IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 5, 2020, accepted December 19, 2020, date of publication December 24, 2020, date of current version January 5, 2021.

Digital Object Identifier 10.1109/ACCESS.2020.3047153

Domain Decomposition With Non-Conforming Polyhedral Grids

FEDERICO MORO[1], (Member, IEEE), AND LORENZO CODECASA[2], (Member, IEEE)
[1]Dipartimento di Ingegneria Industriale, Università degli Studi di Padova, 35131Padua, Italy
[2]Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, 20133 Milan, Italy

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

ABSTRACT A novel mortar approach for the domain decomposition of field problems discretized in terms of nodal variables by the cell method is here proposed. This approach allows the use of both arbitrary polyhedral meshes and non–conforming discretizations, without limitations or complications due to the mesh type or the model geometry. Therefore, it provides a new domain decomposition method that can be practically used in engineering applications for coupling different parts of a model, which can be independently discretized and then reassembled together. More precisely: 1) Any part of the computational domain is first separately modeled in order to assess the mesh type and size that are best suited for ensuring an accurate local field reconstruction; 2) The different discretized parts can be combined together in order to obtain an accurate solution of a composite problem, while maintaining the local discretizations already determined. As a main advantage over existing mortar approaches, the algebraic structure of the final matrix system—derived by the cell method discretization—is not altered by the introduction of mortar interface conditions. As a result, the same preconditioning and iterative solver strategy can be extended as is to the proposed mortar method. This approach is validated by a convergence analysis on an analytical test case and its effectiveness for practical applications is assessed on a real–sized engineering problem.

INDEX TERMS Multiscale, cell method, mortar method, domain decomposition, polyhedral mesh.

I. INTRODUCTION

The key idea of domain decomposition methods (DDMs) is to model a field problem in a computational domain by: 1) subdividing the domain into proper subdomains; 2) independently discretizing any subdomain; and finally 3) properly coupling the discretized model parts [1].

DDMs can be of utmost importance for many engineering applications because the different parts of the model can be independently discretized and then reassembled together to simulate the entire model. Such a use of DDMs has been proposed, e.g., in [2] for large electro–thermal analyses of integrated circuits. DDMs are also needed, for instance, when mesh adaptivity is desired [3]. This may occur with local field anisotropies due to, e.g., singular sources, corner singularities, or discontinuous material coefficients. As a further advantage of DDMs, parallel computing can be applied for large sized problems, by sharing among different processors the discretized models of the single components [4], [5].

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

Among the different types of DDMs, mortar methods (MMs) are widely adopted because they can deal with both geometric and functional non–conformity, so that different kinds of meshes and different numerical methods can be chosen in different parts of the model at the same time [6]. A mortar approach was first proposed in [7] to combine spectral element method (SEM) and finite element method (FEM). MMs make use of separate finite element discretizations on non–overlapping subdomains. Meshes on these subdomains are not required to match on the interface, so that different kind of elements (e.g., quadrilaterals or triangles for 2–D problems) can be used in each subdomain. The subdomain from which the interface inherits its discretization is called *slave* or *non–mortar* side; the other one is termed *master* or *mortar* side. The solution continuity at the interface of the partial differential equation (PDE) is enforced by Lagrange multipliers, suitably chosen to preserve the solution accuracy.

In order to ensure numerical stability and the convergence of MMs, basis functions of local FEM spaces (each of them related to a different subdomain) have to fulfill additional

constraints at boundary nodes [8]. These requirements may lead to an unacceptable complexity, especially in the case of 3–D problems. Moreover, MMs have significant limitations which do not guarantee a full flexibility in numerical modeling, namely MMs are typically restricted to models with: 1) hexahedral or tetrahedral meshes; 2) subdomain interfaces without corners [9], [10]. In particular, hexahedral meshes can be best used for discretizing the structured parts of the model and, after mesh refinement, they typically lead to a propagation of degrees of freedom (DOFs). On the other hand, tetrahedral meshes cannot be used for discretizing parts with high aspect ratio, providing otherwise poor accuracy.

Another important limitation of classical MMs is concerned with the solution of the final system of discretized equations. Lagrange multipliers used for enforcing continuity cannot be eliminated from the final system of equations, which is in a saddle–point form. This class of problems in general cannot be either easily preconditioned or efficiently solved as thoroughly discussed in [11]. In order to get rid of Lagrange multipliers, biorthogonal basis functions, limited to tetrahedral meshes only, have been proposed in [12], [13].

The basic idea of this work is to introduce a new mortar method for generic polyhedral meshes, overcoming all above drawbacks. In this way, the use of polyhedral meshes, combined to a non–conforming discretization approach, provides a MM, which can be used without limitations in engineering applications, for coupling discretized models, independently derived for each part of the computational domain. More precisely: 1) Any part of the domain can be first separately modeled in order to assess the mesh type and size that are best suited for ensuring an accurate local field reconstruction; 2) The different discretized models can be coupled in order to obtain an accurate solution of a composite problem, while maintaining the local discretizations already determined. As a main advantage of the proposed approach, the final matrix system, with the mortar approach, preserves the algebraic structure of a standard CM formulation, on a undecomposed domain, without mortar. As a result the same preconditioning and iterative solver strategy can be applied to such a MM.

In order to make the presentation as simple as possible, the MM is presented for a 3–D electrostatic problem formulated in terms of electric potential. This presentation can however be straightforwardly adapted for any physical problem formulated in terms of a field associated to the mesh nodes, according to Tonti's framework [14], such as heat transfer problems in the temperature field, elasticity problems in the displacement field, acoustic problems in the pressure field.

In order to introduce polyhedral elements, which are not implemented by conventional FEM, the cell method (CM) with piecewise–uniform polyhedral elements, proposed in [15], is here adopted. Unlike FEM, CM provides the field problem equations directly in algebraic form suitable for numerical computation. A combinatorial discrete model is thus constructed and formulated in a similar way to electric network problems, as discussed in [16]. Piecewise–uniform bases have these relevant features compared to other polyhedral methods proposed in literature: 1) The Gaussian quadrature of rational interpolants such as with polytope finite elements [17], which can be inaccurate and computationally expensive, is not required; 2) Matrix assembly is entirely jacobian–free with piecewise-uniform bases; 3) The computation of multiple local matrix systems such as with the mimetic finite difference method [18], or the virtual element method [19], which may highly affect the computational burden when analyzing real–sized problems, is not required. Piecewise–uniform basis functions are defined on star–shaped polyhedra, with planar faces, and are therefore suitable for discretizing any type of model geometry. Since these functions are directly constructed over a polyhedron, rather than over its tetrahedral partition, a limited number of DOFs is required for local field interpolation. The so–called *energy approach* [15] makes it possible to obtain symmetric and positive mass matrices, useful for discretizing local constitutive relationships. These algebraic properties lead to well–behaved final matrix systems, to be treated by the same efficient iterative solvers used with classical FEM over standard hexahedral or tetrahedral partitions.

A CM–based mortar method (MCM), based on point–wise matching for interface gluing, was proposed for tetrahedral meshes in [20], [21]. Accuracy of continuity constraints was then improved in [22], for 2–D elliptic problems, and in [23], for 3–D elliptic problems, by using dual bi-orthogonal bases leading to a local elimination of Lagrange multipliers. The CM has shown to be well suited also for coupling different formulations or physics, since topological relationships are split from constitutive ones and integral variables are used to enforce element continuity. An unsymmetric CM–BEM formulation was proposed in [24] for solving eddy current problems by coupling the CM to the boundary element method (BEM). A symmetric CM–BEM was presented for magnetostatics in [25], and then extended to eddy current problems in multiply–connected domains [26]. Recently, a CM for eddy–problems on multiply–connected domains was proposed in [27]. A coupling between different physics was provided in [28] for solving electro–thermal problems. The CM based on polyhedral elements (poly–CM) was recently extended to 3–D elastic and piezo–elastic problems for discretizing thin–layered composite structures of energy harvesters [29]. Numerical results showed a very good agreement with higher–order FEM, after applying a smoothing procedure. To give an example, in [29], by considering a realistic 3–D model of a cantilevered piezoelectric harvester—simulated in static conditions, with a relatively coarse mesh (14 748 DOFs)—, the maximum discrepancy of the tip displacement from second–order FEM was found to be about 3.5%, for smoothed poly–CM, and 87.5% without smoothing (first–order CM).

The paper is organized as follows. The CM over polyhedral meshes is first presented in Section II without domain partition. The mortar approach derived from this poly–CM is illustrated in Section III. Implementation details for building the mortar projection, at the core of polyhedral MCM, are

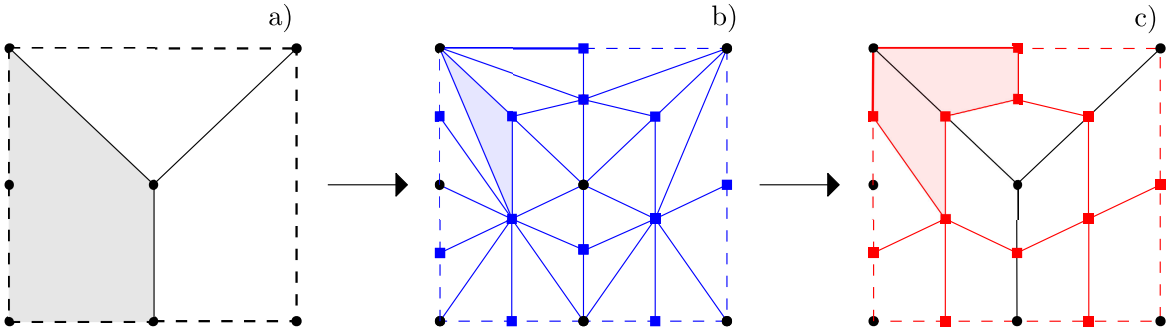

FIGURE 1. (a) Primal grid $\mathcal{G}_\Omega$: primal nodes are indicated with black dots, primals cells are polygons (shaded in gray); (b) Barycentric subdivision: primal cells are split into barycentric triangles (shaded in blue); (c) Augmented dual grid $\widetilde{\mathcal{G}}_{\Omega\Gamma}$: dual polygons (shaded in red) are obtained by aggregating triangles in blue around any primal node. Note that the boundary (black dashed line) is split on its turn into barycentric cells (blue thick line),which are joined into 1-D dual cells (red thick line).

provided in Section IV. Numerical results are discussed in Section V, where an analytical (with a convergence test) and a numerical benchmark (consisting in a real–sized engineering problem) are provided. Implementation details about the constitutive matrix assembly are also given in the Appendix.

II. CELL METHOD ON POLYHEDRAL MESHES

The CM discretization framework is presented first on the whole computational domain, without applying a domain decomposition. The CM variant proposed in [30], which introduces the key idea of *augmented dual grids* (instead of dual grids as in classical CM), is here used. This novel geometric framework fixes some limitations of the original CM approach (e.g., not enforcing properly energy conservation) and introduces new discrete operators for handling both boundary and interface conditions. In such a way continuity conditions, at the core of any MM, can be correctly implemented in the CM in order to ensure field trace continuity.

A. GEOMETRIC SETTING

The computational domain $\Omega \subset \mathbb{R}^3$ is discretized into a polyhedral mesh $\mathcal{G}_\Omega$, i.e., the *domain primal grid*, which is made of N vertexes, E edges, F faces, and V polyhedra or volumes. The restriction of $\mathcal{G}_\Omega$ to the boundary Γ is the *boundary primal grid* $\mathcal{G}_\Gamma$, in which vertexes are traces of bulk primal edges of $\mathcal{G}_\Omega$, edges are traces of bulk primal faces, and faces are traces of bulk primal volumes. Polyhedral and polygonal cell complexes $\mathcal{G}_\Omega$ and $\mathcal{G}_\Gamma$ are then split into their corresponding barycentric subdivisions, which are obtained by splitting any polyhedron or polygon into a set of tetrahedra or triangles having as a common apex the cell centroid. The corresponding *domain dual grid* $\widetilde{\mathcal{G}}_\Omega$ (made of $\widetilde{N}$ vertexes, $\widetilde{E}$ edges, $\widetilde{F}$ faces, and $\widetilde{V}$ volumes) and *boundary dual grid* $\widetilde{\mathcal{G}}_\Gamma$ (made of $\widetilde{N}_\Gamma$ vertexes, $\widetilde{E}_\Gamma$ edges, $\widetilde{F}_\Gamma$ faces) are finally obtained by joining barycentric tetrahedra or triangles. The *augmented dual grid* is the union of dual domain and boundary grids, that is $\widetilde{\mathcal{G}}_{\Omega\Gamma} = \widetilde{\mathcal{G}}_\Omega \cup \widetilde{\mathcal{G}}_\Gamma$ according to [30]. This specific geometric construction provides a one–to–one correspondence between primal and dual grid entities, so that $\widetilde{N} = V$, $\widetilde{E} = F$, $\widetilde{F} = E$, $\widetilde{V} = N$.

For the sake of simplicity, the whole discretization process is sketched for a 2–D domain in Fig. 1, even though both the mathematical formulation in Section III and numerical experiments in Section V refer to a more general 3–D setting. The augmented dual grid in Fig. 1 c, discretizing a unit square $\Omega = [0, 1]^2$, is obtained from the primal grid (Fig. 1 a) by assembling triangles of the barycentric subdivision (Fig. 1 b). $\mathcal{G}_\Omega$, whose nodes are indicated by black dots in Fig. 1 a, is made by polygons with an arbitrary number of vertexes (e.g., the polygonal cell is shaded in gray). The corresponding barycentric subdivision is then built by taking as mesh vertexes primal nodes and centroids of primal edges and polygons. $\widetilde{\mathcal{G}}_{\Omega\Gamma}$ is finally obtained by aggregating barycentric triangles around any primal node in order to get a one–to–one correspondence between primal nodes and dual cells. In Fig. 1 c, a dual polygon (partitioned into barycentric triangles) is shaded in red color. Dual nodes, which are centroids of primal polygons and centroids of boundary edges, are indicated by red (filled) squares. Note that, the other way round, a one–to-one correspondence exists also between dual nodes and primal cells, which can be either primal polygons of $\mathcal{G}_\Omega$ or boundary edges of $\mathcal{G}_\Gamma$.

B. DISCRETE FIELD VARIABLES

Primal and dual cell complexes carry a different orientation according to [16]. Geometric entities of $\mathcal{G}_\Omega$ are endowed by an *inner* orientation so that any vertex is oriented as a sink—i.e., any edge incident to that vertex is pointing inward—, any edge is oriented by a transversing direction (from one end to the other), any face is oriented clockwise or counterclockwise, and any polyhedral cell is oriented by assuming that its facets are oriented counterclockwise with respect to the corresponding exterior normal. Similar considerations hold for the boundary complex $\mathcal{G}_\Gamma$. Geometric entities of $\widetilde{\mathcal{G}}_{\Omega\Gamma}$ are endowed with *outer* orientation, which is simply inherited from that one of primal grid entities by one–to–one correspondence. In such a way, e.g., a dual edge is oriented by the same orientation of its corresponding primal face, that is a clockwise or counterclockwise rotation around it.

Orientation is useful to define properly discrete field variables (arrays of DOFs), because a local reference is required

in order to compute physical quantities such as potentials or line, surface, volume integrals. In such a way, any geometric element carries its specific orientation as a local reference frame. As an example, the case of electrostatics is here considered. Arrays of DOFs are electric potentials on primal nodes n, i.e., $\Phi_\Omega = (\phi_n)_{n\in\mathcal{G}_\Omega}$; voltages along primal edges e, i.e., $\mathbf{u}_\Omega = (u_e)_{e\in\mathcal{G}_\Omega}$, where $u_e = \int_e \mathbf{E}(x)\cdot\mathbf{t}(x)\,d\gamma_x$ is the line integral of the electric field $\mathbf{E}$ and $\mathbf{t}$ is the unit tangent vector related to e; fluxes of the electric displacement $\mathbf{D}$ through dual faces $\widetilde{f}$, i.e., $\widetilde{\mathbf{d}}_\Omega = (\widetilde{d}_{\widetilde{f}})_{\widetilde{f}\in\widetilde{\mathcal{G}}_\Omega}$, where $\widetilde{d}_{\widetilde{f}} = \int_{\widetilde{f}} \mathbf{D}(x)\cdot\mathbf{n}(x)\,d\sigma_x$ and $\mathbf{n}$ is the unit normal vector related to $\widetilde{f}$.

C. TOPOLOGICAL RELATIONS

Local reference frames are related together by connectivity between elements. An *incidence* number is $+1$ if a pair of connected geometric entities carries the same orientation, -1 otherwise, and 0 if they are disconnected. Connectivity between grid entities is established by the following incidence matrices with $\{0,-1,+1\}$ coefficients, i.e., $\mathbf{G}_\Omega$ (edges to nodes on $\mathcal{G}_\Omega$), $\widetilde{\mathbf{D}}_\Omega = -\mathbf{G}_\Omega^{\mathrm{T}}$ (volumes to faces on $\widetilde{\mathcal{G}}_\Omega$), and $\widetilde{\mathbf{D}}_{\Omega\Gamma}$ (volumes on $\widetilde{\mathcal{G}}_\Omega$ to faces on $\widetilde{\mathcal{G}}_\Gamma$) [30].

Let e be a primal edge oriented from vertex m to n. By exploiting the gradient theorem, the electric compatibility condition $\mathbf{E} = -\nabla\phi$ provides the electric voltage along the edge e as a potential difference between m and n, that is:

$$u_e = \int_e \mathbf{E}(x)\cdot\mathbf{t}(x)\,d\gamma_x = \phi_m - \phi_n, \tag{1}$$

which is written for all nodes and edges of $\mathcal{G}_\Omega$, as:

$$\mathbf{u}_\Omega = -\mathbf{G}_\Omega\,\Phi_\Omega. \tag{2}$$

On the augmented dual grid, electric equilibrium is imposed by applying locally Gauss's theorem to any dual volume, that is $\nabla\cdot\mathbf{D} = \rho$, with ρ electric free charge. By integrating this expression over any dual volume $\widetilde{v}$ with boundary $\partial\widetilde{v}$:

$$\begin{aligned}\int_{\widetilde{v}} \nabla\cdot\mathbf{D}(x)\,dx &= \int_{\partial\widetilde{v}} \mathbf{D}(x)\cdot\mathbf{n}(x)\,d\sigma_x \\ &= \sum\nolimits_{\widetilde{f}\,\in\,\partial\widetilde{v}} \pm\widetilde{d}_{\widetilde{f}} = \widetilde{q}_{\widetilde{v}},\end{aligned} \tag{3}$$

where $\widetilde{q}_{\widetilde{v}} = \int_{\widetilde{v}} \rho(x)\;dx$ is the total charge in the dual cell and the sign in the sum over dual facets is positive if $\widetilde{f}$ is endowed with outward orientation, negative otherwise. Writing this equation for all cells pertaining to $\widetilde{\mathcal{G}}_{\Omega\Gamma}$ and recasting in matrix form provide Gauss's theorem in global form, as:

$$\widetilde{\mathbf{D}}_\Omega\,\widetilde{\mathbf{d}}_\Omega + \widetilde{\mathbf{D}}_{\Omega\Gamma}\,\widetilde{\mathbf{d}}_\Gamma = \widetilde{\mathbf{q}}_\Omega, \tag{4}$$

where $\widetilde{\mathbf{d}}_\Gamma = (\widetilde{d}_{\widetilde{f}})_{\widetilde{f}\in\mathcal{G}_\Gamma}$ is the array of electric displacement fluxes through boundary dual faces and $\widetilde{\mathbf{q}}_\Omega = (\widetilde{q}_{\widetilde{v}})_{\widetilde{v}\in\widetilde{\mathcal{G}}_\Omega}$ is the array of total free charges inside dual volumes. $\widetilde{\mathbf{D}}_{\Omega\Gamma}$ is a selection matrix, made of $\{0,1\}$ coefficients, which identifies all the dual boundary faces incident to domain dual cells.

D. DISCRETE CONSTITUTIVE RELATION

The local constitutive relationships for an electrostatic problem with anisotropic media, with permittivity tensor ε, is:

$$\mathbf{D}(x) = \varepsilon(x)\,\mathbf{E}(x). \tag{5}$$

This relationship is discretized with the CM by using *edge elements* $\mathbf{w}_e$, i.e., edge element vector basis functions, defined on polyhedral grids [15]. These functions enjoy the following properties: i) They form a vector basis; ii) They enforce the continuity of the tangential component of the electric field; iii) They enforce a mapping property from primal to dual grid (*consistency*), i.e., $\int_v \mathbf{w}_e(x)\,dx = \widetilde{\mathbf{f}}_e$, where v is a generic primal polyhedral cell, e is a primal edge related to the basis vector, and $\widetilde{\mathbf{f}}_e$ is the area vector related to $\widetilde{f}_e$, i.e., the dual face in one–to–one correspondence with e; iv) They are able to represent exactly any elementwise constant vector field (*uniformity*). These edge functions require a preliminary construction for their definition. The primal cell (polyhedron) is first split into a barycentric subdivision made of tetrahedra sharing as a common apex the polyhedron centroid. In Fig. 2, tetrahedron $(\mathbf{e}_1,\mathbf{e}_2,\mathbf{c}_{f1},\mathbf{c}_v)$ is made up by a triangular face, obtained by joining a primal edge $(\mathbf{e}_1,\mathbf{e}_2)$ with a primal face centroid $\mathbf{c}_{f1}$, and the polyhedron centroid $\mathbf{c}_v$. A similar construction holds for $(\mathbf{e}_1,\mathbf{e}_2,\mathbf{c}_{f2},\mathbf{c}_v)$, sharing the same triangular face. Then, for any edge e', tetrahedra attached to it are assembled to form a *uniformity region* $v_{e'}$ in which $\mathbf{w}_e$ is constant and it is defined as:

$$\mathbf{w}_e(x) = \frac{\widetilde{\mathbf{f}}_e}{\widetilde{\mathbf{f}}_e\cdot\mathbf{e}}\,\delta_{ee'} + \left(\mathbf{I} - \frac{\widetilde{\mathbf{f}}_{e'}\otimes\mathbf{e}'}{\widetilde{\mathbf{f}}_{e'}\cdot\mathbf{e}'}\right)\frac{\widetilde{\mathbf{f}}_e}{|v|}, \quad x\in v_{e'}, \tag{6}$$

where $\delta_{ee'}$ is the Kronecker delta ($\delta_{ee'} = 1$ if $e = e'$, 0 otherwise), $\mathbf{I}$ is the identity tensor, $|v|$ is the cell volume, and $\otimes$ is the dyadic product between vectors $\mathbf{a}$ and $\mathbf{b}$, i.e., $(\mathbf{a}\otimes\mathbf{b})\,\mathbf{c} = \mathbf{a}\,(\mathbf{b}\cdot\mathbf{c})$ for any vector $\mathbf{c}$. From the previous expression, it can be easily proven that also the unity property holds, i.e., $\int_e \mathbf{w}_e(x)\cdot\mathbf{t}(x)\,d\gamma_x = \delta_{ee'}$, from which also property i) can be easily derived. Note that the number of uniformity regions equals the number of primal edges inside v and the support of $\mathbf{w}_e$ is made up of all polyhedra attached to e. Moreover, the area normal of any (non–planar) dual face can be computed as $\widetilde{\mathbf{f}}_e = \int_{\widetilde{f}_e} \mathbf{n}(x)\,d\sigma_x = 1/2\,(\mathbf{c}_v-\mathbf{c}_e)\times(\mathbf{c}_{f1}-\mathbf{c}_{f2})$, where $\mathbf{c}_e$ is the edge centroid related to e (Fig. 2).

By assuming a locally constant field when the polyhedral mesh is very refined, the following expansion holds:

$$\mathbf{E}(x) = \sum_{e=1}^{E} u_e\,\mathbf{w}_e(x), \quad x\in\Omega, \tag{7}$$

where E is the total number of edges of $\mathcal{G}_\Omega$ and, from the unity property, expansion coefficient u_e turns out to be the electric voltage along e, i.e., $u_e = \int_e \mathbf{E}(x)\cdot\mathbf{t}(x)\,d\gamma_x$.

By assuming a locally constant field in the support of $\mathbf{w}_e$, for any e, and by using the consistency property, the dielectric displacement flux through a dual face $\widetilde{f}_e$, related to edge e,

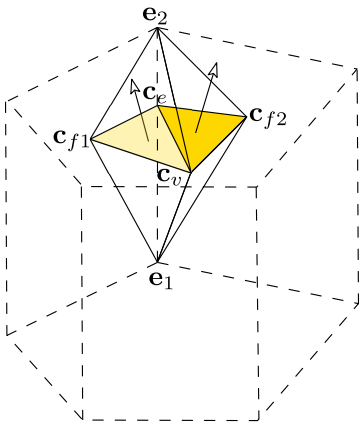

FIGURE 2. Uniformity region (straight line) related to an oriented edge ($\mathbf{e}_1, \mathbf{e}_2$). The polyhedral cell v is in dashed line.

is computed as:

$$\widetilde{d}_{\widetilde{f}_e} = \int_{\widetilde{f}_e} \mathbf{D}(x)\cdot\mathbf{n}(x)\,d\sigma_x \approx \mathbf{D}(x)\cdot\int_{\widetilde{f}_e}\mathbf{n}(x)\,d\sigma_x$$
$$= \mathbf{D}(x)\cdot\widetilde{\mathbf{f}}_e = \int_\Omega \mathbf{D}(x)\cdot\mathbf{w}_e(x)\,dx, \tag{8}$$

which, by using (5) and (7), becomes:

$$\widetilde{d}_{\widetilde{f}_e} = \int_\Omega \mathbf{w}_e(x)\ \cdot\varepsilon(x)\,\mathbf{E}(x)\,dx$$
$$= \sum_{e'=1}^{E}\int_\Omega \mathbf{w}_e(x)\cdot\varepsilon(x)\,\mathbf{w}_{e'}(x)\ u_{e'}(t)\,dx. \tag{9}$$

For non–locally constant fields encountered in general problems this relationships represents a numerical approximation, whose accuracy increases by refining the domain discretization. Writing (9) for any dual face of $\widetilde{\mathcal{G}}_\Omega$, or, correspondingly, for any edge e, one obtains the dielectric constitutive relationship in the whole domain. In matrix form, it is expressed as:

$$\widetilde{\mathbf{d}}_\Omega = \boldsymbol{\varepsilon}_\Omega\,\mathbf{u}_\Omega, \tag{10}$$

where $\boldsymbol{\varepsilon}_\Omega = (\varepsilon_{ee'})_{e,e'\in\mathcal{G}_\Omega}$ is dielectric constitutive matrix, whose coefficients are:

$$\varepsilon_{ee'} = \int_\Omega \mathbf{w}_e(x)\cdot\varepsilon(x)\,\mathbf{w}_{e'}(x)\,dx. \tag{11}$$

Details of the algorithm used for assembling the constitutive matrix are reported in the Appendix.

E. MATRIX SYSTEM

By combining topological relationships (2) and (4) with the consitutive relationship (10), the final matrix system for the electrostatic problem in Ω is obtained. By noting that $\widetilde{\mathbf{D}}_\Omega = -\mathbf{G}_\Omega^{\mathrm{T}}$, it turns out to be:

$$-\mathbf{G}_\Omega^{\mathrm{T}}\,\boldsymbol{\varepsilon}_\Omega\mathbf{G}_\Omega\,\Phi_\Omega + \widetilde{\mathbf{D}}_{\Omega\Gamma}\,\widetilde{\mathbf{d}}_\Gamma = \widetilde{\mathbf{q}}_\Omega. \tag{12}$$

This matrix system is resolved after imposing scalar potentials on the Dirichlet boundary Γ_D, i.e., on a subset of variables Φ_Ω, and/or Neumann boundary conditions on the Neumann boundary Γ_N, i.e., on a subset of variables $\widetilde{\mathbf{d}}_\Gamma$. Note that $\Gamma = \Gamma_D\cup\Gamma_N$ provides a partition of the domain boundary in order to ensure a unique solution of (12).

III. POLYHEDRAL MORTAR FORMULATION

The computational domain Ω is decomposed in K non-overlapping connected sub–domains Ω_k, $k = 1\ldots K$. The boundary of any domain is indicated as $\Gamma_k = \partial\Omega_k$. The interface Γ_{ij} between a pair of adjacent subdomains Ω_i and Ω_j is defined as the intersection of their boundaries, i.e. $\Gamma_{ij} = \Gamma_i\cap\Gamma_j$ with $i,j = 1,\ldots,K$. The union of all interfaces is denoted as $\Gamma = \bigcup\Gamma_{ij}$, with abuse of notation with respect to Section II. Note that Γ does not include the exterior boundary $\Gamma_0 = \partial\Omega$. In the proposed mortar approach each Ω_k is discretized independently with a non–conforming polyhedral mesh so that on the interface Γ two different sides are defined, i.e., the *slave* side Γ^-, from which Γ inherits its discretization, and the *master* or *mortar* side Γ^+, according to definitions already provided in Section I. In the following, Γ is assumed to be a piecewise–planar surface in order to make the use of the intersection algorithm, described in Section IV, possible. After the decomposition of Ω into subdomains, (12) has to be rephrased as follows:

$$-\mathbf{G}_\Omega^{\mathrm{T}}\,\boldsymbol{\varepsilon}_\Omega\mathbf{G}_\Omega\,\Phi_\Omega + \widetilde{\mathbf{D}}_{\Omega\Gamma}\,\widetilde{\mathbf{d}}_\Gamma = \widetilde{\mathbf{q}}_\Omega - \widetilde{\mathbf{D}}_{\Omega\Gamma_0}\,\widetilde{\mathbf{d}}_{\Gamma_0}, \tag{13}$$

where $\widetilde{\mathbf{d}}_\Gamma$, differently from (12), indicate the interface fluxes and $\widetilde{\mathbf{d}}_{\Gamma_0}$ the exterior boundary fluxes. (13) is solved only after imposing continuity constraints between subdomains and Dirichlet and/or Neumann BCs. The construction of mortar continuity constraints is discussed in the following.

The continuity of the electric field tangent component and the continuity of the electric displacement normal component are weakly imposed by introducing novel basis functions. According to Galerkin's approach, test functions are chosen in the same space of shape functions. DOFs are electric potentials at the vertexes of primary boundary grids, and fluxes of the electric displacement through the faces of dual boundary grids. In particular, fluxes are evaluated by assuming slave and mortar sides oriented by opposite unit normals, i.e., $\mathbf{n}^- = -\mathbf{n}^+$ on Γ^+. With this convention, the following DOFs are defined: $\Phi_{\Gamma^-} = (\phi_n^-)_{n\in\mathcal{G}_{\Gamma^-}}$, $\Phi_{\Gamma^+} = (\phi_m^+)_{m\in\mathcal{G}_{\Gamma^+}}$ are the arrays of the electric scalar potentials on the primal boundary vertexes of slave and master sides, and $\widetilde{\mathbf{d}}_{\Gamma^-} = (\widetilde{d}_n^-)_{n\in\widetilde{\mathcal{G}}_{\Gamma^-}}$, $\widetilde{\mathbf{d}}_{\Gamma^+} = (\widetilde{d}_m^+)_{m\in\widetilde{\mathcal{G}}_{\Gamma^+}}$ are the arrays of fluxes of the electric displacement through the faces of corresponding dual grids, respectively. Array coefficients are $\widetilde{d}_n^- = \int_{\widetilde{f}_n} D^-(x)\,d\sigma_x$ and $\widetilde{d}_n^+ = \int_{\widetilde{f}_n} D^+(x)\,d\sigma_x$, for the slave and master side, respectively, where $D^-(x) = \mathbf{D}(x)\cdot\mathbf{n}^-(x)$, $x\in\Gamma^-$, and $D^+(x) = \mathbf{D}(x)\cdot\mathbf{n}^+(x)$, $x\in\Gamma^+$, are the normal components of the electric displacement.

By using the same discretization approach of bulk domains, scalar fields are assumed to be locally piecewise constant so that accuracy increases when mesh is refined. Novel piecewise–constant bases for the potential and the normal component of the electric displacement on both slave

and master sides are defined with a support on the dual mesh. Fig. 3 shows basis function supports for nodes n, m on the slave and master side, respectively.

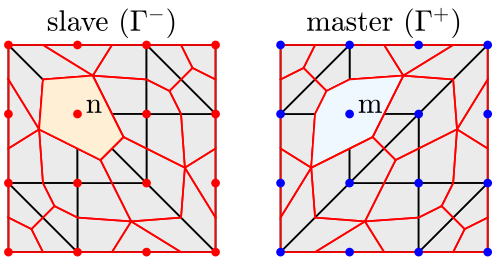

FIGURE 3. Non–conforming slave ad master surfaces discretized by polygonal meshes (in black line). Mortar basis functions related to primal nodes *n*, *m* have the corresponding dual cells as their supports (shaded in color). Primal nodes are depicted in red color on Γ^-, in blue color on Γ^+. Dual mesh (overlapping primal mesh) is indicated in red line.

For any primal vertex n of $\mathcal{G}_{\Gamma^+}$ or $\mathcal{G}_{\Gamma^-}$, related in one–to–one correspondence to a dual cell $\widetilde{f}_n$, its corresponding *primal mortar basis function* is defined as:

$$w_n(x) = \begin{cases} 0, & x \notin \widetilde{f}_n, \\ 1, & x \in \widetilde{f}_n, \end{cases} \tag{14}$$

which is a pulse function on the dual complex. In such a way, these functions are linearly independent and are able to reconstruct exactly an elementwise constant scalar field. Shape functions for the normal component of the electric displacement on dual grids $\widetilde{\mathcal{G}}_{\Gamma^+}$ or $\widetilde{\mathcal{G}}_{\Gamma^-}$, termed *dual mortar basis functions*, are then defined on the same support, as:

$$\widetilde{w}_n = \frac{1}{|\widetilde{f}_n|}\, w_n, \tag{15}$$

where $|\widetilde{f}_n|$ indicates the dual cell area. From this definition, the unity property is automatically fulfilled, that is $\int_{\widetilde{f}_n} \widetilde{w}_m(x)\, d\sigma_x = \delta_{mn}$. By using piecewise–constant bases the following expansions are obtained for slave and master scalar potentials:

$$\phi^-(x) = \sum\nolimits_{n\in\mathcal{G}_{\Gamma^-}} w_n(x)\, \phi_n^-, \quad x \in \Gamma^-, \tag{16}$$

$$\phi^+(x) = \sum\nolimits_{m\in\mathcal{G}_{\Gamma^+}} w_m(x)\, \phi_m^+, \quad x \in \Gamma^+, \tag{17}$$

where ϕ_n^-, ϕ_m^+ indicate the slave and master potentials evaluated on primal nodes n, m, respectively. For the normal components of the electric displacement the following expansions hold:

$$D^-(x) = \sum\nolimits_{n\in\mathcal{G}_{\Gamma^-}} \widetilde{w}_n(x)\, \widetilde{d}_n^-, \quad x \in \Gamma^-, \tag{18}$$

$$D^+(x) = \sum\nolimits_{m\in\mathcal{G}_{\Gamma^+}} \widetilde{w}_m(x)\, \widetilde{d}_m^+, \quad x \in \Gamma^+, \tag{19}$$

where $\widetilde{d}_n^-$, $\widetilde{d}_m^+$ are dielectric displacement fluxes through dual faces $\widetilde{f}_n$ and $\widetilde{f}_m$, which are in one–to–one correspondence with nodes n and m, respectively. Note that the unity property provides, e.g., $\int_{\widetilde{f}_n^-} D^-(x)\, d\sigma_x = \sum_{n\in\mathcal{G}_{\Gamma^-}} \int_{\widetilde{f}_n^-} \widetilde{w}_n(x)\, \widetilde{d}_n^-\, d\sigma_x = \widetilde{d}_n^-$, in such a way that the electric displacement flux is exactly reconstructed.

Unlike FEM-based formulations, scalar field continuity across the mortar interface is enforced on a pair of dual grids. Namely, the continuity of scalar potential is weakly enforced over the slave surface, as:

$$\int_{\Gamma^-} w_n(x)\, \left(\phi^-(x) - \phi^+(x)\right)\, d\sigma_x = 0, \quad \forall n \in \mathcal{G}_{\Gamma^-}. \tag{20}$$

By noting that function w_n has compact support, limited to dual cell $\widetilde{f}_n^-$, and by assuming a locally constant potential, previous equation becomes:

$$|\widetilde{f}_n^-|\, \phi_n^- - \sum\nolimits_{m\in\mathcal{G}_{\Gamma^+}} |\widetilde{f}_n^- \cap \widetilde{f}_m^+|\, \phi_m^+ = 0, \tag{21}$$

where $|\widetilde{f}_n^- \cap \widetilde{f}_m^+|$ denotes the area of the intersection between $\widetilde{f}_n^-$, on the slave side, and $\widetilde{f}_m^+$, on the master side.

The continuity of the normal component of the electric displacement is weakly enforced over the master surface, as:

$$\int_{\Gamma^+} \widetilde{w}_m(x)\, \left(D^-(x) + D^+(x)\right)\, d\sigma_x = 0, \quad \forall m \in \mathcal{G}_{\Gamma^+}, \tag{22}$$

which again, by exploiting the unity property, becomes:

$$\widetilde{d}_m^+ + \sum\nolimits_{n\in\mathcal{G}_{\Gamma^-}} \frac{|\widetilde{f}_m^+ \cap \widetilde{f}_n^-|}{|\widetilde{f}_n^-|}\, \widetilde{d}_n^- = 0. \tag{23}$$

mortar projection relationships for primal and dual variables are obtained by writing (21) for any node of the slave side and by writing (23) for any node of the master side. In matrix form these become:

$$\Phi_{\Gamma^-} = \mathbf{P}_\Gamma\, \Phi_{\Gamma^+}, \tag{24}$$

$$\widetilde{\mathbf{d}}_{\Gamma^+} = -\mathbf{P}_\Gamma^{\mathrm{T}}\, \widetilde{\mathbf{d}}_{\Gamma^-}, \tag{25}$$

where $\mathbf{P}_\Gamma = \mathbf{S}_\Gamma^{-1}\mathbf{M}_\Gamma$ is the *mortar projection matrix* mapping master to slave potentials. $\mathbf{S}_\Gamma = (s_i)_{i\in\mathcal{G}_{\Gamma^-}}$ is the diagonal *slave matrix*, with coefficients $s_i = |\widetilde{f}_i^-|$. $\mathbf{M}_\Gamma = (m_{i,j})_{i\in\mathcal{G}_{\Gamma^-}, j\in\mathcal{G}_{\Gamma^+}}$ is the rectangular *master* matrix, with coefficients $m_{i,j} = |\widetilde{f}_i^- \cap \widetilde{f}_j^+|$. The slave matrix yields diagonal because $\widetilde{f}_i^- \cap \widetilde{f}_j^- = \emptyset$, for $i \neq j$. Sparsity of the master matrix is guaranteed by the compact support of primal mortar basis functions. These properties are important for preserving the sparsity of final matrix system, which makes it possible to use an iterative solver. Note that, with standard mortar FEM approaches, $\mathbf{S}_\Gamma$ is non–diagonal, which makes the matrix inversion for constructing $\mathbf{P}_\Gamma$ unfeasible.

Let $\Phi_\Gamma = (\Phi_{\Gamma^-}, \Phi_{\Gamma^+})^{\mathrm{T}}$ and $\widetilde{\mathbf{d}}_\Gamma = \left(\widetilde{\mathbf{d}}_{\Gamma^-}, \widetilde{\mathbf{d}}_{\Gamma^+}\right)^{\mathrm{T}}$ be the interface scalar potentials and fluxes, partitioned into their slave and master terms. mortar projection relationships (24) and (25) can be rewritten as:

$$\Phi_\Gamma = \mathbf{A}_\Gamma\, \Phi_{\Gamma^+}, \tag{26}$$

$$\widetilde{\mathbf{d}}_\Gamma = \mathbf{B}_\Gamma^{\mathrm{T}}\, \widetilde{\mathbf{d}}_{\Gamma^-}, \tag{27}$$

where $\mathbf{A}_\Gamma = \left(\mathbf{P}_\Gamma^{\mathrm{T}}, \mathbf{I}_{N^+}\right)^{\mathrm{T}}$ is the projection matrix from master to all interface variables, with $\mathbf{I}_{N^+}$ identity matrix of size N^+, i.e., the number of master nodes, and $\mathbf{B}_\Gamma = (\mathbf{I}_{N^-}, -\mathbf{P}_\Gamma)$ is the

projection matrix from slave to all interface variables, with N^- number of slave nodes.

From these definitions, it follows that $\mathbf{B}_\Gamma \mathbf{A}_\Gamma = \mathbf{O}_{N^-,N^+}$, with $\mathbf{O}_{N^-,N^+}$ null matrix of size $N^- \times N^+$ or, equivalently, $\widetilde{\mathbf{d}}_\Gamma^{\mathrm{T}} \Phi_\Gamma = 0$ for any $\widetilde{\mathbf{d}}_{\Gamma^-}$, with Φ_{Γ^+} fixed. This relationship corresponds to assuming no free–charge stored at the interface Γ. In fact, by testing $\nabla \cdot \mathbf{D} = \rho$ with the electric scalar potential and by taking a small volume Ω_ε enclosing Γ, with $\mathbf{n}$ outward unit normal of $\partial\Omega_\varepsilon$, one obtains:

$$\int_{\Omega_\varepsilon} \rho(x)\, \Phi(x)\, dx = \int_{\partial\Omega_\varepsilon} \Phi(x)\, \mathbf{D}(x) \cdot \mathbf{n}(x)\, d\sigma_x + \int_{\Omega_\varepsilon} \mathbf{E}(x) \cdot \mathbf{D}(x)\, dx, \quad (28)$$

which, by taking the limit for $\varepsilon \to 0$, provides the *electrostatic energy conservation principle*:

$$\int_{\Gamma^-} \Phi(x)\, \mathbf{D}(x) \cdot \mathbf{n}^-(x)\, d\sigma_x + \int_{\Gamma^+} \Phi(x)\, \mathbf{D}(x) \cdot \mathbf{n}^+(x)\, d\sigma_x = \widetilde{\mathbf{d}}_{\Gamma^-}^{\mathrm{T}} \Phi_{\Gamma^-} + \widetilde{\mathbf{d}}_{\Gamma^+}^{\mathrm{T}} \Phi_{\Gamma^+} = \widetilde{\mathbf{d}}_\Gamma^{\mathrm{T}} \Phi_\Gamma = 0, \quad (29)$$

where are Γ^-, Γ^+ are simply the negative and positive face of the boundary $\partial\Omega_\varepsilon$ for $\varepsilon \to 0$. From matrix identity $\mathbf{B}_\Gamma \mathbf{A}_\Gamma = \mathbf{O}_{N^-,N^+}$, it can be proven that (26) is equivalent to the following mortar interface condition:

$$\mathbf{B}_\Gamma \Phi_\Gamma = \mathbf{0}_{N^-}, \quad (30)$$

where $\mathbf{0}_{N^-}$ is the null vector of size N^-. Interface potentials can be obtained from Φ_Ω by using the volume–to–face incidence matrix, as $\Phi_\Gamma = \widetilde{\mathbf{D}}_{\Omega\Gamma}^{\mathrm{T}} \Phi_\Omega$. By complementing electrostatic matrix system (13) with (30) and by using (27), the final matrix system in saddle–point form is obtained:

$$\begin{pmatrix} -\mathbf{G}_\Omega^{\mathrm{T}} \boldsymbol{\varepsilon}_\Omega \mathbf{G}_\Omega & \widetilde{\mathbf{D}}_{\Omega\Gamma}\, \mathbf{B}_\Gamma^{\mathrm{T}} \\ \mathbf{B}_\Gamma\, \widetilde{\mathbf{D}}_{\Omega\Gamma}^{\mathrm{T}} & \mathbf{O}_{N^-} \end{pmatrix} \begin{pmatrix} \Phi_\Omega \\ \widetilde{\mathbf{d}}_{\Gamma^-} \end{pmatrix} = \begin{pmatrix} \widetilde{\mathbf{q}}_\Omega - \widetilde{\mathbf{D}}_{\Omega\Gamma_0}\, \widetilde{\mathbf{d}}_{\Gamma_0} \\ \mathbf{0}_{N^-} \end{pmatrix}, \quad (31)$$

where $\mathbf{O}_{N^-}$ is the null matrix of size N^- and slave fluxes $\widetilde{\mathbf{d}}_{\Gamma^-}$ can be regarded as Lagrange multipliers, which have a genuine physical interpretation unlike those of classical mortar FEM. (31) has a unique solution after imposing Dirichlet and/or Neumann conditions on the exterior boundary $\partial\Omega$.

An equivalent (explicit) formulation of (31) can be obtained by observing that only master potentials are independent according to (26). By grouping interface potentials Φ_Γ and non–interface potentials $\Phi_{\Omega\setminus\Gamma}$, the following change of variables is obtained:

$$\Phi_\Omega = \mathbf{P}_\Omega\, \Phi'_\Omega, \quad (32)$$

with:

$$\Phi_\Omega = \begin{pmatrix} \Phi_{\Omega\setminus\Gamma} \\ \Phi_\Gamma \end{pmatrix}, \qquad \mathbf{P}_\Omega = \begin{pmatrix} \mathbf{I}_M & \mathbf{O}_{M,N^+} \\ \mathbf{O}_{N,M} & \mathbf{A}_\Gamma \end{pmatrix}, \qquad \Phi'_\Omega = \begin{pmatrix} \Phi_{\Omega\setminus\Gamma} \\ \Phi_{\Gamma^+} \end{pmatrix}, \quad (33)$$

where $\mathbf{I}_M$ is the identity matrix of size M, i.e., the number of non–interface nodes of $\mathcal{G}_\Omega$, $\mathbf{O}_{M,N^+}$ is the null matrix of size $M \times N^+$, and $\mathbf{O}_{N,M}$ is the null matrix of size $N \times M$, with $N = N^- + N^+$ number of interface nodes. From the identity $\mathbf{B}_\Gamma \mathbf{A}_\Gamma = \mathbf{O}_{N^-,N^+}$ and by observing that incidence matrix $\widetilde{\mathbf{D}}_{\Omega\Gamma}$ has non–zero rows only in correspondence of interface nodes, it turns out that $\mathbf{P}_\Omega^{\mathrm{T}}\, \widetilde{\mathbf{D}}_{\Omega\Gamma}\, \mathbf{B}_\Gamma^{\mathrm{T}} = \mathbf{O}_{N',N^-}$, with N' size of the array Φ'_Ω. In such a way, after inserting (32) in (13), the reduced electrostatic matrix system obtained from mortar projection becomes:

$$\mathbf{G}'^{\mathrm{T}}_\Omega\, \boldsymbol{\varepsilon}_\Omega \mathbf{G}'_\Omega\, \Phi'_\Omega = -\mathbf{P}_\Omega^{\mathrm{T}}\, \widetilde{\mathbf{q}}_\Omega + \mathbf{P}_\Omega^{\mathrm{T}}\, \widetilde{\mathbf{D}}_{\Omega\Gamma_0}\, \widetilde{\mathbf{d}}_{\Gamma_0}, \quad (34)$$

where $\mathbf{G}'_\Omega = \mathbf{G}_\Omega\, \mathbf{P}_\Omega$ is the reduced gradient matrix. Again the reduced system can be solved after imposing Dirichlet and/or Neumann conditions on the exterior boundary. It is worth noticing that algebraic properties and sparsity of the original system (13) are preserved in (34), after applying mortar projection. Therefore, very efficient iterative solvers for standard Poisson's problems such as algebraic multigrid, e.g., AGgregation-based algebraic MultiGrid (*AGMG*) proposed in [31], can be used. Note that classical mortar FEM implies a matrix system in saddle–point form, which can be neither easily preconditioned nor efficiently solved.

IV. MORTAR PROJECTION

The specific choice of the nodal basis (14) for the mortar interface, which have a support on the barycentric dual mesh (made of triangles), makes it possible to compute in linear time the master matrix $\mathbf{M}_\Gamma = (m_{i,j})_{i\in\mathcal{G}_{\Gamma^-}, j\in\mathcal{G}_{\Gamma^+}}$, defined in Section III. For any pair of nodes i, j on the slave and master mesh, the intersection $\widetilde{f}_i^- \cap \widetilde{f}_j^+$ needs to be computed when evaluating the coefficient $m_{i,j}$. For dual faces of generic shape (e.g., irregular polygons) this generally requires quadratic–time complexity since, for any face $\widetilde{f}_i^-$, all master faces are spanned, and the intersection between polygons may lead to additional overhead. In the present mortar projection algorithm, polygon intersections can be reduced to triangle intersections since $\widetilde{f}_i^-$, $\widetilde{f}_j^+$, being dual polygons, can be split into barycentric triangles and have a typical star–shaped topology, as it can be noted in Fig. 1. This makes it possible to exploit the intersection algorithm (PANG) proposed by Gander in [32], [33], which provides linear time–complexity for the intersection of non–matching triangle meshes. In such a way, an algorithm with the same complexity for matching polygonal grids can be obtained as well. This is a key feature in order to use poly–MCM for modeling problems of interest in real–life applications.

The PANG algorithm is an advancing front technique which exploits the information of neighboring triangles and assume that any slave triangle intersects an almost constant

Algorithm 1 Triangle Mesh Intersection

$\tau_s \cap \tau_m \neq \emptyset$ ▷ Start from any intersecting pair of slave and master triangles
$L_s \leftarrow \{\tau_s\}$ ▷ Initialize list of slave triangles to be intersected
$L'_m \leftarrow \{\tau_m\}$ ▷ Initialize list of master triangle candidates for intersection
while $L_s \neq \emptyset$ **do**
  $\tau_s \leftarrow L_s[1]$
  $L_s \leftarrow L_s \setminus L_s[1]$
  $L_m \leftarrow L'_m[1]$ ▷ Initialize list of master triangles to be intersected
  $L'_m \leftarrow L'_m \setminus L'_m[1]$
  while $L_m \neq \emptyset$ **do**
    $\tau_m \leftarrow L_m[1]$
    $L_m \leftarrow L_m \setminus L_m[1]$
    if $\tau_s \cap \tau_m \neq \emptyset$ **then** ▷ Compute intersection between triangles
      $m^{\tau}_{i,j} \leftarrow |\tau_s \cap \tau_m|$ ▷ Fill matrix with intersection areas
      $L_m \leftarrow L_m \cup$ **neigh**(τ_m) ▷ Add all triangle neighbors
    end if
  end while
  $L_s \leftarrow L_s \cup$ **intneigh**(τ_s) ▷ Add only intersecting neighbors
  $L'_m \leftarrow L'_m \cup$ **intneigh**(τ_m)
end while

number of master triangles. This ensures the linear complexity of the algorithm, which in this context is used for finding the intersection area matrix $\mathbf{M}^{\tau}_{\Gamma} = (m^{\tau}_{i,j})_{i\in\mathcal{G}_{\Gamma^-}, j\in\mathcal{G}_{\Gamma^+}}$. Any entry of this matrix is the intersection area between the i–th slave triangle and the j–th master triangle so that size of $\mathbf{M}^{\tau}_{\Gamma}$ is $T_s \times T_m$, where T_s and T_m are the number of slave and master triangles of the barycentric subdivisions, respectively. The procedure, which is schematized in Algorithm 1, starts from any pair of intersecting slave τ_s and master τ_m triangles, which is found after direct search. The list of slave triangles to be projected L_s on the master mesh is initialized by τ_s, whereas the list of master triangles that are candidates for the first intersection, L'_m, is initialized by τ_m. The list of master triangles to be intersected L_m is initialized by $L'_m[1]$, i.e., the first element of the master candidate list. Let τ_s be now the current triangle to be projected and τ_m the fist element in the master list, $L_m[1]$. These elements are initially deleted from lists L_s and L_m. For a given τ_s, the intersection process stops when L_m is empty. If an intersection is found, i.e., $\tau_s \cap \tau_m \neq \emptyset$, the area of intersection polygon $m^{\tau}_{i,j}$ is computed and triangles neighboring to τ_m are added to L_m by operator **neigh**$(\cdot)$. Indeed these are natural candidates for intersecting again τ_s. If τ_m does not intersect τ_s, its neighbors are not added to L_m. This ensures that the procedure will end and that search is locally confined. master triangles intersecting τ_s are added as new candidates for L'_m and neighbors of τ_s, which have not been yet treated and intersect master triangles, are added to L_s (operator **intneigh**$(\cdot)$). The PANG algorithm terminates when the list L_s is empty, i.e., there are no further triangles to be projected. The algorithm is implemented in MATLAB® software starting from the code developed and freely distributed by Gander [34]. Particular effort has been spent in making the code more robust and less prone to intersection errors in the case of very fine meshes. Suitable geometric constraints, depending on the mesh size, have been introduced at this purpose.

The master matrix $\mathbf{M}_{\Gamma}$ for polygonal meshes, with coefficients $m_{i,j} = \widetilde{f}_i \cap \widetilde{f}_j$ and of size $N^- \times N^+$, is constructed from the intersection matrix $\mathbf{M}^{\tau}_{\Gamma}$ as follows. Polygonal slave and master mesh $\mathcal{G}_{\Gamma^-}, \mathcal{G}_{\Gamma^+}$ are decomposed into their corresponding barycentric subdivisions. As described in Algorithm 2, any polygonal cell **polygon** of the cell–array **poly** (input mesh) is first ordered counterclockwise (operator $sort(\cdot)$) and then spit into barycentric triangles, which are assembled around any primal vertex of mesh **poly**.

The list of barycentric subdivision nodes **duanod** is made by primal nodes of the original polygonal mesh **nod2d**, by the polygon centroids, and by the edge centroids of the original mesh. Edges of any cell are spanned in order to fill the barycentric node matrix. If a new edge is found the corresponding centroid (operator $centre(\cdot)$) is computed and new dual node is added to **duanod**. The new node index n is stored in array **iedgcentres**, which is suitable for the dual triangle construction. If the edge is already passed, the entry of **iedgcentres** is the dual node index in the list **duanod**. For any polygon vertex i, a pair of triangles τ_i^- (last) and τ_i^+ (next), with apex the polygon centroid, is constructed. These are the barycentric triangles in which the dual face $\widetilde{f}_i$, related to primal node i, is split. The whole construction is schematized in Fig. 4.

Any vertex i is thus attached to a number of barycentric triangles that form the dual face $\widetilde{f}_i$. Two incidence matrices $\mathbf{E}_{\Gamma} = (e_{i,j})$, connecting slave nodes and barycentric triangles of the slave mesh, and $\mathbf{F}_{\Gamma} = (f_{i,j})$, connecting master nodes and barycentric triangles of the master mesh, are build for mapping the intersection matrix to the master matrix as

Algorithm 2 Barycentric Dual Mesh

nodua ← **nod2d** ▷ Initialize the list of barycentric dual mesh nodes with polygon mesh nodes
tridua ← ∅
isnewedg ← **0** ▷ Flag to check if an edge has been already visited
$n \leftarrow$ *size*(**nod2d**, 1)
for $k = 1 \rightarrow$ *length*(**poly**) **do**
  polygon ← **poly**[k]
  polygon ← *sort*(**polygon**) ▷ Sort cell vertexes counterclockwise
  $N \leftarrow$ *length*(**poly**)
  edges ← [**polygon**$(1 \rightarrow N)$, **polygon**$(1 \rightarrow N \pmod N + 1)$]
  nodua ← **nodua** ∪ *centre*(**polygon**) ▷ Insert polygon centre in list of nodes
  icentre ← n
  iedgcentres ← **0**
  for $e = 1 \rightarrow N$ **do**
    edge ← **edges**$(e, \cdot)$
    indglo ← *find*(**edgs**, **edge**)
    edgcentre ← *centre*(**edge**)
    if isnewedg(*indglo*) $= 0$ **then** ▷ Insert edge centroid in the list of nodes
      nodua ← **nodua** ∪ *centre*(**edge**)
      $n \leftarrow n + 1$
      iedgcentres(e) ← n
      isnewedg(*indglo*) ← 1
    else
      iedgcentres(e) ← *find*(**nodua**, **edgcentre**)
    end if
  end for
  for $i = 1 \rightarrow N$ **do**
    $e^- \leftarrow i - 2 \pmod N + 1$, $e^+ \leftarrow i$ ▷ Last and next edge indexes
    $\tau_i^- \leftarrow$ [**iedgcentres**(e^-), **polygon**(i), *icentre*] ▷ Barycentric triangles for i–th node
    $\tau_i^+ \leftarrow$ [**polygon**(i), **iedgcentres**(e^+), *icentre*]
    tridua ← **tridua** ∪ τ_i^- ∪ τ_i^+ ▷ Insert triangles in the mesh
  end for
end for

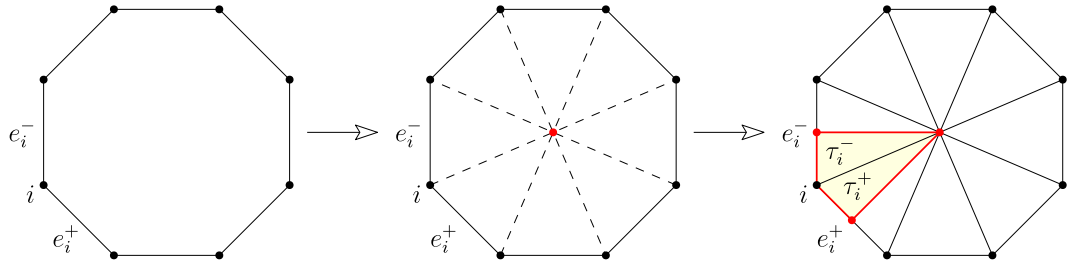

FIGURE 4. Barycentric subdivision: for any vertex i of a polygonal cell a pair of triangles τ_i^-, τ_i^+ is generated.

$\mathbf{M}_\Gamma = \mathbf{E}_\Gamma \mathbf{M}_\Gamma^\tau \mathbf{F}_\Gamma^{\mathrm{T}}$. For instance, coefficient $e_{i,j} = 1$ if the slave node i is attached to the barycentric triangle j, and 0 otherwise. Similar considerations hold for $f_{i,j}$, which define the same kind of connectivity on the master mesh.

The entries of the (diagonal) slave matrix $\mathbf{S}_\Gamma$ are the areas $\widetilde{f}_i$ of the dual polygons on the slave mesh, which can be obtained again from the barycentric dual mesh. The area matrix $\mathbf{S}^\tau = (s_i^\tau)$, where coefficient s_i^τ is the area of the i–th barycentric triangle of the slave mesh, is first defined. By using the same incidence matrix as above, it turns out to be $\mathbf{S}_\Gamma = \mathbf{E}_\Gamma \mathbf{S}_\Gamma^\tau \mathbf{E}_\Gamma^{\mathrm{T}}$. The mortar projection matrix is finally obtained by inverting the slave matrix, i.e., $\mathbf{P}_\Gamma = \mathbf{S}_\Gamma^{-1} \mathbf{M}_\Gamma$.

V. NUMERICAL RESULTS

The mortar cell method was implemented in MATLAB® software environment for matrix computations. Thanks to the CM, the numerical formulation can be written directly in a matrix–oriented style, which is suitable for this environment.

The main advantage is that algebraic operations for sparse matrices in MATLAB® are very efficient, being based on a highly–optimized parallel code.

For generating the polyhedral mesh, *sweeping* and *subgridding* algorithms were implemented. The first one is useful for extruding a polygonal mesh along the vertical direction, thus generating a layered structure made of true polyhedrons [29]. The latter consists in a local mesh refinement technique for hexahedral meshes, which can be useful in practical applications to improve the local accuracy of solution while limiting the increase in the number of DOFs.

In order to speed up assembly time, a Fortran mex–file was implemented for building the poly–CM constitutive matrix, whereas a vectorized MATLAB® code was implemented for building incidence matrices. In contrast, the code for the construction of the mortar projection was implemented in non–vectorized MATLAB® language in order to reduce code complexity and improve code readability although at the expense of computing efficiency. This code consists of three–different steps: generation of the barycentric subdivision with Algorithm 2, mesh intersection with Algorithm 1, and slave and master matrix assembly.

The MCM was validated by considering both analytical and numerical benchmarks. In Section V-A, a 3–D electrostatic problem with closed–form solution is considered in order to assess the MCM convergence and stability properties for the most general case of a master (tetrahedral) mesh and a slave (polyhedral) mesh. In Section V-B, a numerical benchmark consisting in a realistic engineering problem is considered. In that case, the accuracy of MCM analysis (with piecewise–constant basis functions) was verified by considering an equivalent (undecomposed) model solved by a commercial FEM code, implementing second–order tetrahedral elements. Note that eventually all simulations carried out for both benchmarks were run on a standard laptop with 2,9 GHz Intel Core i7 processor, 16 GB RAM memory.

A. UNIT CUBE SPLIT INTO TWO REGIONS

To validate the polyhedral MCM, the analytical benchmark proposed in [35] was adopted. It consists in a unit cube $\Omega = [0, 1]^3$ with unit relative permittivity and no internal charge. Dirichlet boundary conditions are applied over the whole boundary $\partial\Omega$. On the top face $z = 1$ a time–harmonic potential is applied, i.e., $\phi(x, y, z) = 10 \sin(\pi x) \sin(\pi y)$, whereas on the rest of $\partial\Omega$ homogeneous Dirichlet boundary conditions are imposed. By solving Laplace's equation after imposing Dirichlet boundary conditions, the following potential distribution in Ω is obtained:

$$\phi(x, y, z) = \frac{10 \sin(\pi x) \sin(\pi y) \sinh(\sqrt{2}\,\pi z)}{\sinh(\sqrt{2}\,\pi)}. \quad (35)$$

The same Laplace's problem was numerically solved for several polyhedral meshes with different mesh sizes. Ω is split into two halves, i.e., the slave domain $\Omega_s = [0, 1/2] \times [0, 1]^2$ (lower half) and the master domain $\Omega_m = [1/2, 1] \times [0, 1]^2$ (upper half), which are contacting at the interface Γ on $z = 0.5$ horizontal plane. For the sake of comparison, the same problem was also solved by discretizing the whole undecomposed domain Ω with hexahedral or tetrahedral meshes of comparable mesh size. Matrix $\boldsymbol{\varepsilon}_\Omega$ was obtained in that case by using first–order edge element functions (see, e.g., [36] for the implementation of hexahedral elements). Relative errors were estimated by using both energy and L^2 norms over Ω:

$$e_K(\Omega) = \frac{\sqrt{(\mathbf{u} - \mathbf{u}_h)^{\mathrm{T}}\, \mathbf{K}_\Omega\, (\mathbf{u} - \mathbf{u}_h)}}{\sqrt{\mathbf{u}^{\mathrm{T}}\, \mathbf{K}_\Omega\, \mathbf{u}}}, \quad (36)$$

$$e_{L^2}(\Omega) = \frac{\sqrt{\int_\Omega (u(x) - u_h(x))^2\, dx}}{\sqrt{\int_\Omega u(x)^2\, dx}}, \quad (37)$$

where $\mathbf{K}_\Omega = \mathbf{G}'^{\mathrm{T}}_\Omega\, \boldsymbol{\varepsilon}_\Omega\, \mathbf{G}_\Omega$ is the stiffness matrix, $\mathbf{u} = (u(x_i))$ is the array of analytical solution values computed at mesh nodes x_i, and $\mathbf{u}_h = (u_h(x_i))$ is the numerical solution, computed at the same nodes and obtained from (34) by using a mesh grain h. This parameter, which provides a measure of the discretization size for polyhedral, hexahedral, and tetrahedral meshes, was obtained by taking the maximum edge length over the whole mesh of Ω. Note that function u_h was evaluated at a generic point x by interpolating the numerical solution at mesh nodes. Integral in (37) was computed by a Gaussian quadrature 1–point rule, which is enough accurate for a fine mesh, in order to get a reasonable computing cost.

In convergence plots, polyhedral MCM is denoted by the acronym poly–MCM, whereas CM with first–order hexahedral and tetrahedral elements is denoted by hexa–CM and tet–CM, respectively. To assess the optimality of MCM convergence behavior, both linear $\mathcal{O}(h)$, for the energy norm, and quadratic $\mathcal{O}(h^2)$, for the L^2 norm, convergence trends were plotted in dashed line together with convergence curves. The computing complexity of the mortar projection (including barycentric subdivision, Algorithm 1, and matrix assembly) and the computing complexity of the matrix system (34) solution by *AGMG* were assessed by evaluating the CPU time for different mesh sizes. The number of DOFs for the case of the projection algorithm is given by $N^- + N^+$, i.e., the overall number of slave and master nodes, whereas for the system solution it indicates the number of free variables in (34). Note that in all the numerical tests considered, discretizations were treated as generic polyhedral meshes. In such a way, the same data structures (cell–arrays) and numerical algorithms could be used, proving the generality of the proposed approach.

The case of a pure polyhedral mesh contacting with a tetrahedral mesh was considered. Fig. 5 shows the mesh generated in the coarsest case ($h = 0.15$), where 1 152 polyhedra were used to discretize Ω_s and 5 342 tetrahedra were used for Ω_m. Note that in the master domain a subgridding scheme was used for mesh generation, so that cubes of bigger size at the interface $x = 1/2$ in Fig. 5 are indeed 9–face polyhedra. Fig. 6 shows more clearly the non–uniform element spacing in the slave domain. At the mortar interface, for the coarsest mesh, 160 polygons (including 5–node quadrilaterals) were

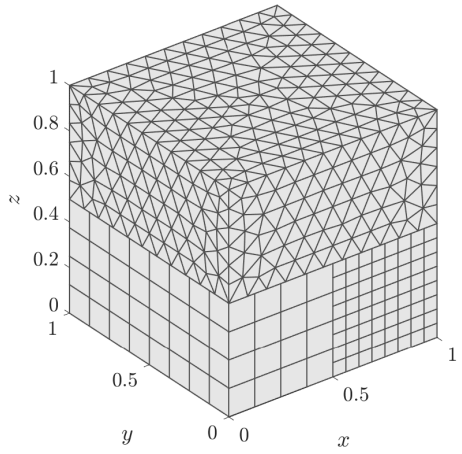

FIGURE 5. Non–conforming tet–poly meshes.

used at the slave side, whereas 278 triangles were used at the master side. Note that for building the mortar projection, the corresponding barycentric subdivisions were considered: 1 296 triangles were used for the slave side, whereas 1 668 triangles were used for the master side. CPU time for building the mortar projection over barycentric meshes was 4.48 s, whereas 10 ms were required by the *AGMG* solver (with 1 661 DOFs). Fig. 7 shows that the convergence rate is linear in the energy norm and quadratic in the L^2 norm. Therefore, optimality is retained in the most general case of a mixed tet–poly grid. For the finest mesh case ($h = 0.0586$, 18 000 polyhedra for Ω_s, and 89,201 tetrahedra for Ω_m) calculation timings are: 67.72 s for the mortar projection, 0.165 s for the *AGMG* solver (with 30 363 DOFs). Note that in this case 1 000 slave polygons (corresponding to 8 040 barycentric triangles) were projected on 1 780 master triangles (split into 10 680 barycentric triangles). The stiffness matrix assembly needs 12.26 s CPU time for a mesh with 107 201 polyhedrons. Fig. 8 shows that, by increasing the number of DOFs, a linear time complexity can be observed for the assembly of both stiffness and projection matrices, and for the iterative solution with *AGMG*.

B. PARALLEL PLATE CAPACITOR WITH AN AIR VOID

As an example of application, derived from the 2–D problem benchmark proposed in [37], the case of a capacitor with an air bubble in the dielectric was considered. Due to the presence of the void with smaller permittivity, the electric field strength locally increases, so that a mesh refinement technique needs to be used in order to capture the electric potential variation around the void. The 3–D model consists in a parallel–plate capacitor (10 V source voltage, square metal plates with 10 mm side, located 7 mm apart) with a spherical inclusion Ω_1 (air with $\varepsilon_1 = 1$ relative permittivity) located at the center of the capacitor. The dielectric (insulating oil) is non–uniform: a 2 mm side cube Ω_2 is with $\varepsilon_2 = 1.5$

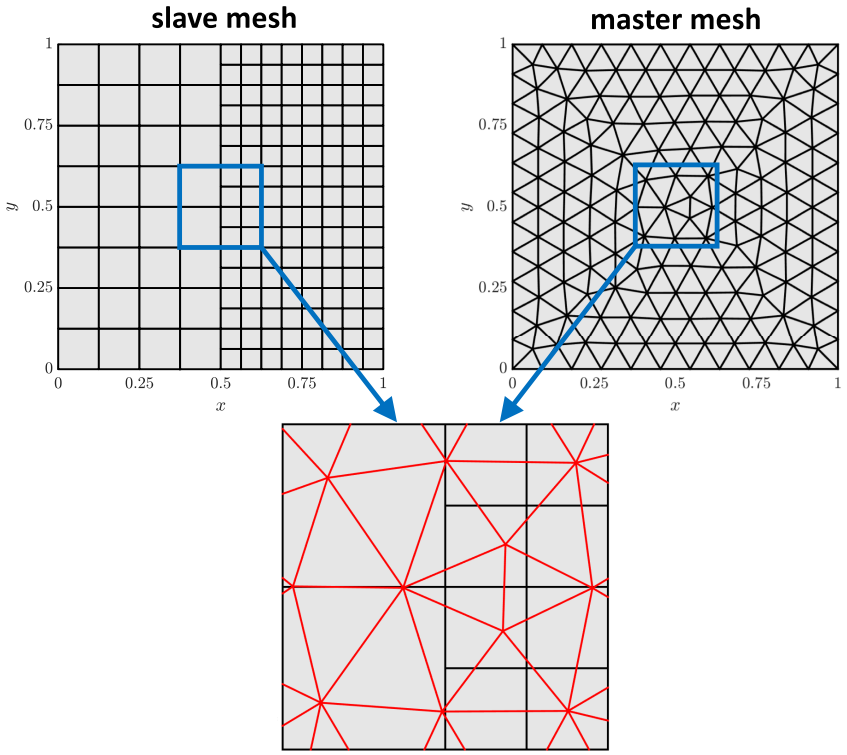

FIGURE 6. Slave and master tet–poly meshes on the mortar interface Γ (a zoom near the square center $x = 0.5$, $y = 0.5$ shows the intersection between master triangles and slave polygons, consisting of both 4-node and 5-node squares).

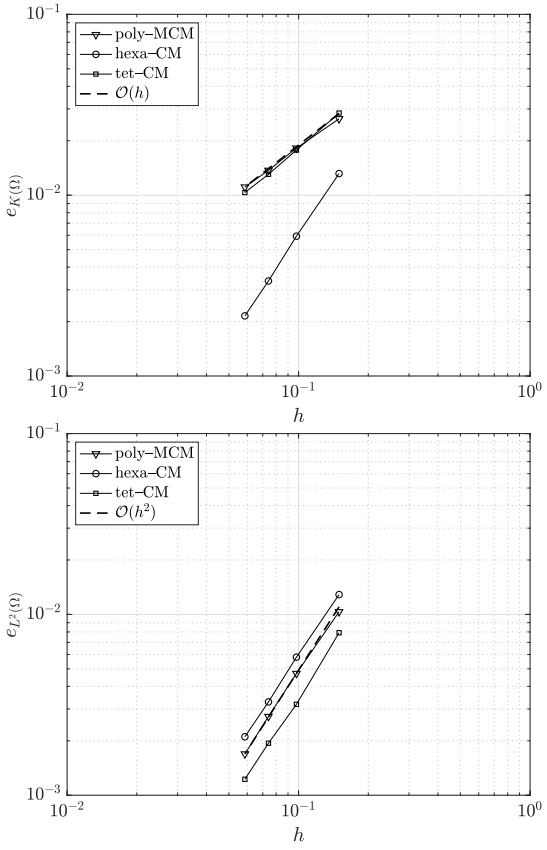

FIGURE 7. Relative energy $e_K(\Omega)$ norm and L^2–norm $e_{L^2}(\Omega)$ errors vs. mesh size h (unit cube Ω discretized by a polyhedral mesh in Ω_s and a tetrahedral mesh in Ω_m).

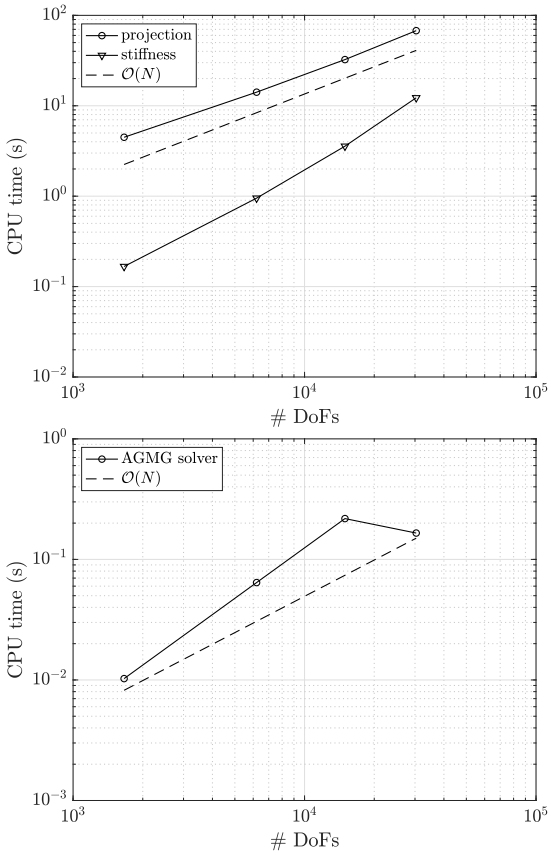

FIGURE 8. CPU time for the assembly of mortar projection and stiffness matrices and for the matrix system solution by *AGMG* solver vs. number of DOFs (tet–poly meshes).

relative permittivity, whereas the rest of the insulating domain Ω_3 is with $\varepsilon_3 = 2.3$ relative permittivity. The effect of the void radius (with $R = 0.25, 0.5, 0.75$ mm) on the local electric field distribution is examined with the poly–MCM.

A Cartesian reference frame (x, y, z) is located at the center of the capacitor; however, due to symmetry, only one eight of the model is considered, i.e., a parallelepiped $\Omega = [0, 5]^2 \times [0, 3.5]$. In such a way, the void Ω_1 is an octant centered at the origin, $\Omega_2 = [0, 2]^3 \setminus \Omega_1$, and $\Omega_3 = \Omega \setminus \Omega_2$. The top electrode (at 5 V potential) is located at $z = 3.5$ mm, whereas the symmetry plane (at $z = 0$) is at ground potential. Neumann BCs are applied to the other boundary sides of Ω.

According to the modeling scheme proposed in the Introduction: 1) Any domain was first simulated independently in order to identify the element type and the mesh size best suited for ensuring a local field reconstruction; 2) Different parts of the model were then reassembled and simulated by using the poly–MCM.

Fig. 9 provides a conceptual sketch of the modeling strategy which can adopted thanks to the poly–mortar approach. It is clearly shown that only the master domain, which include the void domain of varying radius, has to be remeshed. Modeling steps are discussed in detail as follows:

1) A model of the capacitor *without air void*, in which the computational domain is made up of $\Omega_2^* = [0, 2]^3$ (with permittivity ε_2) and Ω_3 (with the same permittivity as above, ε_3), was first simulated. The model is fully 3–D because of the dielectric inhomogeneity. In order to reduce the number of DOFs, a structured grid was used for both subdomains. Because $\varepsilon_2 < \varepsilon_3$ the electric field strength locally increases. Therefore, a refinement of the hexahedral mesh was carried out via subgridding (mesh size in Ω_2^* is a half of that one in Ω_3). The resulting mesh turned out to be a genuine polyhedral mesh because of transition elements interfacing these subdomains. The polyhedral mesh refinement criterion was based on the discrepancy between a reference capacitance value (obtained with 2^{nd} ord. FEM) and that one extracted from the electrostatic energy computed with poly–CM, i.e.,

$$E_\Omega = \frac{1}{2}\, \Phi_\Omega^{\mathrm{T}}\, \mathbf{G}_\Omega^{\mathrm{T}} \boldsymbol{\varepsilon}_\Omega \mathbf{G}_\Omega\, \Phi_\Omega. \tag{38}$$

Table. 1 shows the computational requirements for assembling the constitutive matrix and the relative discrepancy $e_C = |C_{poly} - C_{FEM}|/C_{FEM}$ of the poly–CM capacitance value C_{poly} from 2^{nd} ord. FEM capacitance value C_{FEM}. Mesh sizes were chosen in order to generate a cubical mesh: e.g., with $h = 0.5$ mm, cubes of 0.5 mm side discretize Ω_3. The mesh refinement $h = 0.1$ mm, with 143 500 polyhedrons, was finally adopted for discretizing the capacitor with

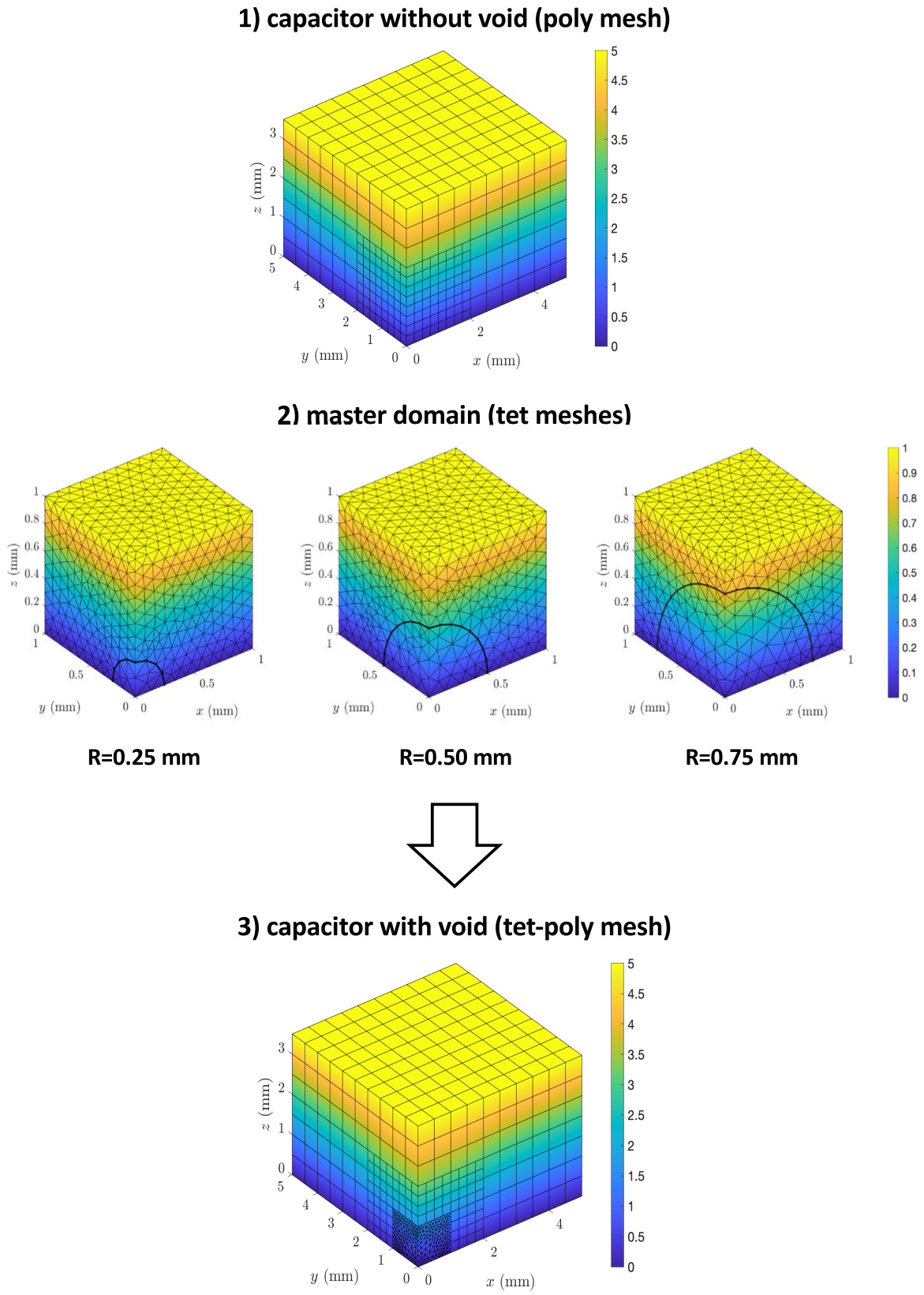

FIGURE 9. Modeling steps of the polyhedral mortar method (poly–MCM): 1) the capacitor without void is meshed with subgridding (no–remeshing needed); 2) the master domain Ω_m is meshed with tetrahedrons (remeshing required because void radius varies); 3) Ω_m is inserted in the mesh of subproblem 1) to solve the original problem, poly–tet meshes are glued together.

TABLE 1. CPU time for the constitutive matrix assembly and discrepancy value e_C for any polyhedral mesh size h.

h (mm)	# poly	CPU time (s) (assembly)	e_C (%)
0.5	1 148	0.07	$4.04 \cdot 10^{-3}$
0.25	9 184	0.66	$1.31 \cdot 10^{-3}$
0.125	73 472	13.54	$3.49 \cdot 10^{-4}$
0.1	143 500	43.78	$2.06 \cdot 10^{-4}$

void model. Such a test was useful in order to asses the performance of poly–MCM software when facing real–sized problems typical of the engineering practice.

2) Three different values of the void radius were considered in the benchmark. Therefore, by using a standard discretization approach such as FEM or CM, one needs a remeshing of the whole domain Ω every time that a geometric parameter is changed. The structured mesh cannot be, however, extended to the void domain Ω_1, which is spherical. In order to avoid remeshing a master domain $\Omega_m = [0, 1]^3$, enclosing region Ω_1, was thus defined. Ω_m was discretized by tetrahedral elements to cope with the curved boundary. A local electrostatic problem was then considered in Ω_m, i.e., a arbitrary potential (e.g., unit) was applied on the top of Ω_m and a ground potential on the bottom, whereas homogeneous Neumann BCs were applied on the rest of the boundary. In such a way, the electric field locally approximates the true field distribution, i.e, that one computed on the full problem (with the presence of the void). This time the mesh was refined until a good agreement with 2^{nd} ord. FEM. electric field strength distribution was achieved. The final mesh refinement (with size $h = 0.07$ mm) consisted of 398 744 tets, 69 227 vertexes, e.g., for the model with void radius $R = 0.25$ mm. Similar meshes were obtained for the other models ($R = 0.50, 0.75$ mm). Solution accuracy on a local scale was verified by considering the electric field distribution along the vertical axis (segment $x = y = 0, z = [0, 1]$ mm). Fig. 10 shows that the electric field strength, reconstructed by piecewise–constant bases defined on polyhedrons, is in very good agreement with 2^{nd} ord. FEM.

3) The structured mesh defined at step 1) was finally glued together with the tetrahedral mesh defined at step 2). Part of the cubical mesh was carved out from Ω_2^*, i.e., the subgridding region, in order to insert the master domain Ω_m. This merging process could be carried out without need of remeshing the entire slave domain $\Omega_s = \Omega \setminus \Omega_m$, simplifying the model pre–processing. Ω_s was discretized with 135 500 hexahedra. The same tetrahedral meshes defined in 2) were used for Ω_m.

Unlike previous test cases, the capacitor model discretization shows a mortar interface Γ—i.e., the inner boundary of the smaller cube in Fig. 9, made by three plane faces—which is fully 3–D and is with corners. The intersection

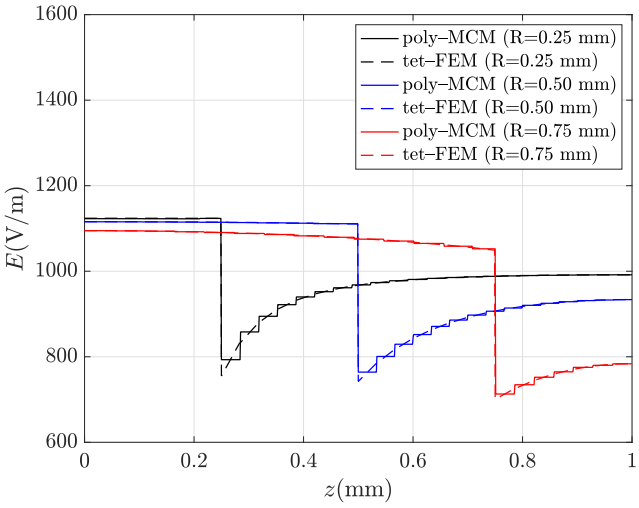

FIGURE 10. Electric field magnitude for poly–MCM and 2^{nd} ord. FEM along the z–axis for different values of the void radius R (master domain only).

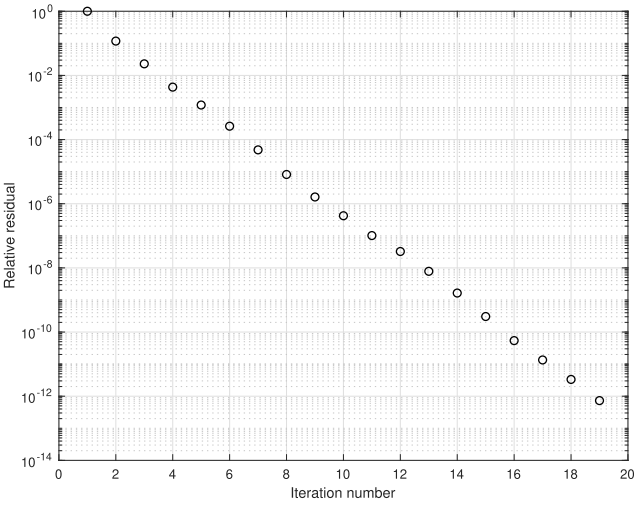

FIGURE 11. Convergence path of the *AGMG* solver for poly–MCM.

algorithm, outlined in Section IV, was applied in this more general context first to any side of the cube (by using a local reference frame projection to work with 2–D coordinates) and slave and master matrices $\mathbf{S}_\Gamma$, $\mathbf{M}_\Gamma$, defined in Section III, were assembled over the whole interface Γ. The particularly simple definition of primal and dual mortar basis functions, which is based on purely geometrical consideration, made it possible to handle also manifold with corners, which is not common for standard mortar FEM formulations. For the case $R = 0.25$ mm, the generation of the mortar projection required 17.60 s with a slave mesh on Γ (piecewise planar) with 15 916 triangles and a master mesh on Γ with 1 200 squares. Similar results were found with other radius values.

The constitutive matrix, e.g., for the case with $R = 0.25$ mm, turned out to be of size 826 479 and it was assembled in 295 s. The resulting matrix system (34), with 195 819 DOFs, was solved by *AGMG* iterative solver in 1.36 s.

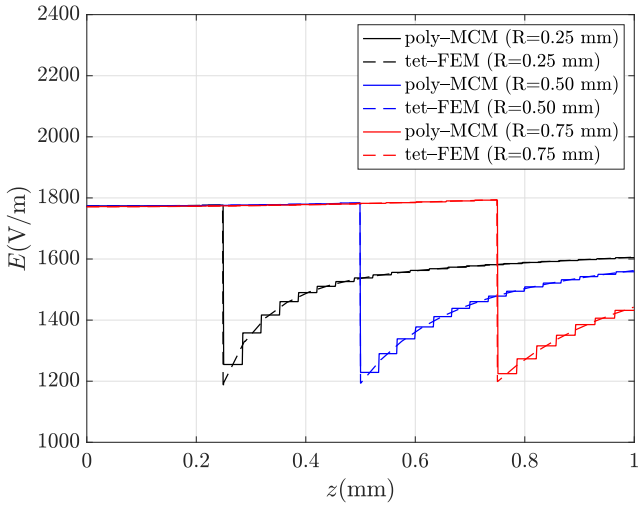

FIGURE 12. Electric field magnitude for poly–MCM and 2^{nd} ord. FEM along the z–axis for different values of the void radius R (full capacitor model with void).

Comparable constitutive matrix sizes and timings were found for the other cases with $R = 0.50, 0.70$ mm. This shows that solution time is negligible even for a large–size problem, proving that the proposed formulation leads to a linear system which can be solved very efficiently. Fig. 11 shows that solver iterations are very few and the convergence path at the fixed tolerance of 10^{-12} is linear. Fig. 12 shows the electric field strength profile along a vertical segment as above for the original problem, i.e., the capacitor with void. It can be noted that the same plot behaviors of Fig. 10 are found and the agreement with 2^{nd} ord. FEM profiles is still very good.

VI. CONCLUSION

In this work a mortar method over general polyhedral discretizations has been presented. The poly–MCM is based on a cell method variant, based on augmented dual grids, which enjoys many appealing properties: i) Matrix assembly is completely Jacobian-free and does not require Gaussian integration; ii) Novel mortar basis functions are defined at the interface by exploiting dual grids, which are at the core of the CM framework. The key feature is that mortar basis functions have their supports defined on dual mesh polygonal faces, which makes it possible to provide a diagonal slave matrix and a sparse master matrix. In such a way, the mortar projection operator can be explicitly obtained and Lagrange multipliers can be eliminated from the final saddle–point system. Moreover, the use of dual mortar bases make it possible to highly reduce the complexity of the intersection between slave and mortar interfaces, even though these are discretized with general polygonal grids. The integration over intersection elements—which are of complex polygonal shape and need to be mapped to the master element—are not required. This is very important in order to reduce the complexity of the assembly procedure of the mortar map and is again due to the peculiar shape of the support chosen for basis functions. By using a barycentric subdivision, which enjoys a natural splitting into a simplicial mesh, it has been possible to use a linear–time algorithm for finding intersections of triangle meshes. Numerical experiments show that the convergence properties of poly–MCM are the same of classical CM for hexahedral and tetrahedral meshes without domain partition.

An example of application, a real–size model discretized into hundred thousands polyhedral cells has been simulated by poly–MCM. It is shown that re-meshing can be avoided when different subproblems are to be simulated and the mortar coupling between slave and master domains can be realized in reasonable computing time even in the complex case of a fully 3–D mortar interface with corners.

APPENDIX
CONSTITUTIVE MATRIX ASSEMBLY

The construction of the constitutive matrix **mat**, obtained by using CM discretization, with piecewise–constant basis functions, is described in Algorithm 3. Data structures passed to the MATLAB® function implementing the assembly process are the coordinate matrix **nod3d**, the edge matrix **edgs** (in which any row is a pair of vertex indexes), and cell–arrays of faces **facs** and polyhedra **cells**. Last data structures are stored in a particular sparse format which is suitable for polygonal and polyhedral meshes. For instance, **facs** is stored by using a pointer matrix **ifacmat** of size $N_f \times 2$, with N_f number of faces, and the edge index array **vfacs**. Any row of **ifacmat** is made up by a pointer, which provides the polygonal face location inside **vfacs**, and by the face length, i.e., the number of polygon edges. Faces, i.e., any face is a list of edge indexes, are stored column–wise inside the array **vfacs**. The cell–array **cells** is stored in a similar way, by using a pair of data structures. For any polyhedron the local mass matrix, to be assembled in **mat**, is constructed as follows. For any face f of the polyhedron **cell**, it is checked if any edge e of f has already been spanned. If the edge is passed for the first time then it is inserted in the local edge matrix **edges**, and local and global edge indices are stored in arrays **iedgloc**, **iedgs**, respectively. The matrix **iedgfac** of size $n \times 2$, where n is the number of edges for the polyhedron, provides the pair of incident faces for any edge. The first entry of any row is thus filled when the edge is spanned for the first time. Geometric data structures for building edge vector bases according to expression (6) are obtained by using operators $vol(\cdot)$ and $centre(\cdot)$ which provide the area / volume and the centroid of a polygon or a polyhedron according to the algorithms proposed in [38]. These are useful for reducing the computing complexity since volume calculation are reduced to surface ones by exploiting the divergence theorem. In order to calculate basis functions, for any edge the corresponding edge vector $\mathbf{e}$, dual face area vector $\widetilde{\mathbf{f}}_e$, and the volume of its uniformity region **supp**, as shown in Fig. 2, are calculated. Edge functions $\mathbf{w}_e$ are computed by function $basis(\cdot)$, which implements (6), and are stored as $n \times 3$ size matrices, in which for any k–th row a constant vector (related uniformity region of k–th edge) is allocated. Local mass matrix coefficient

Algorithm 3 Constitutive Matrix for Poly–MCM

for $c = 1 \rightarrow$ *length*(**cells**) **do**
  $\varepsilon \leftarrow$ **parameters**(c) ▷ Material parameter (constant inside the cell)
  cell $\leftarrow$ **cells**$[c]$ ▷ Any polyhedron is a list of faces
  for $f = 1 \rightarrow$ *length*(**cell**) **do**
    face $\leftarrow$ **cell**$[f]$
    $n \leftarrow 0$, **iedgloloc** $\leftarrow$ **0**
    for $e = 1 \rightarrow$ *length*(**face**) **do** ▷ Find the pair of faces incident to any polyhedron edge
      edge $\leftarrow$ **edgs**[**face**(e)]
      if iedgloc(**face**$(e)) = 0$ **then** ▷ Check if the edge is not already in the list
        $n \leftarrow n + 1$
        iedgloc(**face**$(e)) \leftarrow n$ ▷ Local edge indexing
        iedgs$(n) \leftarrow$ **face**(e) ▷ Global edge indexing
        iedgfac$(1, n) \leftarrow f$ ▷ First incident face added
        edges$(n, \cdot) \leftarrow$ **edge** ▷ Add the new edge to the list
      else
        iedgfac$(2,$ **iedgloc**(**face**$(e)))) \leftarrow f$ ▷ Second incident face added
      end if
    end for
  end for
  $\mathbf{c}_v \leftarrow$ *centre*(**cell**)
  for $e = 1 \rightarrow n$ **do** ▷ Geometric data structures for basis functions
    edge $\leftarrow$ **edges**$(e, \cdot)$
    {**face**(1), **face**(2)} $\leftarrow$ **iedgfac**$(\cdot, e)$
    $\mathbf{c}_{f1} \leftarrow$ *centre*(**face**(1))
    $\mathbf{c}_{f2} \leftarrow$ *centre*(**face**(2))
    $\mathbf{c}_e \leftarrow$ *centre*(**edge**)
    $\mathbf{e}[e] \leftarrow$ **nod3d**(**edge**$(2), \cdot)$ − **nod3d**(**edge**$(1), \cdot)$
    $\widetilde{\mathbf{f}}_e[e] \leftarrow \frac{1}{2}(\mathbf{c}_v - \mathbf{c}_e) \times (\mathbf{c}_{f1} - \mathbf{c}_{f2})$
    supp$(e) \leftarrow \frac{1}{3}\mathbf{e} \cdot \widetilde{\mathbf{f}}_e$
  end for
  for $e, e' = 1 \rightarrow n$ **do**
    $\mathbf{w}_e \leftarrow basis(\mathbf{e}, \widetilde{\mathbf{f}}_e, vol($**cell**$))$ ▷ Build edge basis functions according to (6)
    $\mathbf{w}_{e'} \leftarrow basis(\mathbf{e}', \widetilde{\mathbf{f}}_{e'}, vol($**cell**$))$
    $ie \leftarrow$ **iedgs**(e)
    $ie' \leftarrow$ **iedgs**(e')
    mat$(ie, ie') \leftarrow$ **mat**$(ie, ie') + \sum_{k=1}^{n} \varepsilon$ **supp**$(k)\, \mathbf{w}_e(k, \cdot) \cdot \mathbf{w}_{e'}(k, \cdot)$ ▷ Matrix assembly for any edge pair
  end for
end for

are evaluated according to (11), without need of evaluating Jacobian, i.e., no isoparametric mapping like FEM, and then assembled into **mat**.

REFERENCES

[1] A. Toselli and O. Widlund, *Domain Decomposition Methods—Algorithms Theory*. Berlin, Germany: Springer, 2005.
[2] J. Xie and M. Swaminathan, "Electrical–thermal cosimulation with nonconformal domain decomposition method for multiscale 3-D integrated systems," *IEEE Trans. Compon., Packag., Manuf. Technol.*, vol. 4, no. 4, pp. 588–601, Apr. 2014.
[3] J. Huang, "A mortar element method for elliptic problems with discontinuous coefficients," *IMA J. Numer. Anal.*, vol. 22, no. 4, pp. 549–576, Oct. 2002.
[4] Y. Takahashi, K. Fujiwara, T. Iwashita, and H. Nakashima, "Parallel finite-element method based on space–time domain decomposition for magnetic field analysis of electric machines," *IEEE Trans. Magn.*, vol. 55, no. 6, pp. 1–4, Jun. 2019.
[5] A. Lehikoinen, J. Ikaheimo, A. Arkkio, and A. Belahcen, "Domain decomposition approach for efficient time-domain finite-element computation of winding losses in electrical machines," *IEEE Trans. Magn.*, vol. 53, no. 5, pp. 1–9, May 2017.
[6] M. Kaltenbacher, *Numerical Simulation of Mechatronic Sensors and Actuators*. Berlin, Germany: Springer, 2007.
[7] C. Bernardi, Y. Maday, and A. T. Patera, "Domain decomposition by the mortar element method," in *Asymptotic and Numerical Methods for Partial Differential Equations With Critical Parameters*, vol. 384. H. G. Kaper, M. Garbey, and G. W. Pieper, Eds. Amsterdam, The Netherlands: Springer, 1993, pp. 269–286.
[8] C. Lacour and Y. Maday, "Two different approaches for matching nonconforming grids: The mortar element method and the feti method," *BIT Numer. Math.*, vol. 37, no. 3, pp. 720–738, Sep. 1997.
[9] F. Ben Belgacem and Y. Maday, "The mortar element method for three dimensional finite elements," *ESAIM, Math. Model. Numer. Anal.*, vol. 31, no. 2, pp. 289–302, 1997.

[10] C. Kim, R. D. Lazarov, J. E. Pasciak, and P. S. Vassilevski, "Multiplier spaces for the mortar finite element method in three dimensions," *SIAM J. Numer. Anal.*, vol. 39, no. 2, pp. 519–538, Jan. 2001.
[11] M. Benzi, G. H. Golub, and J. Liesen, "Numerical solution of saddle point problems," *Acta Numerica*, vol. 14, pp. 1–137, May 2005.
[12] B. I. Wohlmuth, "A comparison of dual Lagrange multiplier spaces for mortar finite element discretizations," *ESAIM, Math. Model. Numer. Anal.*, vol. 36, no. 6, pp. 995–1012, Nov. 2002.
[13] B. I. Wohlmuth, "A mortar finite element method using dual spaces for the Lagrange multiplier," *SIAM J. Numer. Anal.*, vol. 38, no. 3, pp. 989–1012, Jan. 2000.
[14] E. Tonti, *The Mathematical Structure of Classical and Relativistic Physics: A General Classification Diagram*. Basel, Switzerland: Birkhäuser, 2013.
[15] L. Codecasa, R. Specogna, and F. Trevisan, "Base functions and discrete constitutive relations for staggered polyhedral grids," *Comput. Methods Appl. Mech. Eng.*, vol. 198, nos. 9–12, pp. 1117–1123, Feb. 2009.
[16] E. Tonti, "Why starting from differential equations for computational physics?" *J. Comput. Phys.*, vol. 257, pp. 1260–1290, Jan. 2014.
[17] P. Milbradt and T. Pick, "Polytope finite elements," *Int. J. Numer. Methods Eng.*, vol. 73, no. 12, pp. 1811–1835, 2008.
[18] F. Brezzi, A. Buffa, and K. Lipnikov, "Mimetic finite differences for elliptic problems," *ESAIM, Math. Model. Numer. Anal.*, vol. 43, no. 2, pp. 277–295, Mar. 2009.
[19] L. Beirão da Veiga, C. Lovadina, and D. Mora, "A virtual element method for elastic and inelastic problems on polytope meshes," *Comput. Methods Appl. Mech. Eng.*, vol. 295, pp. 327–346, Oct. 2015.
[20] F. Moro, P. Alotto, F. Freschi, and M. Guarnieri, "A cell method formulation of 3-D electrothermomechanical contact problems with mortar discretization," *IEEE Trans. Magn.*, vol. 48, no. 2, pp. 503–506, Feb. 2012.
[21] P. Alotto, M. Guarnieri, and F. Moro, "A mortar cell method for electro-thermal contact problems," *IEEE Trans. Magn.*, vol. 49, no. 2, pp. 795–798, Feb. 2013.
[22] F. Moro, P. Alotto, M. Guarnieri, and A. Stella, "Domain decomposition with the mortar cell method," *Int. J. Numer. Modelling, Electron. Netw., Devices Fields*, vol. 27, no. 3, pp. 461–471, May 2014.
[23] F. Moro and M. Guarnieri, "Efficient 3-D domain decomposition with dual basis functions," *IEEE Trans. Magn.*, vol. 51, no. 3, pp. 1–4, Mar. 2015.
[24] P. Alotto, G. Gruosso, F. Moro, and M. Repetto, "A boundary integral formulation for eddy current problems based on the cell method," *IEEE Trans. Magn.*, vol. 44, no. 6, pp. 770–773, Jun. 2008.
[25] F. Moro and L. Codecasa, "Indirect coupling of the cell method and BEM for solving 3-D unbounded magnetostatic problems," *IEEE Trans. Magn.*, vol. 52, no. 3, pp. 1–4, Mar. 2016.
[26] F. Moro and L. Codecasa, "A 3-D hybrid cell boundary element method for time-harmonic eddy current problems on multiply connected domains," *IEEE Trans. Magn.*, vol. 55, no. 3, pp. 1–11, Mar. 2019.
[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 and L. Codecasa, "A 3-D hybrid cell method for induction heating problems," *IEEE Trans. Magn.*, vol. 53, no. 6, pp. 1–4, Jun. 2017.
[29] F. Moro, D. Desideri, A. Doria, A. Maschio, C. Medé, and L. Codecasa, "A face-smoothed cell method for static and dynamic piezoelectric coupled problems on polyhedral meshes," *J. Comput. Phys.*, vol. 386, pp. 84–109, Jun. 2019.
[30] L. Codecasa, "Refoundation of the cell method using augmented dual grids," *IEEE Trans. Magn.*, vol. 50, no. 2, pp. 497–500, Feb. 2014.
[31] A. Napov and Y. Notay, "An algebraic multigrid method with guaranteed convergence rate," *SIAM J. Sci. Comput.*, vol. 34, no. 2, pp. A1079–A1109, Jan. 2012.
[32] M. J. Gander and C. Japhet, "An algorithm for non–matching grid projections with linear complexity," in *Domain Decomposition Methods in Science and Engineering XVIII* (Lecture Notes in Computational Science and Engineering). J. Fagerberg, D. C. Mowery, and R. R. Nelson, Eds. Berlin, Germany: Springer, 2009, pp. 1–8.
[33] M. J. Gander and C. Japhet, "Algorithm 932: PANG: Software for non-matching grid projections in 2D and 3D with linear complexity," *ACM Trans. Math. Softw.*, vol. 40, no. 1, pp. 1–25, Sep. 2013.
[34] M. J. Gander. (2008). *Triangle Mesh Intersection Algorithm*. Accessed: Nov. 3, 2008. [Online]. Available: http://www.unige.ch/ gander/demo.php
[35] N. Z. Lima and R. C. Mesquita, "Face-based gradient smoothing point interpolation method applied to 3-D electromagnetics," *IEEE Trans. Magn.*, vol. 50, no. 2, pp. 537–540, Feb. 2014.
[36] P. Dular, A. Genon, J.-Y. Hody, W. Legros, J. Mauhin, and A. Nicolet, "Coupling between edge finite elements, nodal finite elements and boundary elements for the calculation of 3-D eddy currents," *IEEE Trans. Magn.*, vol. 29, no. 2, pp. 1470–1474, Mar. 1993.
[37] W. Xu, G. Wang, N. Duan, S. Wang, Y. Guo, and J. Zhu, "Extended finite-element method for weak discontinuities in electric fields," *IEEE Trans. Appl. Supercond.*, vol. 26, no. 7, pp. 1–5, Oct. 2016.
[38] Z. J. Wang, "Improved formulation for geometric properties of arbitrary polyhedra," *AIAA J.*, vol. 37, pp. 1326–1327, Jan. 1999.

FEDERICO MORO (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 has been a Visiting Student with the Department of Physics, Swansea University, 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 working as an Associate Professor, since 2020. He is the author of more than 100 articles 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 and Electrimacs 2019 conferences.

LORENZO CODECASA (Member, IEEE) received the Ph.D. degree in electronic engineering from the Politecnico di Milano, in 2001.

From 2002 to 2010, he worked as an Assistant Professor of electrical engineering with the Department of Electronics, Information, and Bioengineering, Politecnico di Milano, where he has worked as an Associate Professor of electrical engineering, since 2010. In his research on heat transfer and thermal management of electronic components, he has introduced original industrial-strength approaches to the extraction of compact thermal models, currently available in market leading commercial software. His main research interest includes the theoretical analysis and the computational investigation of electric circuits and electromagnetic fields. In his research areas, he has authored or coauthored over 200 papers in refereed international journals and conference proceedings. For these activities, in 2016, he received the Harvey Rosten Award for Excellence. He has been serving as the Chair for the conference THERMal INvestigation of Integrated Circuits (THERMINIC). He has also been serving as an Associate Editor for the IEEE TRANSACTIONS OF COMPONENTS, PACKAGING AND MANUFACTURING TECHNOLOGY.