integrating (3) over each dual cell of $\widetilde{\mathcal{G}}_{\Omega_0}$ and using the divergence theorem. Note that boundary currents $\widetilde{\mathbf{j}}_{0,\Gamma_0}$ in (42) are null due to BCs (4). Source currents through the dual faces $\widetilde{f}$ of $\widetilde{\mathcal{G}}_{\Omega_0}$ are obtained from $\mathbf{J}_0$ using piecewise constant edge functions:

$$j_{0,\widetilde{f}} = \int_{\Omega_0} \mathbf{w}_e \cdot \mathbf{J}_0 \, d\Omega. \tag{43}$$

By expanding the source current density in terms of piecewise constant face functions source currents become:

$$j_{0,\widetilde{f}} = \sum_{f \in \mathcal{G}_{\Omega_0}} j_{0,f} \int_{\Omega_0} \mathbf{w}_e \cdot \mathbf{w}_f \, d\Omega, \tag{44}$$

where $j_{0,f}$ are currents thorough the primal faces of $\mathcal{G}_{\Omega_0}$. Equation (44) can be rewritten more compactly as:

$$\widetilde{\mathbf{j}}_{0,\Omega_0} = \mathbf{P}_{\Omega_0} \, \mathbf{j}_{0,\Omega_0}, \tag{45}$$

where projection operator $\mathbf{P}_{\Omega_0} = (p_{ef})_{e,f \in \mathcal{G}_{\Omega_0}}$ is defined as:

$$p_{ef} = \int_{\Omega_0} \mathbf{w}_e \cdot \mathbf{w}_f \, d\Omega. \tag{46}$$

It has to be noted that if source currents on primal faces fulfill (21), then the current density is constant element-by-element and projected currents (44) fulfill (42). The fulfillment of Kirchhoff's current law is important in order to ensure the convergence of Krylov solvers as discussed in [30].

Substituting (39) into (41) and using (38) gives the following system of nonlinear curl-curl equations:

$$\mathbf{C}_\Omega^{\mathrm{T}} \, \mathbf{M}_{\nu(H),\Omega} \, \mathbf{C}_\Omega \, \mathbf{a}_\Omega = \widetilde{\mathbf{j}}_{0,\Omega}, \tag{47}$$

which is solved by imposing null DOFs on boundary edges, $\mathbf{a}_\Gamma = \mathbf{0}$, to comply with the boundary condition (17). It has to be noted that the second term in (41) is dropped due to the enforcement of BCs (i.e., rows and columns of the linear system, corresponding to coefficients of $\mathbf{a}_\Gamma$, are eliminated).

Standard FP method is proven to be unsuitable to solve (47). A more robust variant, the *secant method*, proposed by Bíró in [31], is here adopted. This approach preserves the advantages of the FP method outlined above, e.g., more robust than NR method, and has the remarkable advantage of making use of the B-H curve, directly obtained from the interpolation of measured field data, also in the case of the A–formulation.

The main steps of the secant method to solve nonlinear magnetostatics with the A–method are summarized in Algorithm 2.

At line 3, the initial magnetic reluctivity in any cell of the mesh is set to the minimum reluctivity value, which is obtained from the B-H curve as $\nu_{\min} = \mu_{\max}^{-1}$.

At line 6, the coefficients of the reluctance matrix are calculated with (40) for a given reluctivity value $\nu^{(k)}$, which is estimated for any cell at each iteration. Since $\nu^{(k)} > 0$ the reluctance matrix is positive definite and the related linear system (47) is suitable for the AGMG_CC solver.

From line 7 to 14 the steps are similar to Algorithm 1. At line 15, the curl-curl linear system (47) is iteratively solved using AGMG_CC, starting from $\mathbf{a}_\Omega^{(k)}$ (i.e., last computed solution) as a trial solution to accelerate convergence.

At line 17, the magnetic flux density is calculated at the center of each cell from the array $\mathbf{b}_\Omega^{(k+1)}$, using face basis functions, and then its strength $B^{(k+1)}$ is calculated by taking the Euclidean norm. Note that, using piecewise constant face functions, the interpolated magnetic flux density is constant element-by-element due to Gauss's law $\mathbf{D}_\Omega\, \mathbf{b}_\Omega^{(k+1)} = \mathbf{0}$.

At line 18, for each cell, the magnetic field strength $H^{(k+1)}$ is obtained from $B^{(k+1)}$ using the reluctivity value $\nu^{(k)}$ calculated at the last iteration step.

At line 19, a new value of $B^{(k+1)}$ is calculated from $H^{(k+1)}$ using the B-H curve.

Finally, at line 20, the magnetic reluctivity value $\nu^{(k+1)}$ is calculated for each cell as the ratio of $H^{(k+1)}$ and $B^{(k+1)}$.

Algorithm 2 Secant Method (A–method)

1: $k = 0$
2: $e^{(0)} = 1$
3: $\nu^{(0)} = \nu_{\min}$ (for all mesh elements)
4: $\mathbf{a}_\Omega^{(0)} = \mathbf{0}$
5: **while** $e^{(k)} > \tau \wedge k < k_{\max}$ **do**
6: Assembly of reluctance matrix $\mathbf{M}_{\nu^{(k)},\Omega}$
7: Compute matrix and RHS of (47): $\mathbf{K}_\Omega^{(k)}$, $\mathbf{f}_\Omega^{(k)}$
8: $r^{(k+1)} = \|\mathbf{K}_\Omega^{(k)}\mathbf{a}_\Omega^{(k)} - \mathbf{f}_\Omega^{(k)}\|$
9: **if** $k > 1$ **then**
10: $e^{(k+1)} = r^{(k+1)}/r^{(1)}$
11: **else**
12: $e^{(1)} = 1$
13: **end if**
14: Use $\mathbf{a}_\Omega^{(k)}$ as a trial solution for AGMG_CC
15: Find $\mathbf{a}_\Omega^{(k+1)}$ by solving (47) with AGMG_CC
16: $\mathbf{b}_\Omega^{(k+1)}$ from $\mathbf{a}^{(k+1)}$ using (38)
17: $B^{(k+1)}$ from $\mathbf{b}_\Omega^{(k+1)}$ by interpolation
18: $H^{(k+1)} = \nu^{(k)} B^{(k+1)}$
19: $B^{(k+1)}$ from $H^{(k+1)}$ with B-H curve
20: $\nu^{(k+1)} = H^{(k+1)}/B^{(k+1)}$
21: $\mathbf{a}^{(k)} \leftarrow \mathbf{a}^{(k+1)}$
22: $k \leftarrow k + 1$
23: **end while**

IV. AGGREGATION-BASED MULTIGRID SOLVERS

The discretized formulations in the Section III are posed in a form suitable for aggregation-based algebraic multigrid solvers, with the remarkable feature of linear solution time. In particular, the calculation of the current density during the pre-processing requires the solution of the div-grad system (26) or the curl-curl system (30). The φ–method involves the preliminary calculation of the source field, consisting in the solution of the curl-curl linear system (33), and the calculation of nodal variables (i.e., magnetic scalar potentials) by solving the div-grad linear system (37). Finally, the A–method involves only the calculation of edge variables (i.e., line integrals of the magnetic vector potential), consisting in the solution of the curl-curl linear system (47).

A. AGMG PRECONDITIONER

Div-grad linear systems have the following abstract form:

$$\mathbf{G}^{\mathrm{T}}\mathbf{M}\mathbf{G}\,\mathbf{u} = \mathbf{f}, \tag{48}$$

where $\mathbf{G}$ is a discrete gradient operator (i.e., edges-to-nodes incidence matrix), $\mathbf{M}$ is a positive definite mass matrix, $\mathbf{u}$ is the unknown vector and $\mathbf{f}$ is any admissible RHS that lies in the image of $\mathbf{G}^{\mathrm{T}}$. Equation (48) is singular, with a kernel dimension of $\mathbf{G}$ equal to one for a topologically-connected graph. By putting $\mathbf{G} = \widehat{\mathbf{D}}_{\Omega_0}^{\mathrm{T}}$ and $\mathbf{M}$ equal to the identity matrix with size the number of dual edges of $\widetilde{\mathcal{G}}_{\Omega_0}$, it can be easily observed that (26) is a div-grad system. Likewise, putting $\mathbf{G} = \mathbf{G}_\Omega$ and $\mathbf{M} = \mathbf{M}_{\mu(H),\Omega}$, it can be noted that the linear system of the φ–method (37) reduces to (48).

It is well known in literature that the linear systems of the form (48) can be solved in linear time using algebraic multigrid methods. The AMG preconditioner (or solver) automatically builds a hierarchy of progressively smaller linear systems, and recursively uses the approximate correction from the next coarse system in combination with the smoothing for a current coarse system to produce an approximate solution at the current level. A state-of-the-art implementation of the AMG preconditioner is BoomerAMG present in the HYPRE toolbox. However, this is not always robust and efficient when singular systems such as (48) are to be solved (see Section V).

To address this problem, a different AMG algorithm is here considered: the aggregation-based algebraic multigrid, implemented in the AGMG software [32]. Details of the algorithm can be found in Notay's work [10], and are not discussed here for the sake of brevity. In summary, the preconditioner from this software builds the next system in the multigrid hierarchy from the preceding one by grouping the equations/unknowns into aggregates. Like other AMG preconditioners, the construction of the hierarchy is purely algebraic (or black box), and AGMG only takes as input the system to be solved. The AGMG preconditioner is used here in combination with a Krylov solver such as the flexible conjugate gradient proposed in [33]. As shown in Section V, the choice of AGMG is motivated by the fact that the corresponding preconditioner is more efficient than the state-of-the-art BoomerAMG preconditioner, used here with PCG, and always presents linear solution time in all numerical benchmarks considered.

There are several features that make AGMG particularly computationally efficient. Building the multigrid hierarchy by aggregation of equations/unknowns makes the setup stage of AGMG simpler and faster than that of classical AMG such as BoomerAMG, whereas the resulting multigrid hierarchy is typically sparser and needs less memory. The simplicity of the aggregation approach has its price: it requires a more elaborated traversal of the hierarchy. However, if well chosen (e.g., as K-cycle in AGMG [34]), such traversal has limited impact on the solution time and can enhance the robustness of AGMG. The latter is further strengthened by a rather unique quality control mechanism for selected aggregates [35], [36].

B. AGMG_CC PRECONDITIONER

Curl-curl linear systems have the following abstract form:

$$\mathbf{C}^{\mathrm{T}}\mathbf{M}\mathbf{C}\,\mathbf{u} = \mathbf{f}, \tag{49}$$

where $\mathbf{C}$ is the discrete curl operator (i.e., faces-to-edges incidence matrix), $\mathbf{M}$ is a positive definite mass matrix, $\mathbf{u}$ is the unknown vector and $\mathbf{f}$ is any admissible RHS that lies in the image of $\mathbf{C}^{\mathrm{T}}$. It is known in the literature that linear systems of the form (49) are more difficult to solve than (48) due to the larger kernel size of $\mathbf{C}$ compared to $\mathbf{G}$ [37].

It can be noted that the linear system (30) for the source current calculation reduces to (49) by putting $\mathbf{C} = \mathbf{C}_{\Omega_0}$ and $\mathbf{M} = \mathbf{M}_{\rho,\Omega_0}$. The linear system for the source field calculation (33) reduces to (49) by putting $\mathbf{C} = \mathbf{C}_{\Omega}$ and $\mathbf{M}$ equal to the identity matrix with dimension the number of primal edges of $\mathcal{G}_{\Omega}$. Similarly, with the A–method, (47) reduces to (49) by putting $\mathbf{C} = \mathbf{C}_{\Omega}$ and $\mathbf{M} = \mathbf{M}_{\nu(H),\Omega}$.

It is known in the literature that auxiliary space preconditioners (or solvers) exhibit a linear solution time when applied to the solution of curl-curl linear systems [18], [38]. These preconditioners perform successive applications of smoothing and subspace correction steps, each requiring an approximate solution of a linear system. In the subspace correction step the linear systems are obtained by projecting the associated curl-curl linear system onto the images of interpolating linear operators Π_x, Π_y, Π_z. For instance, Π_x transforms the x components of the nodal values of a continuous vector field, assumed linear in each tetrahedron, onto the line integrals along the edges of the same vector field. It is defined as:

$$\Pi_x = \frac{1}{2}\mathbf{L}_x|\mathbf{G}|, \tag{50}$$

where $\mathbf{L}_x$ is a diagonal matrix whose diagonal coefficients are the x-axis components of the edge vectors of the edges of the mesh, and $|\mathbf{G}|$ is the matrix obtained from the discrete gradient operator by taking the absolute value of each entry. Similar definitions hold for Π_y and Π_z. The projected systems arising in the correction step are known to be suitable for approximation by an AMG preconditioner.

The state-of-the-art implementation of the auxiliary space preconditioner is the AMS preconditioner from the HYPRE toolbox, which uses BoomerAMG to approximately solve the projected systems. However, as shown in Section V, this preconditioner occasionally lacks of robustness. This observation motivated one of the authors to develop a variant of the AS preconditioner that is based on the AGMG software; it is freely distributed with the AGMG_CC software [39].

The AGMG_CC software takes as input the coefficient matrix and the RHS of the linear system, the discrete gradient operator, and the coordinates of the mesh nodes. Once the auxiliary space preconditioner is formed, it can be used in combination with the flexible PCG to solve the curl-curl systems of the form (49). The application of the AGMG_CC preconditioner to a vector $\mathbf{r}$ is described in Algorithm 3. In this algorithm, $B_{AGMG}(\mathbf{M}, \mathbf{v})$ represents a single application to a vector $\mathbf{v}$ of the AGMG preconditioner for a given matrix $\mathbf{M}$, $\mathbf{K}$ indicates the system matrix in the discrete curl-curl equation, and $\mathbf{L}_{\mathbf{K}}$, $\mathbf{U}_{\mathbf{K}}$ are the lower and the upper triangular parts of $\mathbf{K}$, respectively. In particular, the smoothing steps amount to one forward Gauss-Seidel pre-smoothing step (lines 1-2) and one backward Gauss-Seidel post-smoothing step (line 7). Regarding the subspace corrections (lines 3-6), they are computed concurrently for each projected system.

Algorithm 3 Steps of the AGMG_CC Preconditioner

Pre-smoothing
1: $\mathbf{e}_1 = \mathbf{L}_{\mathbf{K}}^{-1}\,\mathbf{r}$
2: $\mathbf{r} = \mathbf{r} - \mathbf{K}\,\mathbf{e}_1$
Correction
3: $\mathbf{e}_2 = \Pi_x\, B_{AGMG}\left(\Pi_x^{\mathrm{T}}\mathbf{K}\,\Pi_x,\ \Pi_x^{\mathrm{T}}\mathbf{r}\right)$
4: $\mathbf{e}_2 = \mathbf{e}_2 + \Pi_y\, B_{AGMG}\left(\Pi_y^{\mathrm{T}}\mathbf{K}\,\Pi_y,\ \Pi_y^{\mathrm{T}}\mathbf{r}\right)$
5: $\mathbf{e}_2 = \mathbf{e}_2 + \Pi_z\, B_{AGMG}\left(\Pi_z^{\mathrm{T}}\mathbf{K}\,\Pi_z,\ \Pi_z^{\mathrm{T}}\mathbf{r}\right)$
6: $\mathbf{r} = \mathbf{r} - \mathbf{K}\,\mathbf{e}_2$
Post-smoothing
7: $\mathbf{e}_3 = \mathbf{U}_{\mathbf{K}}^{-1}\,\mathbf{r}$
8: $\mathbf{e} = \mathbf{e}_1 + \mathbf{e}_2 + \mathbf{e}_3$

It is shown in the next section that AGMG and AGMG_CC can be exploited to solve magnetostatics in linear time.

V. NUMERICAL RESULTS

Both φ–method and A–method presented in Section III, together with the fixed-point iteration method (Algorithm 1) and the secant method (Algorithm 2), were implemented in vectorized MATLAB® codes. Algebraic multigrid solvers were implemented in C and FORTRAN programming languages and compiled using GCC and GFORTRAN compilers with optimization options.

Both φ–method and A–method were validated by means of the same benchmarks: 1) an axisymmetric model (the clapper solenoid actuator) for which an accurate reference third-order 2D FEM solution could be computed, 2) a fully 3D model (the toroidal inductor) with highly complex geometry. These benchmarks were chosen to test the robustness of algebraic multigrid when solving linear systems

stemming from the discretization of magnetostatic problems of practical interest, encompassing a large number of DOFs, inhomogeneous media, nonlinearity, and parts with complex geometry. To the authors' knowledge, such magnetostatic problems are not commonly considered in the literature on algebraic multigrid. It is shown below that both AGMG and AGMG_CC outperform state-of-the-art algebraic multigrid solvers such as Boomer AMG and AMS from the HYPRE library [13]. In particular, AGMG_CC shows to be more robust than AMS.

All numerical tests used for validation of the CM codes implementing the φ–method and the A–method were run on a laptop equipped with an Intel Core i7-6920HQ processor (2.9 GHz clock frequency) and 16 GB random access memory (RAM). Convergence tests for assessing the performance of multigrid solvers were carried out using a desktop with an Intel Xeon E5-2620 processor at 2.10 GHz and 128 GB RAM.

A. CLAPPER SOLENOID ACTUATOR

Fig. 1 shows a sketch of the clapper solenoid actuator with axial symmetry (it is the cross-section on the plane $\theta = 0$ in a cylindrical coordinate system r, θ, z). This model (derived from the clapper solenoid actuator described in [40], p. 101) was analyzed either with the CM or the FEM used for comparisons. The clapper solenoid model (computational domain Ω) consists of a magnetic core (stator Ω_1 and armature Ω_2), excited by a DC current-driven coil Ω_0, and the air domain Ω_3, delimited by a bounding box (48 × 63 mm) to which magnetic wall BCs are applied. The stator (30 mm radius, 23 mm height) and the armature (same radius, 8 mm height) are made of nonlinear and isotropic magnetic material (i.e., the soft-iron of the COMSOL Multiphysics® material library).

The magnetization curve in Fig. 2 of the soft-iron shows a strong nonlinearity and enters the saturation region with a magnetic field strength of 0.5 A/m. In the CM codes scattered field data of the curve were interpolated using a piecewise cubic hermite polynomial (PCHIP function in MATLAB®). This provided a smooth representation of the constitutive law (6) that has proven critical to the convergence of nonlinear solvers. The coil of $N = 2000$ turns ($S = 10 \times 15$ mm cross-section area) carries a current of $I_0 = 1$ A per turn and is modeled by a constant source density $\mathbf{J}_0 = N\, I_0/S\, \mathbf{e}_\theta$, where $\mathbf{e}_\theta$ is the azimuthal unit vector.

The clapper solenoid model in Fig. 1 was first simulated in the COMSOL Multiphysics® software using axisymmetric (third-order) FEM. The triangle mesh with 5046 elements, discretizing Ω, was refined up to convergence in order to obtain highly accurate field values, taken as a reference solution. The nonlinear system (with 22 978 DOFs) was solved using the default nonlinear solver of COMSOL—based on the NR method—, with a relative tolerance of 10^{-12}. A PCG solver with successive over-relaxation (SOR) preconditioner was used for inner iterations, as default option in COMSOL.

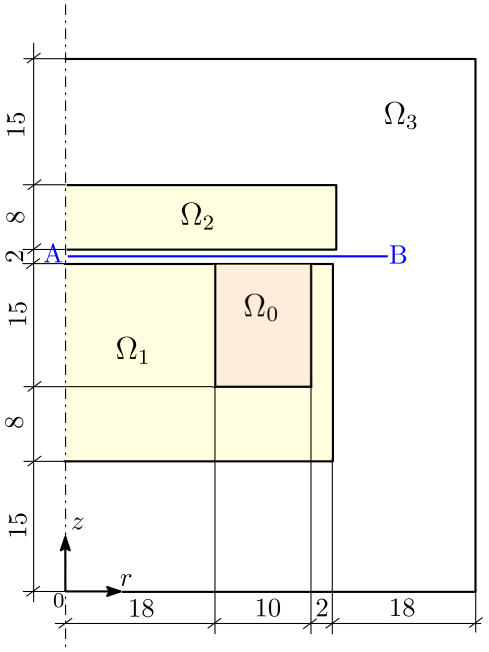

FIGURE 1. Axisymmetric model of the clapper solenoid: coil Ω_0, stator Ω_1, armature Ω_2, air domain Ω_3. The magnetic flux density is calculated along the line A–B (blue).

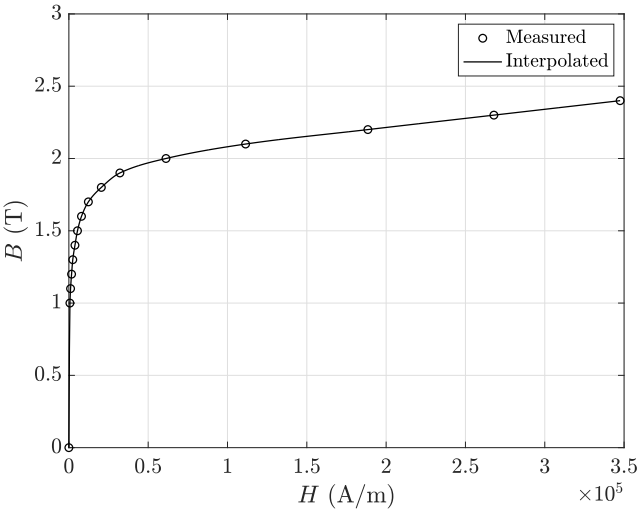

FIGURE 2. B-H curve of soft iron used in simulations: measured data are taken from the COMSOL Multiphysics® material library; scattered data are interpolated in MATLAB® using a piecewise cubic hermite polynomial.

Due to symmetry only one six of the geometry was considered in either 3D CM or 3D (first-order) FEM models. Fig. 3 shows the computational domain meshed by the coarsest tetrahedral mesh used in the numerical tests. For 3D CM and 3D FEM the same tetrahedral mesh with 430 248 nodes, 2 976 044 edges, 506 9470 faces, and 2 523 673 volumes was used. With the 3D model, together with magnetic wall BCs, symmetry BCs were applied to the vertical parts of the boundary (planes $\theta = 0$ and $\theta = 60°$ in cylindrical coordinates).

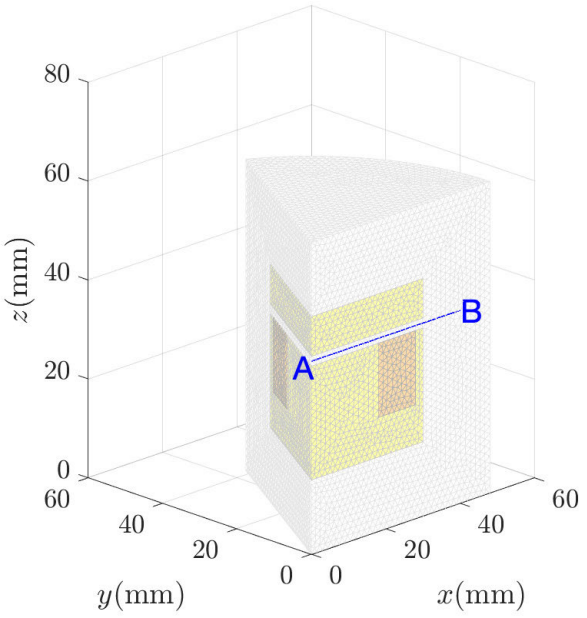

FIGURE 3. 3D model of the clapper solenoid (only 1/6 due to symmetry) discretized with the coarsest mesh used in simulations. Field computation line A–B is on the plane $y = 0$, corresponding to $\theta = 0$ in cylindrical coordinates.

The default nonlinear solver of COMSOL—based on the NR method—was used also for 3D FEM, with a relative tolerance of $\tau = 10^{-3}$. This value was chosen in order to get the model convergence to the reference solution obtained with axisymmetric FEM. In COMSOL, for each NR iteration, the FEM linear system—with a number of unknowns equal to the number of edges—was solved using PCG with geometric multigrid preconditioner, with 10^{-4} stopping tolerance for the relative residual (i.e., lower than nonlinear solver tolerance). As a general criterion, relative tolerances for the linear system solvers were typically chosen an order of magnitude smaller than the tolerance for the nonlinear solver. This ensures that the number of iterations of the nonlinear solver is essentially independent of the linear solver being used. On the other hand, by choosing larger tolerance for linear solvers the number of iterations of the nonlinear solver can increase, and the nonlinear solver can even fail to converge. Note that geometric multigrid could be efficiently used in this case due to the rather simple geometry of the clapper solenoid model. The NR solver converged to the set tolerance in 8 iterations, with central processing unit (CPU) time of 1261 s.

The pre-processing stage in the φ–method also included the construction of the source field. The linear system (33)—with a number of unknowns equal to the number of edges—was solved by AGMG_CC. The latter converged in 6 iterations to a stopping tolerance for the relative residual of 10^{-5}, with a CPU time of 31.24 s. To obtain the convergence of fixed-point iterations (Algorithm 1) to the reference solution, a lower relative tolerance than that of the NR method was adopted ($\tau = 10^{-4}$). The FP method converged in 20 iterations, with a CPU time of 40.15 s. For each FP iteration the final system of the φ–method (37)—with a number of unknowns equal to the number of nodes—was solved by AGMG, with a stopping tolerance for the relative residual of 10^{-5} (i.e., this value for the linear solver one was chosen order of magnitude lower than the nonlinear solver tolerance using the above criterion).

The A–method, unlike the φ–method, did not require the source field computation. The secant method (Algorithm 2) converged to the reference solution in 30 iterations, with a CPU time of 840.63 s. To get the convergence to the reference solution a relative tolerance of 10^{-3} was chosen. For each secant method iteration the linear system (47)—with a number of unknowns equal to the number of edges—was solved using AGMG_CC, with a stopping tolerance for the relative residual of 10^{-4}. Also in this case, the tolerance was chosen an order of magnitude lower with the usual criterion.

The z-axis component of the magnetic flux density B_z was calculated at 1001 equally spaced points along the line A-B, shown in Fig. 3 ($r = [0, 40]$ mm, $\theta = 0$, $z = 39$ mm) using both φ–method and A–method. These field distributions were compared with those of 3D FEM and axisymmetric FEM, the latter taken as the reference solution. Fig. 4 shows the B_z profile calculated with the φ–method, with a maximum discrepancy from the reference solution of 5.29%. A similar discrepancy (4,17%) was obtained by comparison with 3D FEM. Fig. 5 shows the B_z profile calculated by the A–method, with a maximum discrepancy from the reference solution of 3.76%. The A–method solution also agrees very well with that of the 3D FEM (0.45% maximum discrepancy).

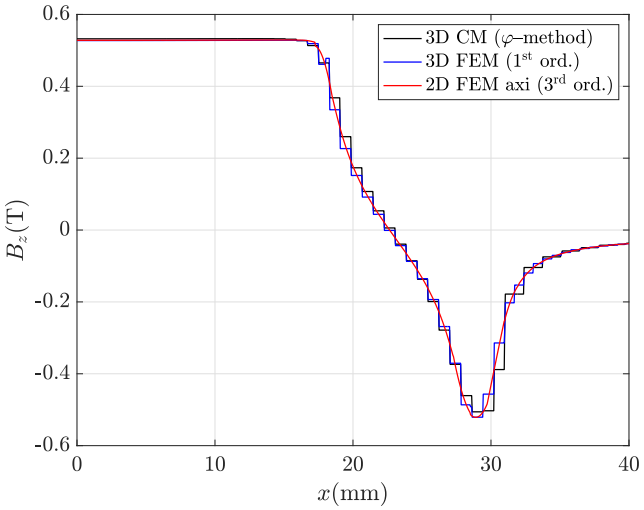

FIGURE 4. Vertical component of the magnetic flux density B_z calculated along the line A-B with 3D CM (φ–method), 3D FEM (1st ord.) and axisymmetric FEM (3rd ord).

To evaluate the computational performance and robustness of both AGMG and AGMG_CC, the 3D model of the clapper solenoid was solved with the φ–method (based on both AGMG and AGMG_CC solvers) or the A–method (based on AGMG_CC solver only) by considering tetrahedral

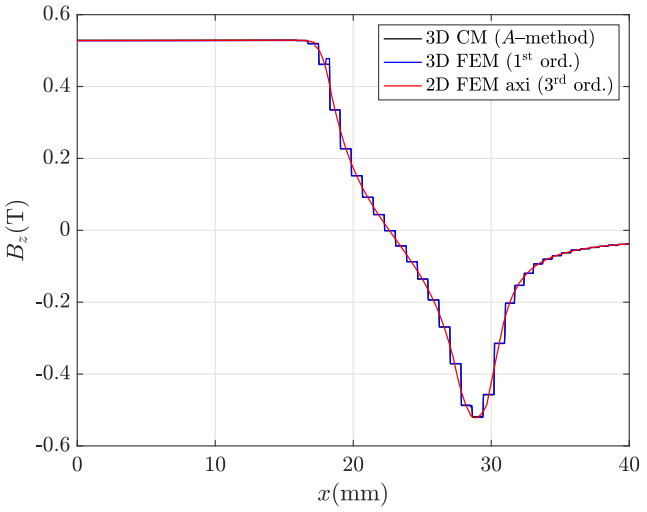

FIGURE 5. Vertical component of the magnetic flux density B_z calculated along the line A-B with 3D CM (*A*–method), 3D FEM (1st ord.) and axisymmetric FEM (3rd ord).

meshes of decreasing mesh size. The DOFs required by the φ–method—when solving the curl-curl system (33) and the div–grad system (37)—and by the *A*–method—when solving the curl-curl system (47)—are reported in Table 1, for each mesh size h. These tests were useful to verify whether linear solution time is achieved by both AGMG and AGMG_CC when dealing with magnetostatic problems of practical interest.

TABLE 1. Number of DOFs of φ–method and *A*–method for different discretizations of the clapper solenoid model.

		φ–method		*A*–method
h (mm)	# tets	source field	solution	solution
1.5	374 881	446 953	65 580	427 474
1.0	1 291 392	1 527 190	221 593	1 484 572
0.8	2 523 673	2 976 044	430 248	2 909 672
0.7	3 765 861	4 433 969	639 413	4 347 881
0.6	5 984 695	7 036 346	1 012 963	6 920 279

The solution time of AGMG was compared to that of PCG+AMG (i.e., conjugate gradient solver with BoomerAMG preconditioner). For completeness, PCG+SOR, usually adopted in FEM software for EM analysis, was also considered. Similarly, the solution time of AGMG_CC was compared to that of PCG+AMS (i.e., conjugate gradient solver with AMS preconditioner). Also in this case, PCG+SOR was considered as the reference resolution strategy.

In these complexity analyses the same stopping tolerance for the relative residual of 10^{-4} was adopted for all solvers to have a fair comparison. The above value of the stopping tolerance also ensured that the number of nonlinear solver iterations was practically independent of the type of linear system solver: they changed by at most one iteration when changing the linear system solver.

Fig. 6 shows the CPU time required by AGMG_CC, PCG+AMS, and PCG+SOR solvers when solving (33) to calculate the source field for different discretization of the clapper solenoid model (Table 1, φ–method). It can be observed that the solution time profiles of both multigrid solvers are $\mathcal{O}(N)$, i.e., essentially linear, with respect to the number of unknowns N (dashed line in Fig. 6), whereas PCG+SOR has slightly superlinear solution time, since its number of iterations increases with N. Furthermore, AGMG_CC needs half the CPU time of PCG+AMS, illustrating its efficiency.

A more precise check of the $\mathcal{O}(N)$ complexity can be realized by comparing the number of iterations for different solvers when calculating the source field. In Table 2 it can be noted that the number of iterations is almost independent of the number of DOFs only in the case of AGMG_CC solver. In addition, PCG+SOR exhibits superlinear behavior.

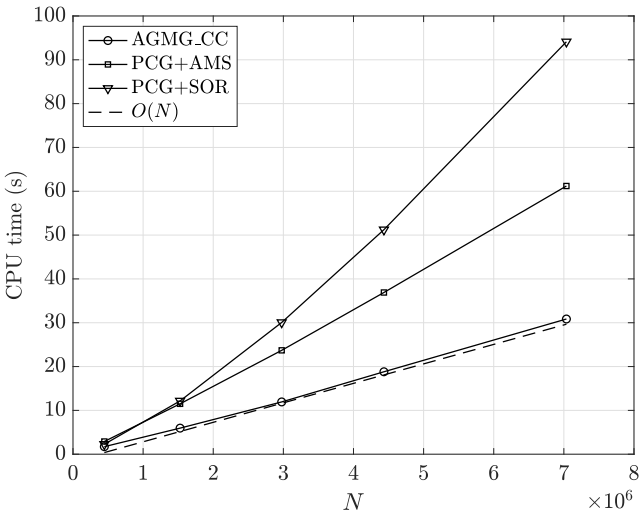

FIGURE 6. CPU time (s) vs. number of DOFs for different discretizations of the clapper solenoid model (solution of curl-curl system (33) to calculate the source field).

TABLE 2. Number of iterations to calculate the source field for different discretizations of the clapper solenoid model.

h (mm)	# DOFs	AGMG_CC	PCG+AMS	PCG+SOR
1.5	446 953	11	13	84
1.0	1 527 190	11	17	126
0.8	2 976 044	11	18	156
0.7	4 433 969	12	19	178
0.6	7 036 346	12	20	207

Fig. 7 shows the overall CPU time required to solve the final system of the φ–method (37) by fixed point iterations

(Algorithm 1) when using different Krylov solvers for the solution of the linear system (line 15, Algorithm 1), namely AGMG, BoomerAMG or PCG+SOR. It can be clearly observed that AGMG attains $\mathcal{O}(N)$ time and outperforms both BoomerAMG and PCG+SOR solvers. This confirms the observations made in Section IV on the performance of AGMG. Also note that since linear system (37) is singular, a regularized version was used for BoomerAMG solution.

This behavior can be verified more precisely by checking the total number of iterations (i.e., the sum of the number of Krylov solver iterations over all fixed-point iterations) required for the solution of (37). The number of FP iterations is constant (equal to 20), as expected, because the same tolerance (small enough) was chosen for all solvers. Table 3 shows that both AGMG and BoomerAMG maintain an almost constant number of iterations when reducing the mesh size, whereas it increases for PCG+SOR.

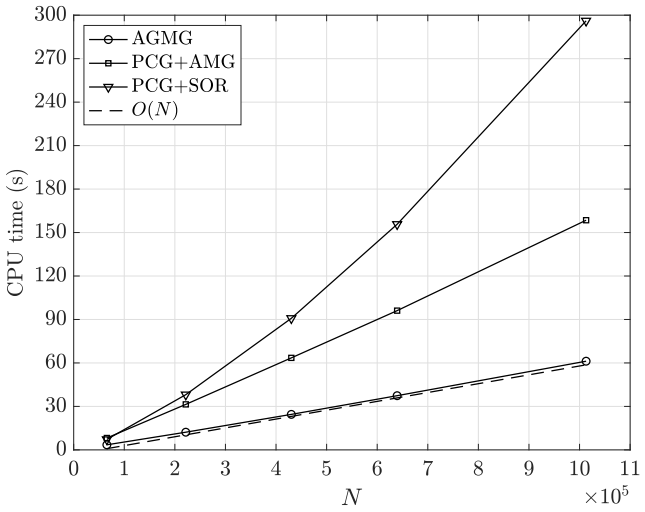

FIGURE 7. CPU time (s) vs. number of DOFs for different discretizations of the clapper solenoid model (φ –method).

TABLE 3. Number of iterations to solve (37) by the φ-method for different discretizations of the clapper solenoid model.

h (mm)	# DOFs	AGMG	BoomerAMG	PCG+SOR
1.5	65 580	110	88	1611
1.0	221 593	116	96	2405
0.8	430 248	113	99	2940
0.7	4 433 969	115	101	3355
0.6	7 036 346	119	109	3975

Similar considerations apply to a greater extent to the solution of the curl-curl system (47) arising in the A–method. The CPU time for the cases listed in Table 1 (A–method) is shown in Fig. 8, where AGMG_CC attains $\mathcal{O}(N)$ time, whereas computational performance of both PCG+AMS and PCG+SOR downgrades as the problem size increases up to almost seven million DOFs. In particular, the number of PCG+AMS iterations increases with N. This different behavior compared to results in Fig. 6 may be due to a non-identity mass matrix in the curl-curl system.

Table 4 shows the total number of iterations for the above solvers when calculating (47). It can be clearly observed that the number of iterations for AGMG_CC is almost independent of the number of DOFs, therefore $\mathcal{O}(N)$ time is achieved, whereas it is more than doubled for both PCG+AMS and PCG+SOR when reducing the mesh size.

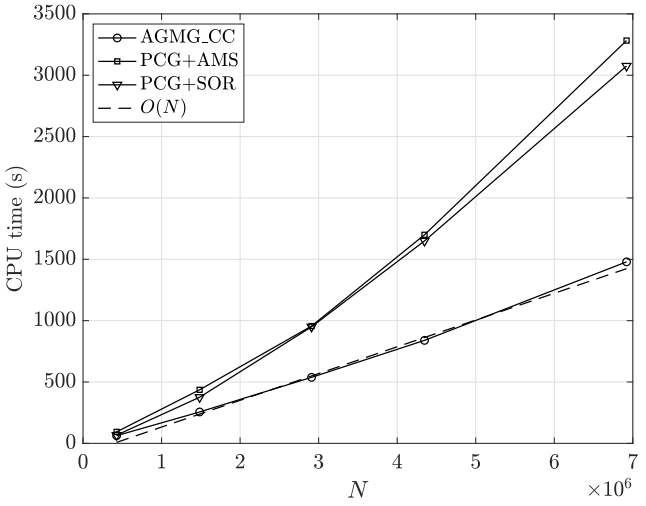

FIGURE 8. CPU time (s) vs. number of DOFs for different discretizations of the clapper solenoid model (A–method).

TABLE 4. Number of iterations to solve (47) by the A-method for different discretizations of the clapper solenoid model.

h (mm)	# DOFs	AGMG_CC	PCG+AMS	PCG+SOR
1.5	427 474	470	451	2104
1.0	1 484 572	562	657	3293
0.8	2 909 672	597	752	4168
0.7	4 347 881	628	931	4830
0.6	6 920 279	647	1102	5662

B. TOROIDAL INDUCTOR

To test the robustness of the proposed solution approaches, based on both AGMG and AGMG_CC, a fully 3D magnetostatic problem with complex geometry was considered. The toroidal inductor model in Fig. 9 consists of a spiral–shaped coil (1 mm wire radius, 40 mm knot major radius, 12.5 mm knot minor radius) wrapped around a toroidal core (40 mm major radius, 10.5 mm minor radius), which is made of a linear magnetic material with $\mu_r = 10$ relative permeability.

These data, which are close to those of a real inductor core (Micrometals T400-2 core), are taken from [41]. The core is centered at the origin of the Cartesian coordinate system x, y, z. The coil is excited by a current density of 1 MA/m^2, which is assumed to be constant over the wire cross-section. The whole model is embedded into a cube of 200 mm side length, which is centered at the origin and delimits the air domain. Magnetic wall BCs are applied to the cube boundary.

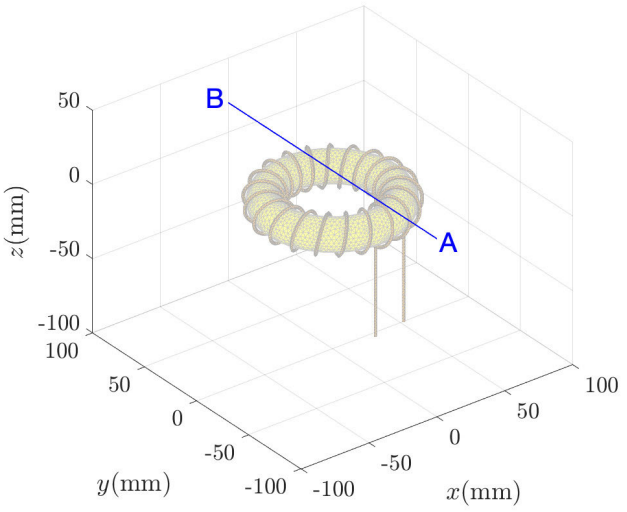

FIGURE 9. 3D model of the toroidal inductor with the coarsest discretization. Field computation line A–B is on the plane $x = 0$ at $z = 20$ mm.

The toroidal inductor model was first simulated in COMSOL using 3D FEM with first-order elements. The tetrahedral mesh was refined until convergence of the field profile calculated along the horizontal line A–B shown in Fig. 9, from point A $(0, -100, 20)$ to point B $(0, 100, 20)$. This mesh, also used for the CM codes, consisted of 113 691 nodes, 786 822 edges, 1 343 031 faces, and 669 899 cells.

The FEM final system with 786 823 unknowns was solved in COMSOL by the PCG+AMS. This solver converged in 14 iterations to the stopping tolerance for the relative residual of 10^{-6}, with a CPU time of 26 s. This tolerance value was chosen by observing that with smaller values the field profile along the A–B line did not vary significantly. It should be noted that, unlike the clapper solenoid benchmark, in this case it was not possible to use in COMSOL the PCG solver with geometric multigrid preconditioner due to the highly complex geometry of the toroidal inductor model.

In the CM code implementing the φ–method the pre-processing stage encompassed the calculation of the source field, not required by the A–method. The curl-curl system (33)—with a number of unknowns equal to the number of edges—was solved by AGMG_CC, which converged in 7 iterations to the stopping tolerance for the relative residual of 10^{-6}, with a CPU time of 4.47 s. After pre-processing, the final system of the φ–method (37) with 113 691 unknowns, was solved by AGMG in 0.57 s. The latter converged to the stopping tolerance for the relative residual of 10^{-6} in 9 iterations.

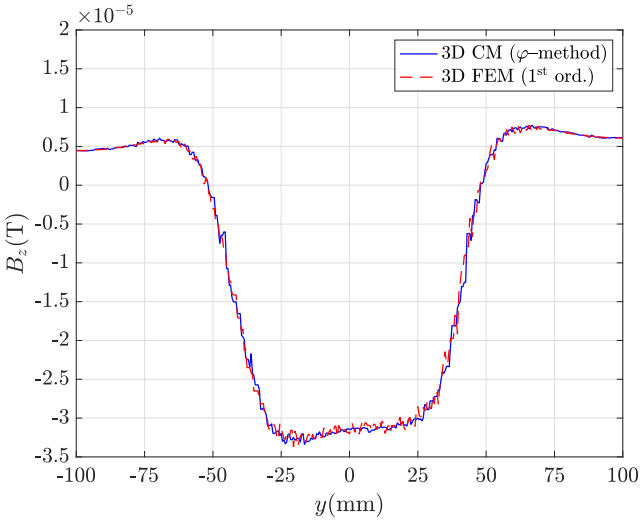

FIGURE 10. Vertical component of the magnetic flux density B_z calculated along the line A-B with 3D CM (φ–method) and 3D FEM (1st ord.) taken as a reference.

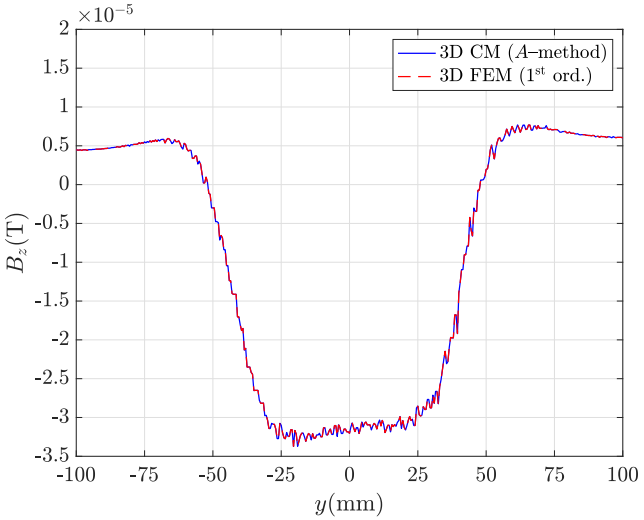

FIGURE 11. Vertical component of the magnetic flux density B_z calculated along the line A-B with 3D CM (A–method) and 3D FEM (1st ord.) taken as a reference.

With the A–method, not requiring the source field calculation, the final system (47) with 777 123 unknowns was solved by AGMG_CC in 7.50 s, converging in 10 iterations to the same tolerance as above. It is interesting to observe that the overall processing time for the φ–method (encompassing both pre-processing and solution time) was 5.04 s, comparable to that of the A–method.

The z-axis component of the magnetic flux density distribution B_z was calculated either using 3D CM or 3D FEM (reference solution) along the line A–B, above the toroidal inductor. Fig. 10 shows the field distribution calculated using the φ–method, for which the maximum discrepancy from

the FEM solution is 3.98%. Fig. 11 shows that the A–method provides almost the same solution as 3D FEM (0.01% discrepancy).

To benchmark the computational performance and robustness of both AGMG and AGMG_CC, the toroidal inductor model was solved for different CM discretizations. Numerical tests similar to the clapper solenoid model were performed. Table 5 shows the DOFs required by the φ–method, to solve the curl-curl system (33) and the div-grad system (37), and by the A–method, to solve the curl-curl system (47).

TABLE 5. Number of DOFs of φ–method and A–method for different discretizations of the toroidal inductor model.

		φ–method		A–method
h (mm)	# tets	source field	solution	solution
7.0	513 123	607 639	88 456	589 456
5.0	1 188 157	1 402 843	203 231	1 368 475
4.0	2 226 142	2 623 047	378 802	2 568 735
3.5	3 289 942	3 871 416	557 994	3 800 973
3.0	5 194 538	6 104 208	877 719	6 008 352

The solution time of AGMG_CC for the source field calculation was compared to that of AMS of the HYPRE library and to PCG+SOR, for the CM discretizations reported in Table 5. For all solvers the same stopping tolerance for the relative residual of 10^{-6} was used to have a fair comparison.

Fig. 12 shows that AGMG_CC exhibits $\mathcal{O}(N)$ time also in this benchmark. For the largest problem, AGMG_CC requires roughly one half of the CPU time of PCG+AMS, and roughly one third of the CPU time of PCG+SOR. Table 6 shows that the number of iterations for AGMG_CC is constant when reducing the mesh size, so $O(N)$ time is achieved, whereas it increases for both PCG+AMS and PCG+SOR.

As in the clapper solenoid benchmark, two different solution strategies to solve magnetostatics were compared. For the φ–method, the solution time of AGMG was compared to that of BoomerAMG and PCG+SOR, suitable for div–grad systems such as (37). For the A–method, the solution time of AGMG_CC was compared to that of AMS and PCG+SOR, suitable for curl-curl systems such as (47). As in the preprocessing stage, the same stopping tolerance for the relative residual of 10^{-6} was used for all solvers.

Fig. 13 shows the CPU time of AGMG, BoomerAMG, and PCG+SOR solvers for each discretization in Table 5 (φ–method). It can be noted that AGMG, as expected from Section IV, outperforms both BoomerAMG and PCG+SOR. In particular, AGMG is about twice as fast, although both AGMG and BoomerAMG show a linear solution time. A more detailed analysis, based on the number of iterations, confirms the latter result: in Table 7 it can be noted that both AGMG and BoomerAMG have an almost constant number of iterations, resulting thus in a linear solution time.

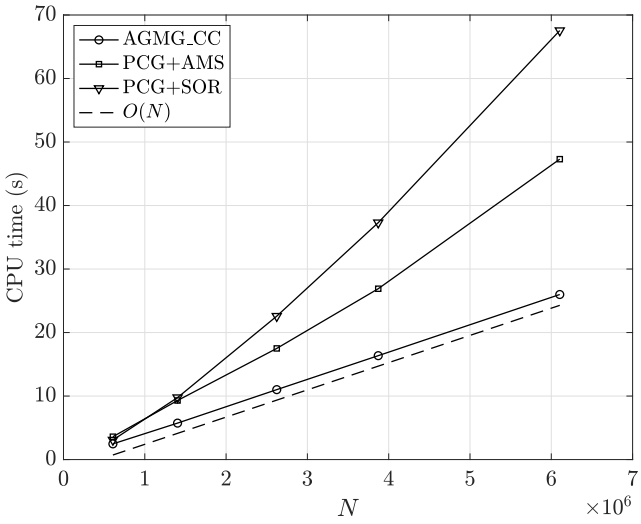

FIGURE 12. CPU time (s) vs. number of DOFs for different discretizations of the toroidal inductor model (solution of curl-curl system (33) to compute the source field).

TABLE 6. Number of iterations to calculate the source field for different discretizations of the toroidal inductor model.

h (mm)	# DOFs	AGMG_CC	PCG+AMS	PCG+SOR
7.0	607 639	12	11	81
5.0	1 402 843	12	13	109
4.0	2 623 047	12	13	132
3.5	3 871 416	12	14	149
3.0	6 104 208	12	16	171

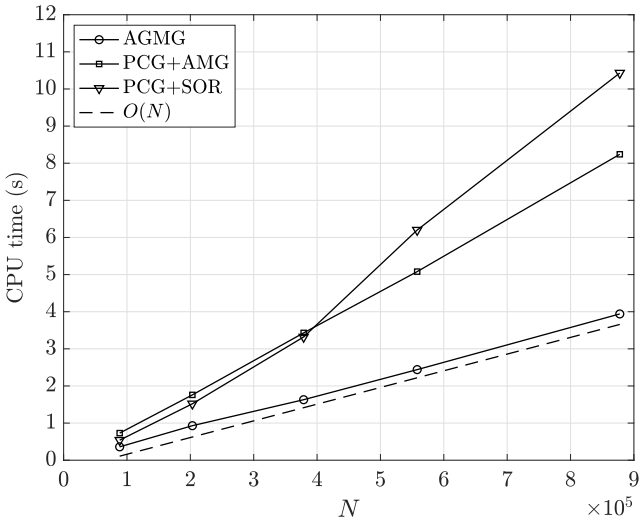

FIGURE 13. CPU time (s) vs. number of DOFs for different discretizations of the toroidal inductor model (φ–method).

Fig. 14 shows the peak memory usage (PMU) required by AGMG for each discretization in Table 5. PMU estimates

TABLE 7. Number of iterations to solve (37) by the φ-method for different discretizations of the toroidal inductor model.

h (mm)	# DOFs	AGMG	BoomerAMG	PCG+SOR
7.0	88 456	9	9	91
5.0	203 231	11	9	107
4.0	378 802	11	10	121
3.5	557 994	11	10	154
3.0	877 719	11	10	163

consider the memory consumed by the solver during the different stages (setup/solve) and, as a reference, the memory needed to store the input data. It should be noted that both setup and solve memories include the memory for input data. During the solution stage the memory consumption encompassed both the construction of the multigrid preconditioner and the Krylov acceleration of flexible PCG. For all solver steps the PMU has a linear increase with N and is approximately three times the PMU required by the input data.

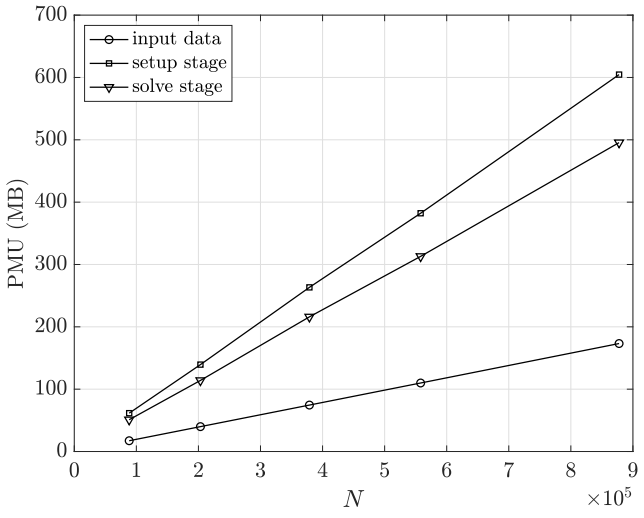

FIGURE 14. PMU (MB) vs. number of DOFs for different discretizations of the toroidal inductor model (φ–method).

The numerical results for the A–method, obtained with the same meshes as the φ–method, are given in Fig. 15. It can be observed that AGMG_CC here as well attains $\mathcal{O}(N)$ time to reach the stopping tolerance for the relative residual of 10^{-6}, whereas convergence deteriorates rapidly for PCG+AMS. In particular, PCG+AMS does not converge for the largest linear system, encompassing six million unknowns, and the CPU time of PCG+SOR is more than four times that of AGMG_CC. As noted for the clapper solenoid benchmark, this different behavior compared to results in Fig. 12 may be due to a non-identity mass matrix in the curl-curl system.

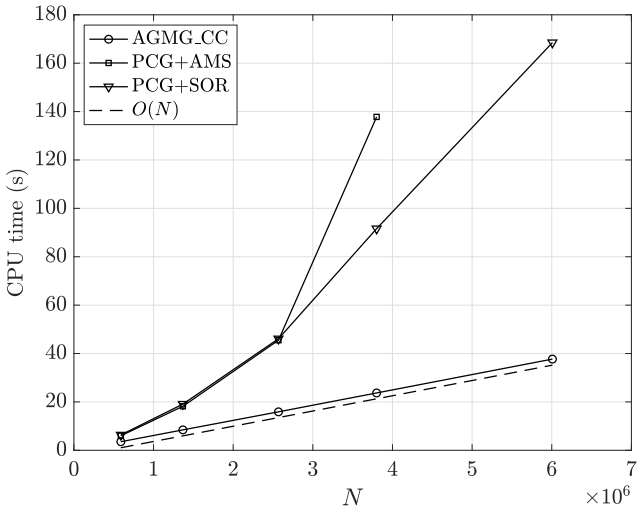

FIGURE 15. CPU time (s) vs. number of DOFs for different discretizations of the toroidal inductor model (A–method).

TABLE 8. Number of iterations to solve (47) by the A-method for different discretizations of the toroidal inductor model.

h (mm)	# DOFs	AGMG_CC	PCG+AMS	PCG+SOR
7.0	589 456	20	24	144
5.0	1 368 475	20	35	183
4.0	2 568 735	20	52	215
3.5	3 800 973	20	111	247
3.0	6 008 352	20	—	286

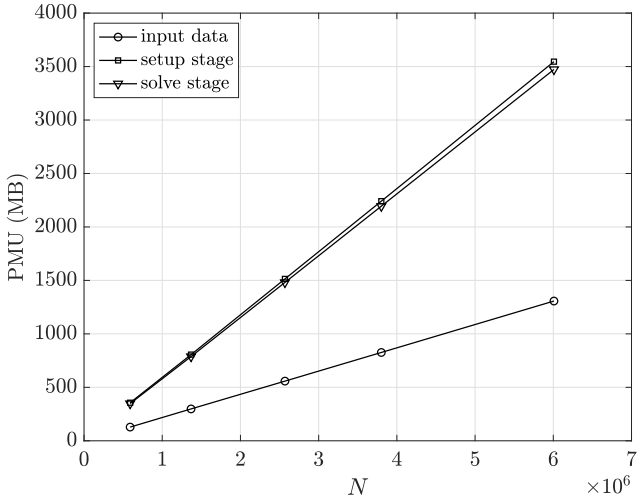

FIGURE 16. PMU (MB) vs. number of DOFs for different discretizations of the toroidal inductor model (A–method).

Table 8 shows the iteration count to solve the above problems with the A-method. It can be noted that AGMG_CC perfectly maintains a constant number of iterations for different discretizations, therefore $\mathcal{O}(N)$ time is achieved also for a model with highly complex geometry.

Fig. 16 shows that the memory needed to solve the toroidal inductor model by the A–method is roughly five times that of the φ–method. This depends on the larger number of unknowns required by the first method. Also in this benchmark, the PMU has a linear increase with N for all solver stages, and is approximately three times the PMU for input data.

VI. CONCLUSION

Two linear-time approaches (i.e., φ–method and A–method) for the solution of 3D magnetostatic field problems encompassing large-scale linear systems with millions of DOFs have been proposed. The key elements behind these approaches are computationally efficient and robust linear-time algebraic multigrid solvers for the solution of div-grad and curl-curl linear systems. Here, AGMG was used for div-grad systems, as this solver is known to be both robust and often faster than the state-of-the art alternatives. For curl-curl systems, AGMG_CC solver was considered; it corresponds to a variant of the auxiliary space preconditioner relying on AGMG for the approximate solution of the tree auxiliary systems. Numerical experiments on benchmarks of practical interest have shown that both AGMG and AGMG_CC reach linear time in all cases considered, while at the same time being faster and more robust than state-of-the-art algebraic multigrid and SOR-preconditioned Krylov solvers. For these reasons they constitute a valid alternative to the latter, typically adopted in commercial FEM software for EM analysis.

ACKNOWLEDGMENT

The authors would like to thank Prof. Oszkár Bíró, Institute for Fundamentals in Electrical Engineering (IGTE), Graz University of Technology; and Dr. Olivier Chadebec, G2Elab, Grenoble Alpes University, for their helpful suggestions on iterative solvers for nonlinear magnetostatics.

REFERENCES

[1] H. A. van der Vorst, *Iterative Krylov Methods for Large Linear Systems*. Cambridge, U.K.: Cambridge Univ. Press, 2003.

[2] E. Tonti, *The Mathematical Structure of Classical and Relativistic Physics: A General Classification Diagram*. Cambridge, MA, USA: Birkhäuser, 2013.

[3] O. Axelsson, *Iterative Solution Methods*. Cambridge, U.K.: Cambridge Univ. Press, 1994.

[4] Y. Saad, *Iterative Methods for Sparse Linear Systems*. Philadelphia, PA, USA: SIAM, 2003.

[5] J. Schoberl, S. Reitzinger, and M. Kaltenbacher, "Algebraic multigrid method for solving 3D nonlinear electrostatic and magnetostatic field problems," *IEEE Trans. Magn.*, vol. 36, no. 4, pp. 1561–1564, Jul. 2000.

[6] Y. Zhu and A. Cangellaris, *Multigrid Finite Element Methods for Electromagnetic Field Modeling*. Hoboken, NJ, USA: Wiley, 2005.

[7] M. Kaltenbacher, J. Schoberl, and M. Schinnerl, "Nested multigrid methods for the fast numerical computation of 3D magnetic fields," *IEEE Trans. Magn.*, vol. 36, no. 4, pp. 1557–1560, Jul. 2000.

[8] A. Brandt, S. McCormick, and J. Ruge, "Algebraic multigrid (AMG) for sparse matrix equations," in *Sparsity and Its Applications*, D. J. Evans, Ed., Cambridge, U.K.: Cambridge Univ. Press, 1985, pp. 257–284.

[9] U. Trottenberg, C. Oosterlee, and A. Schüller, *Multigrid*. London, U.K.: Academic, 2001.

[10] Y. Notay, "An aggregation-based algebraic multigrid method," *Electron. Trans. Numer. Anal.*, vol. 37, pp. 123–146, Oct. 2010.

[11] Y. Notay, "Numerical comparison of solvers for linear systems from the discretization of scalar PDEs," Service de Métrologie Nucléaire, Université Libre de Bruxelles, Brussels, Belgium, Tech. Rep., 2012. [Online]. Available: https://agmg.eu/numcompsolv.pdf

[12] A. Napov and Y. Notay, "Algebraic multigrid for moderate order finite elements," *SIAM J. Sci. Comput.*, vol. 36, no. 4, pp. A1678–A1707, Jan. 2014.

[13] Livermore Laboratories. *HYPRE: Software and Documentation*. Accessed: Jun. 10, 2024. [Online]. Available: http://www.llnl.gov/casc/hypre/

[14] S. Reitzinger and M. Kaltenbacher, "Algebraic multigrid methods for magnetostatic field problems," *IEEE Trans. Magn.*, vol. 38, no. 2, pp. 477–480, Mar. 2002.

[15] K. Watanabe, H. Igarashi, and T. Honma, "Comparison of geometric and algebraic multigrid methods in edge-based finite-element analysis," *IEEE Trans. Magn.*, vol. 41, no. 5, pp. 1672–1675, May 2005.

[16] R. Hiptmair and J. Xu, "Nodal auxiliary space preconditioning in H(curl) and H(div) spaces," *SIAM J. Numer. Anal.*, vol. 45, no. 6, pp. 2483–2509, Jan. 2007.

[17] R. Hiptmair and J. Xu, "Auxiliary space preconditioning for edge elements," *IEEE Trans. Magn.*, vol. 44, no. 6, pp. 938–941, Jun. 2008.

[18] T. V. Kolev and P. S. Vassilevski, "Parallel auxiliary space AMG for H(Curl) problems," *J. Comput. Math.*, vol. 27, no. 5, pp. 604–623, Jun. 2009.

[19] K. Preis, I. Bardi, O. Biro, C. Magele, W. Renhart, K. R. Richter, and G. Vrisk, "Numerical analysis of 3D magnetostatic fields," *IEEE Trans. Magn.*, vol. 27, no. 5, pp. 3798–3803, Sep. 1991.

[20] O. Biro, K. Preis, G. Vrisk, K. R. Richter, and I. Ticar, "Computation of 3-D magnetostatic fields using a reduced scalar potential," *IEEE Trans. Magn.*, vol. 29, no. 2, pp. 1329–1332, Mar. 1993.

[21] K. Preis, I. Bardi, O. Biro, C. Magele, G. Vrisk, and K. R. Richter, "Different finite element formulations of 3D magnetostatic fields," *IEEE Trans. Magn.*, vol. 28, no. 2, pp. 1056–1059, Mar. 1992.

[22] O. Bíró, "Edge element formulations of eddy current problems," *Comput. Methods Appl. Mech. Eng.*, vol. 169, nos. 3–4, pp. 391–405, Feb. 1999.

[23] P. Dular, "The benefits of nodal and edge elements coupling for discretizing global constraints in dual magnetodynamic formulations," *J. Comput. Appl. Math.*, vol. 168, nos. 1–2, pp. 165–178, Jul. 2004.

[24] F. Moro and L. Codecasa, "Coupling the cell method with the boundary element method in static and quasi–static electromagnetic problems," *Mathematics*, vol. 9, no. 12, p. 1426, Jun. 2021.

[25] L. Codecasa, "Refoundation of the cell method using augmented dual grids," *IEEE Trans. Magn.*, vol. 50, no. 2, pp. 497–500, Feb. 2014.

[26] L. Codecasa, V. Minerva, and M. Politi, "Use of barycentric dual grids for the solution of frequency domain problems by FIT," *IEEE Trans. Magn.*, vol. 40, no. 2, pp. 1414–1419, Mar. 2004.

[27] F. Moro, J. Smajic, and L. Codecasa, "A novel h–φ approach for solving eddy–current problems in multiply connected regions," *IEEE Access*, vol. 8, pp. 170659–170671, 2020.

[28] F. Moro, A. Napov, and L. Codecasa, "A hybrid a–φ cell method for solving eddy–current problems in 3–D multiply–connected domains," *IEEE Access*, vol. 9, pp. 158247–158260, 2021.

[29] N. Takahashi, K. Shimomura, D. Miyagi, and H. Kaimori, "Speed-up of nonlinear magnetic field analysis using a modified fixed-point method," *COMPEL-Int. J. Comput. Math. Electr. Electron. Eng.*, vol. 32, no. 5, pp. 1749–1759, Sep. 2013.

[30] Z. Ren, "Influence of the RHS on the convergence behaviour of the curl-curl equation," *IEEE Trans. Magn.*, vol. 32, no. 3, pp. 655–658, May 1996.

[31] O. Bíró and K. Preis, "Finite element calculation of time-periodic 3D eddy currents in nonlinear media," in *Proc. Sel. Papers 3rd Japan-Hung. Joint Seminar Appl. Electromagn. Mater. Comput. Technol.*, 1995, pp. 62–74.

[32] Y. Notay. *AGMG Software and Documentation*. Accessed: Apr. 10, 2024. [Online]. Available: http://agmg.eu

[33] Y. Notay, "Flexible conjugate gradients," *SIAM J. Sci. Comput.*, vol. 22, no. 4, pp. 1444–1460, Jan. 2000.

[34] Y. Notay and P. S. Vassilevski, "Recursive Krylov-based multigrid cycles," *Numer. Linear Algebra Appl.*, vol. 15, no. 5, pp. 473–487, Jun. 2008.

[35] A. Napov and Y. Notay, "Algebraic analysis of aggregation-based multigrid," *Numer. Linear Algebra Appl.*, vol. 18, no. 3, pp. 539–564, 2011.

[36] A. Napov and Y. Notay, "An algebraic multigrid method with guaranteed convergence rate," *SIAM J. Sci. Comput.*, vol. 34, no. 2, p. A1109, Jan. 2012.
[37] H. Igarashi, "On the property of the curl-curl matrix in finite element analysis with edge elements," *IEEE Trans. Magn.*, vol. 37, no. 5, pp. 3129–3132, Jun. 2001.
[38] T. Kolev and P. Vassilevski, "Some experience with a H1-based auxiliary space AMG for H(curl) problems," LLNL, Livermore, CA, USA, Tech. Rep. UCRL-TR-221841, 2006.
[39] A. Napov. *AGMG_CC: Software and Documentation*. Accessed: Jun. 10, 2024. [Online]. Available: http://metronu.ulb.ac.be/AGMG_CC/
[40] J. R. Brauer, *Magnetic Actuators and Sensors*. Hoboken, NJ, USA: Wiley, 2014.
[41] A. Pokryvailo, "Calculation of inductance of sparsely wound toroidal coils," in *Proc. COMSOL Conf.*, Boston, MA, USA, 2016, pp. 1–7.

FEDERICO MORO (Senior Member, IEEE) received the Laurea degree in electrical engineering, the Ph.D. degree in bioelectromagnetic and electromagnetic compatibility, and the B.S. degree in mathematics from the University of Padova, Italy, in 2003, 2007, and 2012, respectively. He was a Visiting Student with the Department of Physics, Swansea University, Wales, U.K., in 2005; and a Visiting Professor with the G2ELab, Grenoble, France, in 2020. From 2007 to 2010, he was a Research Associate with the Department of Electrical Engineering, University of Padova. From 2010 to 2020, he was an Assistant Professor of electrical engineering with the Department of Industrial Engineering, University of Padova, where he has been an Associate Professor of electrical engineering, since 2020. He is the author of more than 100 papers in peer-reviewed international journals and conference proceedings. His research interests include numerical methods for computing electromagnetic problems and the numerical modeling of multiphysics and multiscale problems. He was awarded the Best Oral Presentation at UPEC 2006 and the Best Paper at ASME IDETC/CIE 2017, IDETC/CIE 2022, ECCE/IAS 2022, and Electrimacs 2019 Conferences. He received the Outstanding Reviewer Award from Emerald Publishing, in 2024. He received the National Scientific Qualification as a Full Professor (09/E1-Elettrotecnica), in 2021.

LORENZO CODECASA (Member, IEEE) received the Ph.D. degree in electronics engineering from Politecnico di Milano, Milan, Italy, in 2001. Since 2002, he has been with the Department of Electronics, Information and Bioengineering (DEIB), Politecnico di Milano, where he is currently a Full Professor of electrical engineering. In his research areas, he has authored or co-authored more than 260 papers in refereed international journals and conference proceedings. His research contributions are in the theoretical analysis and the computational investigation of electronic circuits and electromagnetic fields. He is particularly active in the research of heat transfer and thermal management of electronic components and packages, in which he has been introducing original industrial-strength approaches to the extraction of compact thermal models currently available in market-leading commercial software. He is listed in Stanford University's ranking, as One of the World's 2% Most Influential Scientists. He received the Harvey Rosten Award for Excellence (twice), in 2016 and 2023; and the Three Best Papers Awards at THERMINIC, in 2014, 2017, and 2019. In 2024, he was awarded the Best Associate Editor Award from IEEE TRANSACTIONS ON COMPONENTS, PACKAGING AND MANUFACTURING TECHNOLOGY. He has served as the Program Chair, the Vice-General Chair, and the General Chair of THERMINIC.

ARTEM NAPOV received the Ph.D. degree in engineering from Université Libre de Bruxelles (ULB), in 2010. From 2010 to 2012, he was a Postdoctoral Fellow with the Computational Science Division, Lawrence Berkeley National Laboratory. In 2012, he was an Assistant Professor with École Poytechnique de Bruxelles (ULB), where he was an Associate Professor, in 2017. His research interest includes numerical methods for the solution of large sparse linear systems of equations arising from the discretization of partial differential equations.

Open Access funding provided by 'University of Padua' within the CRUI CARE Agreement