

An a posteriori-driven adaptive mixed high-order method with application to electrostatics

Daniele Antonio Di Pietro, Ruben Specogna

▶ **To cite this version:**

Daniele Antonio Di Pietro, Ruben Specogna. An a posteriori-driven adaptive mixed high-order method with application to electrostatics. Journal of Computational Physics, 2016, 326, pp.35-55. 10.1016/j.jcp.2016.08.041 . hal-01310313v2

HAL Id: hal-01310313
https://hal.science/hal-01310313v2

Submitted on 22 Sep 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

An a posteriori-driven adaptive Mixed High-Order method with application to electrostatics[☆]

Daniele A. Di Pietro

University of Montpellier, Institut Montpelliérain Alexander Grothendieck, 34095 Montpellier, France

Ruben Specogna

University of Udine, Polytechnic Department of Engineering and Architecture, 33100 Udine, Italy

Abstract

In this work we propose an adaptive version of the recently introduced Mixed High-Order method and showcase its performance on a comprehensive set of academic and industrial problems in computational electromagnetism. The latter include, in particular, the numerical modeling of comb-drive and MEMS devices. Mesh adaptation is driven by newly derived, residual-based error estimators. The resulting method has several advantageous features: It supports fairly general meshes, it enables arbitrary approximation orders, and has a moderate computational cost thanks to hybridization and static condensation. The a posteriori-driven mesh refinement is shown to significantly enhance the performance on problems featuring singular solutions, allowing to fully exploit the high-order of approximation.

Keywords: Electrostatics, computational electromagnetism, Mixed High-Order methods, Hybrid High-Order methods, Discrete Geometric Approach, discontinuous skeletal methods, variable diffusion, a posteriori error analysis

2010 MSC: 65N08, 65N30, 65N12, 78A30

1. Introduction

Let $\Omega \subset \mathbb{R}^d$, $d \geqslant 1$, denote a bounded connected polyhedral domain of boundary $\partial\Omega$, f a square-integrable volumetric source term, and $\boldsymbol{K}$ a bounded tensor-valued diffusion coefficient piecewise constant on a polyhedral partition P_Ω of Ω, and with eigenvalues in the interval $[\underline{K}, \overline{K}]$, $0 < \underline{K} \leqslant \overline{K} < +\infty$. We consider the problem that consists in seeking a vector-valued field $\boldsymbol{\sigma} : \Omega \to \mathbb{R}^d$ and a scalar-valued field

[☆]This work was supported by Agence Nationale de la Recherche project HHOMM (ANR-15-CE40-0005)

Email addresses: daniele.di-pietro@umontpellier.fr (Daniele A. Di Pietro), ruben.specogna@uniud.it (Ruben Specogna)

$u : \Omega \to \mathbb{R}$ such that

$$\boldsymbol{K}^{-1}\boldsymbol{\sigma} - \boldsymbol{\nabla} u = 0 \qquad \text{in } \Omega, \tag{1a}$$

$$-\boldsymbol{\nabla}\cdot\boldsymbol{\sigma} = f \qquad \text{in } \Omega, \tag{1b}$$

$$u = 0 \qquad \text{on } \partial\Omega. \tag{1c}$$

Our goal is to develop an a posteriori-driven, adaptive version of the Mixed High-Order method of [1] for the numerical approximation of problem (1); cf. also [2] for more general diffusion coefficients and boundary conditions. The performance of the proposed method is thoroughly demonstrated on a comprehensive set of academic and industrial test cases.

Problems of the form (1) arise in a variety of fields, ranging from porous media flows to heat transfer and electrostatics. The latter application provides the original motivation for this work. In this context, the variable u represents the electric scalar potential, whereas $\boldsymbol{e} = -\boldsymbol{\nabla} u$ is the electric field. Since $\boldsymbol{e}$ is conservative by assumption, the harmonic field component is assumed to be zero. The inverse of the diffusion coefficient, $\boldsymbol{K}^{-1}$, represents the electrical permittivity, usually denoted by $\boldsymbol{\varepsilon}$. The vector variable $\boldsymbol{\sigma}$ represents the electric displacement vector field, often denoted by $\boldsymbol{d}$, whereas f is a known volumetric source charge density. Thus, (1b) represents Gauss's law, whereas (1a) results from the composition of the constitutive law $\boldsymbol{d} = \boldsymbol{\varepsilon}\boldsymbol{e}$ and the definition of the scalar potential u.

In electrostatics, the solution of the scalar pure diffusion problem (1) plays a fundamental role, e.g., in the extraction of parasitic circuit parameters in integrated circuits [3, 4], in the solution of forward problems in Electrical Capacitance Tomography [5], and in the optimization of epoxy spacers in high-voltage direct current transmission lines [6]. Moreover, multi-physics problems involving electrostatics coupled with elastostatics or with the Schrödinger problem are fundamental to simulate the behaviour of micro electro mechanical systems (MEMS) [7] and nano-electronic devices [8]. Solving the Poisson–Boltzmann electrostatic problem is used to produce an implicit solvation to speed up molecular dynamics simulations, see [9, 10]. Finally, the electrostatic problem is similar to the steady-state current conduction problem which has applications, e.g., to the solution of forward problems in Electrical Impedance Tomography [11] and to the computation of stationary 3D halo currents in fusion devices [12]. In all of the above examples, the exact solution typically contains singularities due to both the geometry and the problem data. Disposing of efficient mesh adaptation strategies is thus paramount to benefit from the use of high-order methods.

Several classical methods have been used to numerically solve problem (1) or variations thereof in the context of computational electromagnetism. A (by far) non-exhaustive list includes, in particular, [13, 14, 15, 16, 4, 17, 18] and the high-order method of [19]. Such methods are limited to standard element shapes and, in most cases, do not support nonconforming mesh refinement. Other methods exist, on the other hand, that support more general meshes, among which we cite, in particular, the Discrete Geometric Approach (DGA) of [20] (which, however, is restricted to the lowest-order). Recently, novel paradigms have appeared

that enable arbitrary approximation orders on general polyhedral meshes. Our focus here is on the Mixed High-Order (MHO) method of [1] which, in its lowest-order version, can be linked to the DGA method (cf. Remark 5). The MHO scheme considered here has several advantageous features which set it apart from the aforementioned methods: (i) Unlike standard finite element methods, it supports fairly general meshes possibly containing polyhedral elements and nonmatching interfaces; (ii) unlike the DGA method, it allows arbitrary approximation orders; (iii) these features come at only a moderate computational cost thanks to the existence of an equivalent primal formulation (cf. [21]) together with the possibility to reduce the global number of degrees of freedom by static condensation.

The main results of this work are new a posteriori error estimate for the MHO method and the use of the corresponding estimators in the context of an adaptive resolution algorithm for problems in electrostatics. For the derivation of our error estimators, we use a residual-based approach that relies on an abstract estimate inspired by [22]; cf. also [23]. We also cite [24] for the idea of relying on the equivalent primal formulation to derive a posteriori bounds for mixed methods. A key point to use the abstract error estimate in our context is a new reformulation of the stabilization term in the MHO method, which establishes a new connection with the hybridized version studied in [21]. The upper bound thus derived has no undetermined constants and, possibly up to minor modifications, also extends to sibling schemes such as the Hybrid High-Order method of [25] (cf. also [26]) and the DGA method of [20]. Adaptive resolution methods for related high-order polyhedral discretization methods, each with its own distinguishing features, are developed in [27, 28, 29].

The performance of the proposed error estimators to drive mesh adaptivity is thoroughly demonstrated on several three-dimensional academic and industrial test cases. Two strategies are explored for the mesh adaptation: a standard refinement procedure based on matching simplicial meshes, and an adaptive coarsening procedure inspired by [30]. The former strategy has a clear interest in engineering applications due to the fact that many efficient, automatic mesh generators exist for meshes composed of standard elements. Such mesh generators are also well-integrated in computer-assisted modeling chains. The adaptive coarsening strategy, on the other hand, is a new procedure enabled by the availability of polyhedral methods which shows promise as a simple means to reduce the number of degrees of freedom without the need to regenerate a new mesh. The idea consists in starting from an initial mesh composed of standard elements fine enough to capture the geometric features of the domain and the scales of the exact solution, and performing the actual computations on a sequence of polyhedral meshes obtained by adaptive coarsening of the former.

Our numerical tests show that the adaptive algorithm delivers a significant advantage with respect to computations on uniformly refined mesh sequences, both in terms of error vs. the number of degrees of freedom and in terms of error vs. computational wall time. In academic test cases with analytical singular solutions, we show in particular that the optimal orders of convergence are recovered on sequences of adap-

tively refined meshes. In the industrial test cases, for which an exact solution is not available, we show, on the other hand, that the adaptive algorithm yields a much better approximation of the capacitance with respect to computations on uniformly refined meshes.

The paper is organized as follows. In Section 2 we introduce the notion of general mesh along with the principal notations. In Section 3 we recall the Mixed High-Order method, its equivalent primal reformulation, and we state our a posteriori bound in Theorem 2. An extensive numerical validation of an adaptive resolution algorithm based on the estimators of Section 3 is presented in Section 4. Section 5 contains the proof of Theorem 2 preceeded by the required preparatory material. Readers mainly interested in the numerical recipe and results can skip this section at first reading.

2. Mesh and notations

The domain Ω is discretized by means of a mesh $\mathcal{T}_h$ defined as a finite collection of nonempty disjoint open polyhedral elements T of boundary ∂T and diameter h_T such that $\overline{\Omega} = \bigcup_{T\in\mathcal{T}_h} \overline{T}$ and $h = \max_{T\in\mathcal{T}_h} h_T$. A mesh face F is defined as a hyperplanar closed connected subset of $\overline{\Omega}$ with positive $(d-1)$-dimensional Hausdorff measure and such that (i) either there exist $T_1, T_2 \in \mathcal{T}_h$ such that $F \subset \partial T_1 \cap \partial T_2$ and F is called an interface or (ii) there exists $T \in \mathcal{T}_h$ such that $F \subset \partial T \cap \partial\Omega$ and F is called a boundary face. Interfaces are collected in the set $\mathcal{F}_h^{\mathrm{i}}$, boundary faces in the set $\mathcal{F}_h^{\mathrm{b}}$, and we let $\mathcal{F}_h := \mathcal{F}_h^{\mathrm{i}} \cup \mathcal{F}_h^{\mathrm{b}}$. For all $T \in \mathcal{T}_h$, $\mathcal{F}_T := \{F \in \mathcal{F}_h \mid F \subset \partial T\}$ denotes the set of faces contained in ∂T, and we denote by $\boldsymbol{n}_{\partial T}$ the piecewise constant field on ∂T such that, for all $F \in \mathcal{F}_T$, $\boldsymbol{n}_{\partial T|F} = \boldsymbol{n}_{TF}$ with $\boldsymbol{n}_{TF}$ unit normal vector to F pointing out of T. When working with refined mesh sequences $(\mathcal{T}_h)_{h\in\mathcal{H}}$ (where $\mathcal{H} = \{h\}$ denotes a set of positive meshsizes), we assume that the regularity assumptions of [31, Definition 1.38] hold.

We additionally assume that the mesh is compatible with the partition P_Ω, meaning that any element $T \in \mathcal{T}_h$ belongs to one and only one subdomain $\Omega_i \in P_\Omega$. As a consequence, the jumps of $\boldsymbol{K}$ can only occur at interfaces. For all $T \in \mathcal{T}_h$, $\underline{K}_T$ and $\overline{K}_T$ denote, respectively, the smallest and largest eigenvalues of $\boldsymbol{K}_T := \boldsymbol{K}_{|T}$.

For any subset $X \subset \overline{\Omega}$, we respectively denote by $(\cdot,\cdot)_X$ and $\|\cdot\|_X$ the standard inner product and norm of $L^2(X)$, the space of square-integrable functions on X:

$$(v,w)_X := \int_X vw, \qquad \|v\|_X := (v,v)_X^{1/2}.$$

When $X = \Omega$, the subscript is omitted. The same notation is used in the vector-valued case $L^2(X)^d$. Additionally, when $X \in \mathcal{T}_h$ or $X \in \mathcal{F}_h$, for any $l \in \mathbb{N}$ we denote by $\mathbb{P}^l(X)$ the space spanned by the restrictions to X of polynomials of total degree l, and we introduce the L^2-orthogonal projector $\pi_X^l : L^2(X) \to \mathbb{P}^l(X)$ such that, for all $w \in L^2(X)$,

$$(\pi_X^l w - w, v)_X = 0 \qquad \forall v \in \mathbb{P}^l(X). \tag{2}$$

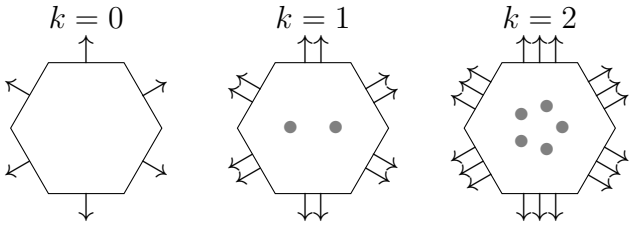


Figure 1: Local degrees of freedom in $\underline{\boldsymbol{\Sigma}}_T^k$ for $k \in \{0, 1, 2\}$.

For every mesh element $T \in \mathcal{T}_h$, we also define the space of broken polynomial functions on ∂T:

$$\mathbb{P}^l(\mathcal{F}_T) := \left\{ v \in L^2(\partial T) \mid v_{|F} \in \mathbb{P}^l(F) \quad \forall F \in \mathcal{F}_T \right\}, \tag{3}$$

and we denote by $\pi_{\partial T}^l$ the corresponding L^2-orthogonal projector such that, for all $w \in L^2(\partial T)$, $(\pi_{\partial T}^l w)_{|F} = \pi_F^l w$ for all $F \in \mathcal{F}_T$. Finally, we introduce the global spaces of discontinuous polynomial functions of degree l on the mesh $\mathcal{T}_h$ and on its skeleton $\partial\mathcal{T}_h = \bigcup_{F \in \mathcal{F}_h} F$:

$$\mathbb{P}^l(\mathcal{T}_h) := \left\{ v \in L^2(\Omega) \mid v_{|T} \in \mathbb{P}^l(T) \; \forall T \in \mathcal{T}_h \right\}, \quad \mathbb{P}^l(\mathcal{F}_h) := \left\{ v \in L^2(\partial\mathcal{T}_h) \mid v_{|F} \in \mathbb{P}^l(F) \; \forall F \in \mathcal{F}_h \right\}. \tag{4}$$

3. A posteriori error estimate for the Mixed High-Order method

We recall in this section the MHO method, its reformulation as a primal coercive problem, and state our a posteriori error estimate.

3.1. The Mixed High-Order method

The MHO method is inspired by the following classical weak formulation of problem (1): Find $(\boldsymbol{\sigma}, u) \in \boldsymbol{\Sigma} \times U$ such that, for all $(\boldsymbol{\tau}, v) \in \boldsymbol{\Sigma} \times U$,

$$(\boldsymbol{K}^{-1}\boldsymbol{\sigma}, \boldsymbol{\tau}) + (u, \boldsymbol{\nabla}\cdot\boldsymbol{\tau}) = 0, \tag{5a}$$

$$-(\boldsymbol{\nabla}\cdot\boldsymbol{\sigma}, v) = (f, v), \tag{5b}$$

where $\boldsymbol{\Sigma} := \boldsymbol{H}(\mathrm{div}; \Omega)$ denotes the space of square-integrable vector-valued functions whose divergence is also square integrable, while $U := L^2(\Omega)$. The idea consists in selecting suitable degrees of freedom (DOFs) inside each element, and in using them to locally reconstruct the divergence operator and the vector variable $\boldsymbol{\sigma}$. These reconstructions are then used to write discrete counterparts of the terms in (5).

Let a polynomial degree $k \in \mathbb{N}$ be fixed, and consider a mesh element $T \in \mathcal{T}_h$. The local space of DOFs for the vector variable is defined as (cf. Figure 1),

$$\underline{\boldsymbol{\Sigma}}_T^k := (\boldsymbol{K}_T \boldsymbol{\nabla} \mathbb{P}^k(T)) \times \mathbb{P}^k(\mathcal{F}_T),$$

with $\mathbb{P}^k(\mathcal{F}_T)$ given by (3). Notice that element-based DOFs are present only for $k \geqslant 1$ since $\boldsymbol{K}_T\boldsymbol{\nabla}\mathbb{P}^0(T) = \{\boldsymbol{0}\}$. For a collection of DOFs in $\underline{\boldsymbol{\Sigma}}_T^k$, we use the underlined notation $\underline{\boldsymbol{\tau}}_T = (\boldsymbol{\tau}_T, \tau_{\partial T})$, where $\boldsymbol{\tau}_T$ contains the vector-valued element-based DOFs, while $\tau_{\partial T}$ contains the scalar-valued face-based DOFs (which can be interpreted as the polynomial moments of the outward normal component of the vector variable).

The divergence reconstruction operator $D_T^k : \underline{\boldsymbol{\Sigma}}_T^k \to \mathbb{P}^k(T)$ is such that, for all $\underline{\boldsymbol{\tau}}_T \in \underline{\boldsymbol{\Sigma}}_T^k$ and all $q \in \mathbb{P}^k(T)$,

$$(D_T^k\underline{\boldsymbol{\tau}}_T, q)_T = -(\boldsymbol{\tau}_T, \boldsymbol{\nabla}q)_T + (\tau_{\partial T}, q)_{\partial T}. \tag{6}$$

The right-hand side of (6) is designed to resemble an integration by parts formula where the role of the vector variable inside T and on ∂T is played by the element-based and face-based DOFs, respectively. By similar principles, the vector variable reconstruction operator $\boldsymbol{S}_T^k : \underline{\boldsymbol{\Sigma}}_T^k \to \boldsymbol{K}_T\boldsymbol{\nabla}\mathbb{P}^{k+1}(T)$ is such that, for all $\underline{\boldsymbol{\tau}}_T \in \underline{\boldsymbol{\Sigma}}_T^k$ and all $w \in \mathbb{P}^{k+1}(T)$,

$$(\boldsymbol{S}_T^k\underline{\boldsymbol{\tau}}_T, \boldsymbol{\nabla}w)_T = -(D_T^k\underline{\boldsymbol{\tau}}_T, w)_T + (\tau_{\partial T}, w)_{\partial T}. \tag{7}$$

Notice that, for all $\underline{\boldsymbol{\tau}}_T = (\boldsymbol{\tau}_T, \tau_{\partial T}) \in \underline{\boldsymbol{\Sigma}}_T^k$, $\boldsymbol{S}_T^k\underline{\boldsymbol{\tau}}_T$ is a polynomial one degree higher than the element-based DOFs $\boldsymbol{\tau}_T$.

For all $T \in \mathcal{T}_h$, the local contribution $(\boldsymbol{K}^{-1}\boldsymbol{\sigma}, \boldsymbol{\tau})_T$ to the first term in (5a) is approximated by the bilinear form m_T on $\underline{\boldsymbol{\Sigma}}_T^k \times \underline{\boldsymbol{\Sigma}}_T^k$ such that

$$\mathrm{m}_T(\underline{\boldsymbol{\sigma}}_T, \underline{\boldsymbol{\tau}}_T) := (\boldsymbol{K}_T^{-1}\boldsymbol{S}_T^k\underline{\boldsymbol{\sigma}}_T, \boldsymbol{S}_T^k\underline{\boldsymbol{\tau}}_T)_T + \mathrm{s}_{\boldsymbol{\Sigma},T}(\underline{\boldsymbol{\sigma}}_T, \underline{\boldsymbol{\tau}}_T), \tag{8}$$

with stabilization bilinear form given by

$$\mathrm{s}_{\boldsymbol{\Sigma},T}(\underline{\boldsymbol{\sigma}}_T, \underline{\boldsymbol{\tau}}_T) := (\gamma_{\partial T}(\boldsymbol{S}_T^k\underline{\boldsymbol{\sigma}}_T\cdot\boldsymbol{n}_{\partial T} - \sigma_{\partial T}), \boldsymbol{S}_T^k\underline{\boldsymbol{\tau}}_T\cdot\boldsymbol{n}_{\partial T} - \tau_{\partial T})_{\partial T}, \quad \gamma_{\partial T} := \frac{h_T}{\boldsymbol{K}_T\boldsymbol{n}_{\partial T}\cdot\boldsymbol{n}_{\partial T}}. \tag{9}$$

In (9), we penalize in a least-square sense the difference between two quantities both of which represent the normal component of the vector variable on the boundary of T. This stabilization is required for the bilinear form m_T to be coercive on $\underline{\boldsymbol{\Sigma}}_T^k$; cf. [1, Lemma 4].

We next introduce the global space $\widehat{\underline{\boldsymbol{\Sigma}}}_h^k$ of fully discontinuous DOFs for the vector variable as well as its subspace $\underline{\boldsymbol{\Sigma}}_h^k$ with continuous interface DOFs:

$$\widehat{\underline{\boldsymbol{\Sigma}}}_h^k := \bigtimes_{T\in\mathcal{T}_h} \underline{\boldsymbol{\Sigma}}_T^k, \qquad \underline{\boldsymbol{\Sigma}}_h^k := \left\{\underline{\boldsymbol{\tau}}_h \in \widehat{\underline{\boldsymbol{\Sigma}}}_h^k \mid \forall F \in \mathcal{F}_{T_1} \cap \mathcal{F}_{T_2},\ \tau_{\partial T_1|F} + \tau_{\partial T_2|F} = 0\right\}.$$

The approximation of the scalar variable is sought in

$$U_h^k := \mathbb{P}^k(\mathcal{T}_h). \tag{10}$$

The MHO method reads: Find $(\underline{\boldsymbol{\sigma}}_h, u_h) \in \underline{\boldsymbol{\Sigma}}_h^k \times U_h^k$ such that for all $(\underline{\boldsymbol{\tau}}_h, v_h) \in \underline{\boldsymbol{\Sigma}}_h^k \times U_h^k$ it holds

$$\mathrm{m}_h(\underline{\boldsymbol{\sigma}}_h, \underline{\boldsymbol{\tau}}_h) + \mathrm{b}_h(\underline{\boldsymbol{\tau}}_h, u_h) = 0, \tag{11a}$$

$$-\mathrm{b}_h(\underline{\boldsymbol{\sigma}}_h, v_h) = (f, v_h), \tag{11b}$$

with global bilinear forms m_h on $\underline{\widehat{\boldsymbol{\Sigma}}}_h^k \times \underline{\widehat{\boldsymbol{\Sigma}}}_h^k$ and b_h on $\underline{\widehat{\boldsymbol{\Sigma}}}_h^k \times U_h^k$ obtained by element-by-element assembly setting

$$\mathrm{m}_h(\underline{\boldsymbol{\sigma}}_h, \underline{\boldsymbol{\tau}}_h) := \sum_{T\in\mathcal{T}_h} \mathrm{m}_T(\underline{\boldsymbol{\sigma}}_T, \underline{\boldsymbol{\tau}}_T), \qquad \mathrm{b}_h(\underline{\boldsymbol{\tau}}_h, v_h) := \sum_{T\in\mathcal{T}_h} (D_T^k \underline{\boldsymbol{\tau}}_T, v_T)_T,$$

where the restrictions to T of the global quantities $\underline{\boldsymbol{\tau}}_h \in \underline{\widehat{\boldsymbol{\Sigma}}}_h^k$ and $v_h \in U_h^k$ have been denoted by $\underline{\boldsymbol{\tau}}_T$ and v_T, respectively. The link with the Discrete Geometric Approach of [20] is detailed in Remark 5 below. Convergence as h^{k+1} for the vector-variable and as h^{k+2} for the scalar-variable is proved in [1].

3.2. Reformulation as a coercive problem

We next discuss a reformulation of (11) as a coercive problem which is required to derive our a posteriori error estimate. At the continuous level, this corresponds to eliminating the vector variable $\boldsymbol{\sigma}$ in (1) and considering the following classical primal formulation where the scalar variable u is the sole unknown: Find $u \in V := H_0^1(\Omega)$ such that, for all $v \in V$,

$$\mathrm{a}(u, v) := (\boldsymbol{K}\boldsymbol{\nabla} u, \boldsymbol{\nabla} v) = (f, v), \tag{12}$$

where $H_0^1(\Omega)$ contains square-integrable, finite energy functions which comply with the boundary condition (1c). Classically, $\boldsymbol{\sigma} = \boldsymbol{K}\boldsymbol{\nabla} u$. At the discrete level, the local elimination of the vector variable is performed enforcing the continuity of interface DOFs in $\underline{\boldsymbol{\Sigma}}_h^k$ by Lagrange multipliers (which can be interpreted as traces of the scalar variable) and inverting inside each element $T \in \mathcal{T}_h$ the local constitutive laws expressed by (11a). Here, we only recall the main points and refer the reader to [21] for the details.

Define the following space of hybrid DOFs (cf. (10) and (4) for the definitions of the Cartesian factors):

$$\underline{V}_h^k := U_h^k \times \mathbb{P}^k(\mathcal{F}_h).$$

For a generic element of $\underline{V}_h^k$ we use the underlined notation $\underline{v}_h = ((v_T)_{T\in\mathcal{T}_h}, (v_F)_{F\in\mathcal{F}_h})$, and we denote by v_h (no underline) the broken polynomial function in U_h^k obtained from element-based DOFs and such that $v_{h|T} = v_T$ for all $T \in \mathcal{T}_h$. The discrete counterpart of u regarded as an element of V is sought in the following subspace $\underline{V}_{h,0}^k$ of $\underline{V}_h^k$ incorporating the homogeneous Dirichlet condition on $\partial\Omega$:

$$\underline{V}_{h,0}^k := \left\{ \underline{v}_h \in \underline{V}_h^k \mid v_F = 0 \quad \forall F \in \mathcal{F}_h^{\mathrm{b}} \right\}.$$

Let now a mesh element $T \in \mathcal{T}_h$ be fixed. We denote by $\underline{V}_T^k$ the restriction of $\underline{V}_h^k$ to T (cf. Figure 2) and, for all $\underline{v}_T \in \underline{V}_T^k$, we let $v_{\partial T}$ be the broken polynomial function on ∂T such that $v_{\partial T|F} = v_F$ for all $F \in \mathcal{F}_T$, so that $\underline{v}_T = (v_T, v_{\partial T})$. From the scalar variable DOFs in $\underline{V}_T^k$, we can define a reconstruction of vector variable DOFs in $\underline{\boldsymbol{\Sigma}}_T^k$ through the scalar-to-vector variable mapping operator $\underline{\boldsymbol{\varsigma}}_T^k : \underline{V}_T^k \to \underline{\boldsymbol{\Sigma}}_T^k$ such that, for all $\underline{v}_T \in \underline{V}_T^k$ and all $\underline{\boldsymbol{\tau}}_T \in \underline{\boldsymbol{\Sigma}}_T^k$,

$$\mathrm{m}_T(\underline{\boldsymbol{\varsigma}}_T^k \underline{v}_T, \underline{\boldsymbol{\tau}}_T) = -(v_T, D_T^k \underline{\boldsymbol{\tau}}_T)_T + (v_{\partial T}, \tau_{\partial T})_{\partial T} \tag{13a}$$

$$= (\boldsymbol{\nabla} v_T, \boldsymbol{\tau}_T)_T + (v_{\partial T} - v_T, \tau_{\partial T})_{\partial T}, \tag{13b}$$

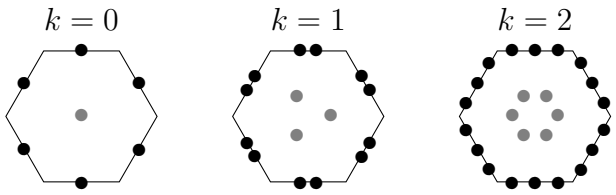


Figure 2: Local degrees of freedom in $\underline{V}_T^k$ for $k \in \{0,1,2\}$. Degrees of freedom represented by gray dots can be eliminated locally by the static condensation procedure described in [32, Section 2.4]

where we have used the definition (6) of $D_T^k \underline{\boldsymbol{\tau}}_T$ with $q = v_T$ to pass to the second line. Recalling that the stabilization term (9) is designed so that m_T is coercive on $\underline{\boldsymbol{\Sigma}}_T^k$, (13) defines a well-posed problem in $\underline{\boldsymbol{\varsigma}}_T^k \underline{v}_T$. A high-order reconstruction of the scalar variable can then be defined through the operator $p_T^{k+1} : \underline{V}_T^k \to \mathbb{P}^{k+1}(T)$ such that, for all $\underline{v}_T \in \underline{V}_T^k$,

$$\boldsymbol{K}_T \boldsymbol{\nabla} p_T^{k+1} \underline{v}_T = \boldsymbol{S}_T^k(\underline{\boldsymbol{\varsigma}}_T^k \underline{v}_T), \qquad (p_T^{k+1} \underline{v}_T - v_T, 1)_T = 0. \tag{14}$$

Notice that, for a given $\underline{v}_T \in \underline{V}_T^k$, $p_T^{k+1} \underline{v}_T$ is a polynomial one degree higher than the element based DOFs v_T. The scalar value reconstruction $p_T^{k+1} \underline{v}_T$ can be computed directly from the hybrid DOFs in $\underline{v}_T$ by means of formula (31) below.

Consider the following approximation of (12): Find $(\underline{\boldsymbol{\sigma}}_h, \underline{u}_h) \in \widehat{\underline{\boldsymbol{\Sigma}}}_h^k \times \underline{V}_{h,0}^k$ such that, for all $T \in \mathcal{T}_h$,

$$\underline{\boldsymbol{\sigma}}_T = \underline{\boldsymbol{\varsigma}}_T^k \underline{u}_T, \tag{15a}$$

and, for all $\underline{v}_h \in \underline{V}_{h,0}^k$,

$$\mathrm{a}_h(\underline{u}_h, \underline{v}_h) := \sum_{T \in \mathcal{T}_h} \mathrm{a}_T(\underline{u}_T, \underline{v}_T) = (f, v_h), \tag{15b}$$

where the local bilinear form a_T on $\underline{V}_T^k \times \underline{V}_T^k$ is such that

$$\mathrm{a}_T(\underline{u}_T, \underline{v}_T) := (\boldsymbol{K}_T \boldsymbol{\nabla} p_T^{k+1} \underline{u}_T, \boldsymbol{\nabla} p_T^{k+1} \underline{v}_T)_T + \mathrm{s}_{U,T}(\underline{u}_T, \underline{v}_T), \quad \mathrm{s}_{U,T}(\underline{u}_T, \underline{v}_T) := \mathrm{s}_{\boldsymbol{\Sigma},T}(\underline{\boldsymbol{\varsigma}}_T^k \underline{u}_T, \underline{\boldsymbol{\varsigma}}_T^k \underline{v}_T). \tag{16}$$

It can be proved that $\underline{\boldsymbol{\sigma}}_h \in \underline{\boldsymbol{\Sigma}}_h^k$ and, recalling that $u_h \in U_h^k$ is the broken polynomial function obtained from element-based DOFs in $\underline{u}_h$, $(\underline{\boldsymbol{\sigma}}_h, u_h) \in \underline{\boldsymbol{\Sigma}}_h^k \times U_h^k$ coincides with the solution of (11).

Remark 1 (Implementation)**.** *In the practical implementation, the equivalent primal form* (15) *is used, as it corresponds to a coercive problem (as opposed to the mixed form* (11)*, which is a saddle point problem). The size of the linear system to solve is further reduced by locally eliminating element-based DOFs by static condensation (represented in gray in Figure 2). In the end, accounting for the strong enforcement of Dirichlet boundary conditions, a matrix of size* $N_{\mathrm{dof}} \times N_{\mathrm{dof}}$ *with*

$$N_{\mathrm{dof}} = \mathrm{card}(\mathcal{F}_h^{\mathrm{i}}) \times \binom{k+d-1}{k}$$

has to be inverted. The static condensation procedure is analogous to the one described in [32, Section 2.4], to which we refer for further details.

3.3. A posteriori error estimate

We state in this section our a posteriori error estimate, whose proof is postponed to Section 5.3. Denote by $\boldsymbol{\nabla}_h$ the broken gradient on the broken Sobolev space $H^1(\mathcal{T}_h) := \big\{v \in L^2(\Omega) \mid v_{|T} \in H^1(T) \quad \forall T \in \mathcal{T}_h\big\}$. Let $(\underline{\boldsymbol{\sigma}}_h, \underline{u}_h) \in \underline{\boldsymbol{\Sigma}}_h^k \times \underline{V}_{h,0}^k$ solve (15). We introduce the broken polynomial fields $\breve{\boldsymbol{\sigma}}_h \in \boldsymbol{K}\boldsymbol{\nabla}_h\mathbb{P}^{k+1}(\mathcal{T}_h)$ and $\breve{u}_h \in \mathbb{P}^{k+1}(\mathcal{T}_h)$ such that, for all $T \in \mathcal{T}_h$,

$$\breve{\boldsymbol{\sigma}}_{h|T} = \breve{\boldsymbol{\sigma}}_T := \boldsymbol{S}_T^k \underline{\boldsymbol{\sigma}}_T, \qquad \breve{u}_{h|T} = \breve{u}_T := p_T^{k+1}\underline{u}_T, \tag{17}$$

and observe that, by (14),

$$\breve{\boldsymbol{\sigma}}_h = \boldsymbol{K}\boldsymbol{\nabla}_h\breve{u}_h.$$

To treat the stabilization term in the spirit of [32], we will also need the boundary residual operator $r_{\partial T}^k : \mathbb{P}^k(\mathcal{F}_T) \to \mathbb{P}^k(\mathcal{F}_T)$ such that, for all $\lambda \in \mathbb{P}^k(\mathcal{F}_T)$,

$$r_{\partial T}^k\lambda := \pi_{\partial T}^k\left(\lambda - p_T^{k+1}(0,\lambda) + \pi_T^k p_T^{k+1}(0,\lambda)\right), \tag{18}$$

and its adjoint $r_{\partial T}^{*,k}$ satisfying

$$\forall \lambda \in \mathbb{P}^k(\mathcal{F}_T), \qquad (r_{\partial T}^k\lambda, \mu)_{\partial T} = (\lambda, r_{\partial T}^{*,k}\mu)_{\partial T} \qquad \forall \mu \in \mathbb{P}^k(\mathcal{F}_T). \tag{19}$$

Finally, we recall the following local Poincaré and Friedrichs inequalities, valid for all $T \in \mathcal{T}_h$ and all $\varphi \in H^1(T)$:

$$\|\varphi - \pi_T^0\varphi\|_T \leqslant C_{\mathrm{P},T}\underline{K}_T^{-1/2}h_T\|\boldsymbol{K}_T^{1/2}\boldsymbol{\nabla}\varphi\|_T, \tag{20}$$

$$\|\varphi - \pi_T^0\varphi\|_{\partial T} \leqslant C_{\mathrm{F},T}^{1/2}\underline{K}_T^{-1/2}h_T^{1/2}\|\boldsymbol{K}_T^{1/2}\boldsymbol{\nabla}\varphi\|_T. \tag{21}$$

In (20), $C_{\mathrm{P},T} = \pi^{-1}$ if T is convex (cf. [33, 34]), while we refer to [24] for upper bounds on nonconvex elements. In (21), $C_{\mathrm{F},T} = C_{\mathrm{P},T}(h_T|\partial T|_{d-1}|T|_d^{-1})(2d^{-1} + C_{\mathrm{P},T})$ when T is a simplex, while, for more general element shapes, it can be bounded in terms of $C_{\mathrm{P},T}$, d, k and the mesh regularity parameter.

Theorem 2 (Error upper bound)**.** *Let $(\boldsymbol{\sigma}, u) \in \boldsymbol{\Sigma} \times U$ and $(\underline{\boldsymbol{\sigma}}_h, \underline{u}_h) \in \underline{\boldsymbol{\Sigma}}_h^k \times \underline{V}_{h,0}^k$ solve* (5) *and* (15)*, respectively (so that, in particular, $u \in V$ solves* (12) *and $(\underline{\boldsymbol{\sigma}}_h, u_h) \in \underline{\boldsymbol{\Sigma}}_h^k \times U_h^k$ solves* (11)*). Let u_h^* be an arbitrary function in V. Then, it holds*

$$\|\boldsymbol{K}^{-1/2}(\boldsymbol{\sigma} - \breve{\boldsymbol{\sigma}}_h)\| = \|\boldsymbol{K}^{1/2}\boldsymbol{\nabla}_h(u - \breve{u}_h)\| \leqslant \eta := \left\{\sum_{T\in\mathcal{T}_h}\left[\eta_{\mathrm{nc},T}^2 + (\eta_{\mathrm{res},T} + \eta_{\mathrm{sta},T})^2\right]\right\}^{1/2}, \tag{22}$$

where the local nonconformity, residual and stabilization estimators are such that, for all $T \in \mathcal{T}_h$, letting $u_T^* := u_{h|T}^*$,

$$\eta_{\mathrm{nc},T} := \|\boldsymbol{K}_T^{1/2}\boldsymbol{\nabla}(\breve{u}_T - u_T^*)\|_T, \tag{23a}$$

$$\eta_{\mathrm{res},T} := C_{\mathrm{P},T}\underline{K}_T^{-1/2}h_T\|(f + \boldsymbol{\nabla}\cdot\breve{\boldsymbol{\sigma}}_T) - \pi_T^0(f + \boldsymbol{\nabla}\cdot\breve{\boldsymbol{\sigma}}_T)\|_T, \tag{23b}$$

$$\eta_{\mathrm{sta},T} := C_{\mathrm{F},T}^{1/2}\underline{K}_T^{-1/2}h_T^{1/2}\|r_{\partial T}^{*,k}(\sigma_{\partial T} - \boldsymbol{\sigma}_T\cdot\boldsymbol{n}_{\partial T})\|_{\partial T}. \tag{23c}$$

The efficiency of the above estimators can be proved by standard techniques based on the use of bubble functions; cf., e.g., [35]. Providing the details lies outside of the scope of the present, application-oriented paper. The use of the a posteriori error estimators of Theorem 2 in the context of adaptive mesh refinement is exemplified in Algorithm 1 below (cf. Section 4 for further details).

Several remarks are of order.

Remark 3 (V-conforming potential reconstruction u_h^*)**.** *The V-conforming potential $u_h^* \in V$ used for the actual computation of the estimator $\eta_{\mathrm{nc},T}$ in Section 4 is obtained as follows: If $\mathcal{T}_h$ is a matching simplicial mesh, we simply set*

$$u_h^* := \mathcal{I}_{\mathrm{av}}^{k+1} \breve{u}_h,$$

where, for any polynomial degree $l \geqslant 1$, $\mathcal{I}_{\mathrm{av}}^l$ denotes the averaging operator on the standard Lagrange finite element space $\mathbb{P}^l(\mathcal{T}_h) \cap V$ such that, for all $v_h \in \mathbb{P}^l(\mathcal{T}_h)$ and all interpolation node N inside Ω,

$$(\mathcal{I}_{\mathrm{av}}^l v_h)(N) := \frac{1}{\mathrm{card}(\mathcal{T}_N)} \sum_{T \in \mathcal{T}_N} v_{h|T}(N),$$

where $\mathcal{T}_N \subset \mathcal{T}_h$ collects the simplices to which N belongs, while $(\mathcal{I}_{\mathrm{av}}^l v_h)(N) := 0$ if $N \in \partial\Omega$. Clearly, $u_h^* \in V$ since u_h^* is continuous in $\overline{\Omega}$ and it vanishes on $\partial\Omega$. On general meshes, u_h^* can be obtained using the averaging operator on a conforming simplicial submesh of the computational mesh $\mathcal{T}_h$. When performing adaptive coarsening as in Section 4.1.3, the underlying matching simplicial mesh can be used for this purpose.

Remark 4 (Alternative form for $\eta_{\mathrm{sta},T}$)**.** *By virtue of equation* (33) *below, $\eta_{\mathrm{sta},T}$ can take the following alternative form, which is more convenient for the implementation since it does not require the construction of the operator $r_{\partial T}^{*,k}$:*

$$\eta_{\mathrm{sta},T} := C_{\mathrm{F},T}^{1/2} \left(\frac{\overline{K}_T}{\underline{K}_T}\right)^{1/2} \mathrm{s}_{\boldsymbol{\Sigma},T}(\underline{\boldsymbol{\sigma}}_T, \underline{\boldsymbol{\sigma}}_T)^{1/2}. \tag{24}$$

When the diffusion coefficient $\boldsymbol{K}$ is strongly anisotropic, however, the form (23c) *may be preferable since it yields a sharper estimate.*

Remark 5 (A posteriori error estimate for the DGA method)**.** *Assume that, for every mesh element $T \in \mathcal{T}_h$, there exists a point $\boldsymbol{x}_T$ with respect to which T is star-shaped. In this case, the DGA method is intimately linked to the lowest-order version of the MHO method* (11) *corresponding to $k = 0$. Indeed, it can be proved that the sole difference consists in replacing the the stabilization parameter $\gamma_{\partial T}$ defined in* (9) *by $\gamma_{\partial T}^{\mathrm{dga}} \in \mathbb{P}^0(\mathcal{F}_T)$ such that, for all $F \in \mathcal{F}_T$,*

$$\gamma_{\partial T}^{\mathrm{dga}}{}_{|F} = \frac{\mathfrak{h}_{TF}}{dK_{TF}},$$

where $\mathfrak{h}_{TF} = \mathrm{dist}(\boldsymbol{x}_T, F)$ and $K_{TF} := \mathfrak{h}_{TF}^2 \left(\boldsymbol{K}_T^{-1}(\boldsymbol{x}_F - \boldsymbol{x}_T)\cdot(\boldsymbol{x}_F - \boldsymbol{x}_T)\right)^{-1}$ with $\boldsymbol{x}_F$ denoting the barycenter of F. For a regular mesh, $\gamma_{\partial T}^{\mathrm{dga}}$ is equivalent to $\gamma_{\partial T}$ up to factors that depend on the mesh regularity parameter and on the anisotropy ratio of $\boldsymbol{K}_T$. Scanning the proofs of Lemma 8 and of Theorem 2 below, one can check that this change has no effect on the expressions (23) *of the estimators (including $\eta_{\mathrm{sta},T}$).*

Remark 6 (A posteriori error estimate for the HHO method)**.** *In [25], a primal Hybrid High-Order method is proposed whose formulation is analogous to* (15b) *except that the bilinear form* a_T *defined by* (16) *is replaced by*

$$\mathrm{a}_T^{\mathrm{hho}}(\underline{u}_T,\underline{v}_T) := (\boldsymbol{K}_T\boldsymbol{\nabla}p_T^{k+1}\underline{u}_T,\boldsymbol{\nabla}p_T^{k+1}\underline{v}_T)_T + \mathrm{s}_{U,T}^{\mathrm{hho}}(\underline{u}_T,\underline{v}_T).$$

A quick inspection reveals that the sole difference with respect to (16) *lies in the (equivalent uniformly in* h*) stabilization bilinear form*

$$\mathrm{s}_{U,T}^{\mathrm{hho}}(\underline{u}_T,\underline{v}_T) := \big(\gamma_{\partial T}^{-1}\pi_{\partial T}^k(u_{\partial T}-\widehat{p}_T^{k+1}\underline{u}_T),\pi_{\partial T}^k(v_{\partial T}-\widehat{p}_T^{k+1}\underline{v}_T)\big)_{\partial T},$$

with $\widehat{p}_T^{k+1}\underline{v}_T := v_T + (p_T^{k+1}\underline{v}_T - \pi_T^k p_T^{k+1}\underline{v}_T)$*. The a posteriori error estimate* (22) *holds for the HHO method provided we modify the stabilization estimator* (23c) *as follows:*

$$\eta_{\mathrm{sta},T}^{\mathrm{hho}} := C_{\mathrm{F},T}^{1/2}\underline{K}_T^{-1/2}h_T^{1/2}\|r_{\partial T}^{*,k}(\gamma_{\partial T}^{-1}r_{\partial T}^k(u_{\partial T}-u_T))\|_{\partial T}.$$

To prove this result, it suffices to replace equation (32) *in the proof of Theorem 2 given in Section 5.3 below by the following formula proved in [32, Section 3.1]:*

$$\mathrm{s}_{U,T}^{\mathrm{hho}}(\underline{u}_T,\underline{v}_T) = \big(r_{\partial T}^{*,k}(\gamma_{\partial T}^{-1}r_{\partial T}^k(u_{\partial T}-u_T)),v_{\partial T}-v_T\big)_{\partial T}.$$

4. Numerical results

In this section we present an extensive panel of numerical tests for a resolution algorithm where local mesh adaptation is driven by the a posteriori error estimators of Theorem 2. All the test cases presented here are fully three-dimensional, and include both academic and industrial problems.

When working with matching simplicial meshes, mesh generation and adaptive refinement are performed using the open source software Netgen [36]. Each new mesh is produced by specifying the local meshsize in the barycenters of the elements. We have found good results by refining at each iteration the 5% of elements T where the local error measured by the quantity $\big[\eta_{\mathrm{nc},T}^2 + (\eta_{\mathrm{res},T}+\eta_{\mathrm{sta},T})^2\big]^{1/2}$ is larger (cf. (23) for the definitions of the local estimators). The implemented automatic mesh adaptivity procedure is summarized in Algorithm 1.

All the sparse linear systems are solved with the algebraic multigrid solver AGMG [37] by stopping the iterations once the relative residual reaches $1\cdot10^{-8}$. The computations were run on a laptop equipped with an Intel Core i7-3720QM processor clocked at 2.60GHz and 16Gb of RAM.

The implementation relies on the primal reformulation (15). We measure the error in the energy- and L^2-norms defined as, respectively,

$$e_{\mathrm{en}} := \|\boldsymbol{K}^{1/2}\boldsymbol{\nabla}_h(u-\breve{u}_h)\|, \qquad e_{L^2} := \|u-\breve{u}_h\|,$$

Algorithm 1 Pseudocode of the automatic mesh adaptivity procedure.

Set a tolerance $\epsilon > 0$ and a maximum number of iterations $N_{\max}$
Generate an initial coarse mesh $\mathcal{T}_h^{(0)}$, set $n \leftarrow 0$, and let $\mathcal{T}_h^{(n)} \leftarrow \mathcal{T}_h^{(0)}$
repeat
  Solve the electrostatic problem formulated in primal form (15) on $\mathcal{T}_h^{(n)}$
  for $T \in \mathcal{T}_h^{(n)}$ **do**
    Compute and store the local estimator $\eta_T := \left[\eta_{\mathrm{nc},T}^2 + (\eta_{\mathrm{res},T} + \eta_{\mathrm{sta},T})^2\right]^{1/2}$ (cf. (23))
  end for
  for $T \in \mathcal{T}_h^{(n)}$ **do**
    if η_T is in the top 5% **then**
      Set a target diameter of $h_T/2$
    else
      Set a target diameter of h_T
    end if
  end for
  Set $n \leftarrow n+1$ and generate a novel mesh $\mathcal{T}_h^{(n)}$ by using the target element diameters
until $\eta < \epsilon$ (cf. (22)) **or** $n > N_{\max}$

where u denotes the exact solution while $\breve{u}_h$ is defined by (17). We also monitor the approximation of the electrostatic energy $E := \frac{1}{2}\,\mathrm{a}(u,u) - (f,u)$ defined as $E_h := \frac{1}{2}\,\mathrm{a}_h(\underline{u}_h, \underline{u}_h) - (f, u_h)$. This error measure is very important in electromagnetic applications given that most global quantities of interest, such as the capacitance considered in the industrial test cases of Section 4.2, are related to it. The displayed results also include the computational time defined as the total wall time needed for the simulation, including the pre-processing (mesh generation and creation of mesh incidences), the assembly of the sparse matrix, the solution of the linear system, and the post-processing (computation all the local variables, electrostatic energy, error estimators data storage).

4.1. Test cases with analytical solution

The performance of the adaptive method is first assessed on two problems for which an analytical solution is available.

4.1.1. Cube benchmark

To demonstrate the orders of convergence of the MHO method for smooth solutions on uniformly refined meshes, we numerically solve problem (1) on the unit cube $\Omega = (0,1)^3$ with $\boldsymbol{K} = \boldsymbol{I}_3$ and the right-hand f

selected so that the exact solution is

$$u = \sin(\pi x_1)\sin(\pi x_2)\sin(\pi x_3).$$

We evaluate the convergence rates for polynomial degrees $0 \leqslant k \leqslant 4$ using a sequence of four uniformly refined tetrahedral meshes. The convergence results displayed in Figures 3a–b confirm the theoretical predictions of [1, Theorems 6 and 7] for the convergence rates both in energy- and L^2-norms.

Figures 3c–d show the convergence in energy norm with respect to N_{dof} and the total computational wall time, respectively. Notice that, for $k \geqslant 1$, the linear system solution time has a suboptimal scaling with respect to the number of unknowns. This could probably be improved resorting to a discretization-tailored geometric multigrid method (an investigation on this subject is presently ongoing).

Figure 3e shows the difference between the estimated total electrostatic energy and the true one. Finally, Figure 3f represents the convergence of the computed electrostatic energy with respect to N_{dof}.

4.1.2. Fichera corner benchmark (adaptive mesh refinement)

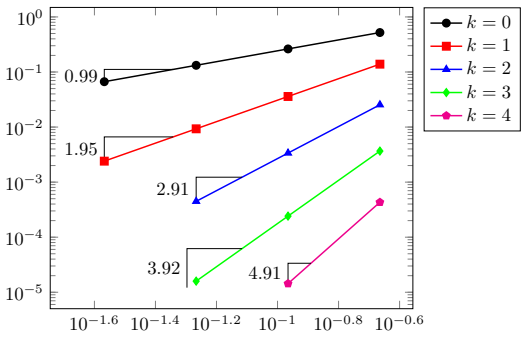
To demonstrate the effectiveness of our adaptive algorithm, we next consider a problem with a known singular solution and show that optimal orders of convergence (in terms of error vs. N_{dof}) are recovered when considering adaptively refined mesh sequences. Our focus is on the so-called Fichera corner problem [38], which consists in solving problem (1) on the domain $\Omega = (-1,1)^3 \backslash [0,1]^3$ with $\varepsilon = 1$, $f = -\frac{3}{4}\left(x_1^2 + x_2^2 + x_3^2\right)^{-3/4}$, $\boldsymbol{K} = \boldsymbol{I}_3$, and inhomogeneous Dirichlet boundary conditions chosen to comply with the solution

$$u(x_1, x_2, x_3) = \sqrt[4]{x_1^2 + x_2^2 + x_3^2}.$$

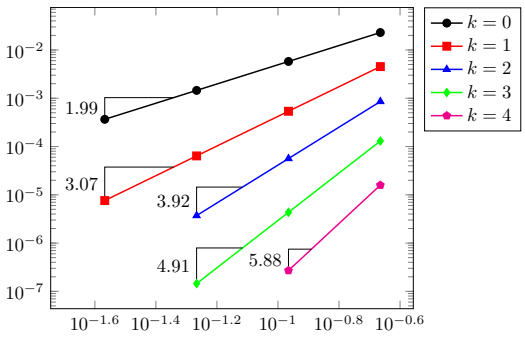
The convergence results displayed in Figures 4a–b confirm that optimal convergence rates in both the energy- and L^2-norms are recovered on adaptively refined mesh sequences. Figure 4c shows the convergence of the electrostatic energy with respect to N_{dof}, whereas Figure 4d displays the convergence of the energy error with respect to the total computational wall time.

The efficiency index $\eta / \|\boldsymbol{K}^{1/2}\boldsymbol{\nabla}_h(u - \breve{u}_h)\|$ is in most cases between one and three, see Figure 4e. The fact that the efficiency index is not monotonically decreasing when refinining is justified since each mesh is generated from scratch by using a scalar field that represents the required local meshsize. As a result, the mesh regularity parameter remains bounded but not constant when refining, which has an effect on the related quantities that appear in the error estimate. Nonconforming local mesh refinement could successfully address this problem, but this feature is not presently available in our implementation.

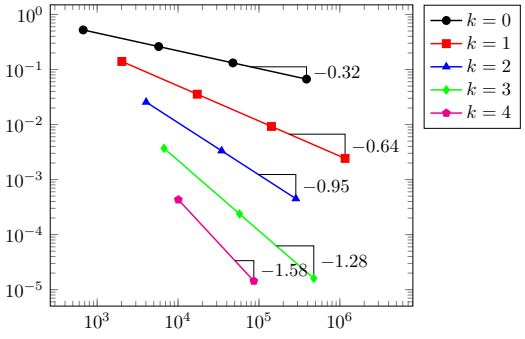
Figure 5 shows the distribution of the total error (a) and of its estimate (b) after two levels of mesh refinement. It can be seen that the error distribution is well-captured by the estimators of Theorem 2. Finally, Figure 6 represents a sequence of eight adaptively refined meshes. Notice that, as expected, refinement occurs mainly in the vicinity of the reentrant corner.



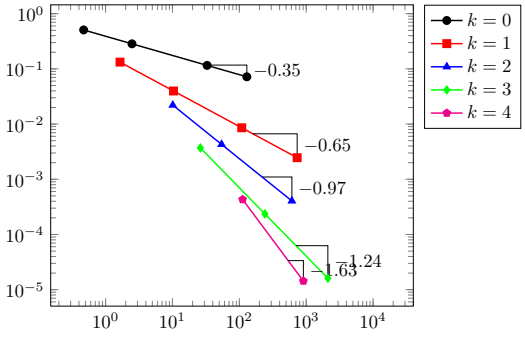
(a) e_{en} vs. h



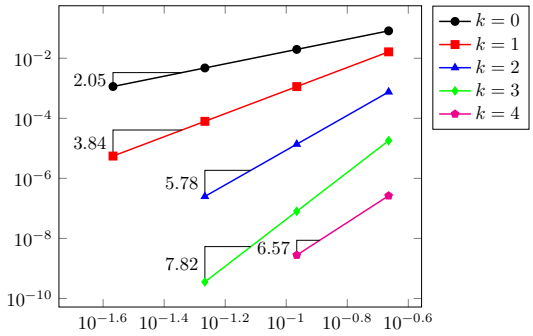
(b) e_{L^2} vs. h



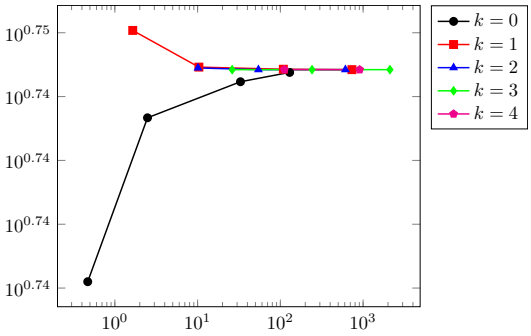
(c) e_{en} vs. N_{dof}



(d) e_{en} vs. computing time

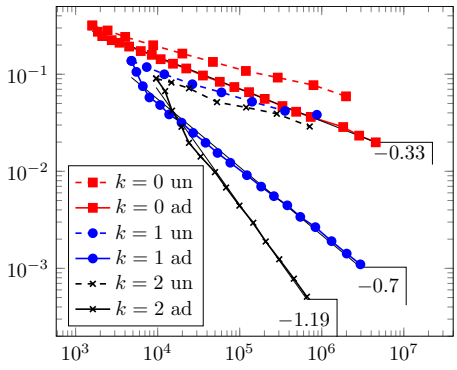


(e) Electrostatic energy difference $E - E_h$ vs. h

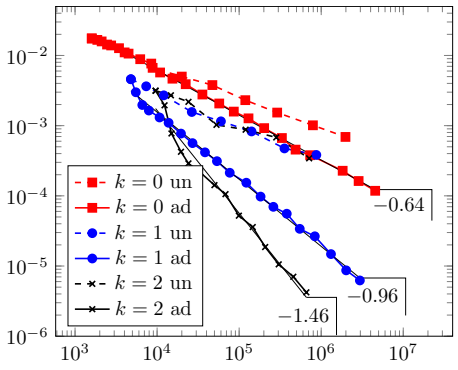


(f) E_h vs. N_{dof}

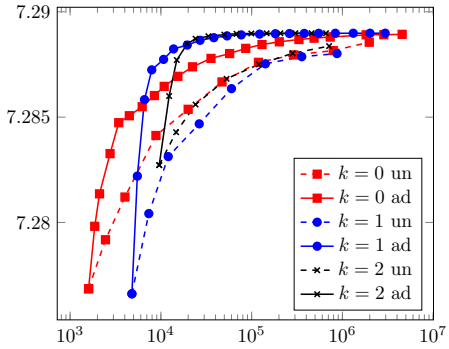
Figure 3: Results for the cube benchmark described in Section 4.1.1.



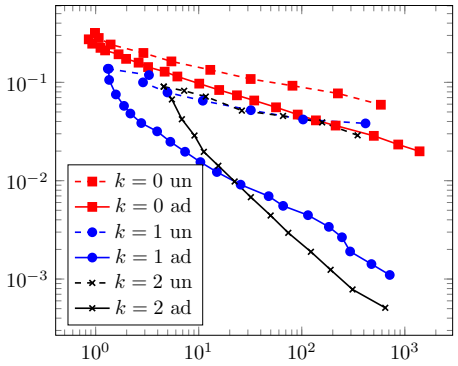
(a) e_{en} vs. N_{dof}



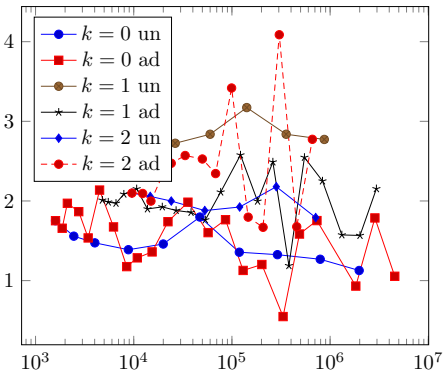
(b) e_{L^2} vs. N_{dof}



(c) E_h vs. N_{dof}



(d) e_{en} vs. computing time



(e) Efficiency index vs. N_{dof}

Figure 4: Results for the Fichera corner benchmark described in Section 4.1.2.

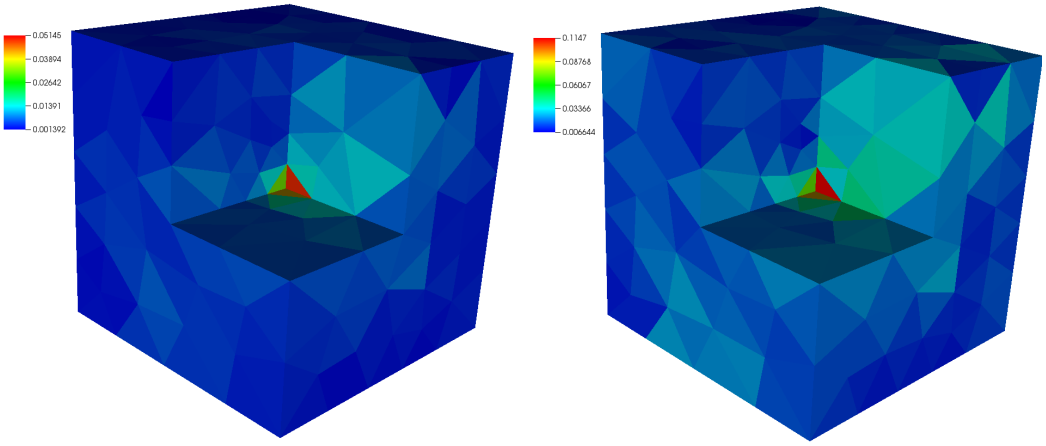


Figure 5: Distribution of the total error (left) and of its estimate (right) after two levels of mesh refinement for the Fichera corner benchmark described in Section 4.1.2.

4.1.3. Fichera corner benchmark (adaptive mesh coarsening)

The possibility to handle polyhedral elements offered by the MHO method paves the way to adaptive mesh coarsening in the spirit of [30]; cf. also [39]. The idea consists in starting from a fine mesh (chosen, e.g., to accurately represent the geometric features of the domain or to capture the finest scales in the solution) and solve the problem on a polyhedral mesh obtained by selectively merging the fine elements into polyhedral conglomerates. If the coarsening procedure is well-designed, one can achieve a precision comparable to that of a computation on the fine mesh, but for a smaller number of DOFs.

In our case, we start from a tetrahedral mesh consisting of 51,534 nodes and $2.72 \cdot 10^5$ tetrahedra, and we create an initial agglomerated mesh with a modified version of MGridGen [40] by setting a typical agglomeration target of 1,024 tetrahedra per coarse element. An adaptive mesh sequence is then generated by locally reducing the local meshsize by the same procedure as for the standard meshes considered in the previous section. Figure 7a shows on the left the initial polyhedral mesh and on the right the one obtained at the final refinement step. Figure 7b displays the fields u_h^* defined as in Remark 3 (i.e., by an averaging interpolation on the matching simplicial submesh) corresponding to the two meshes of Figure 7a.

In order to assess the efficiency of the adaptive coarsening procedure, we compare the results with a sequence of meshes obtained by uniform coarsening of the same initial mesh (i.e, the size of the agglomerated elements is not adapted in accordance with the distribution of the error). Figures 8a–b show the energy- and L^2-errors as functions of the number of DOFs on both the adaptively and uniformly agglomerated mesh sequences. In Figure 8, one can see that a similar precision in the energy-norm as the one achievable on the fine mesh is obtained for less than a half of the number of DOFs. The gain is even larger when considering the L^2-norm. Similar considerations hold for the error on the electrostatic energy depicted in Figure 8c. In all the cases, the error stagnates when the finest possible derefinement (corresponding to the elements of

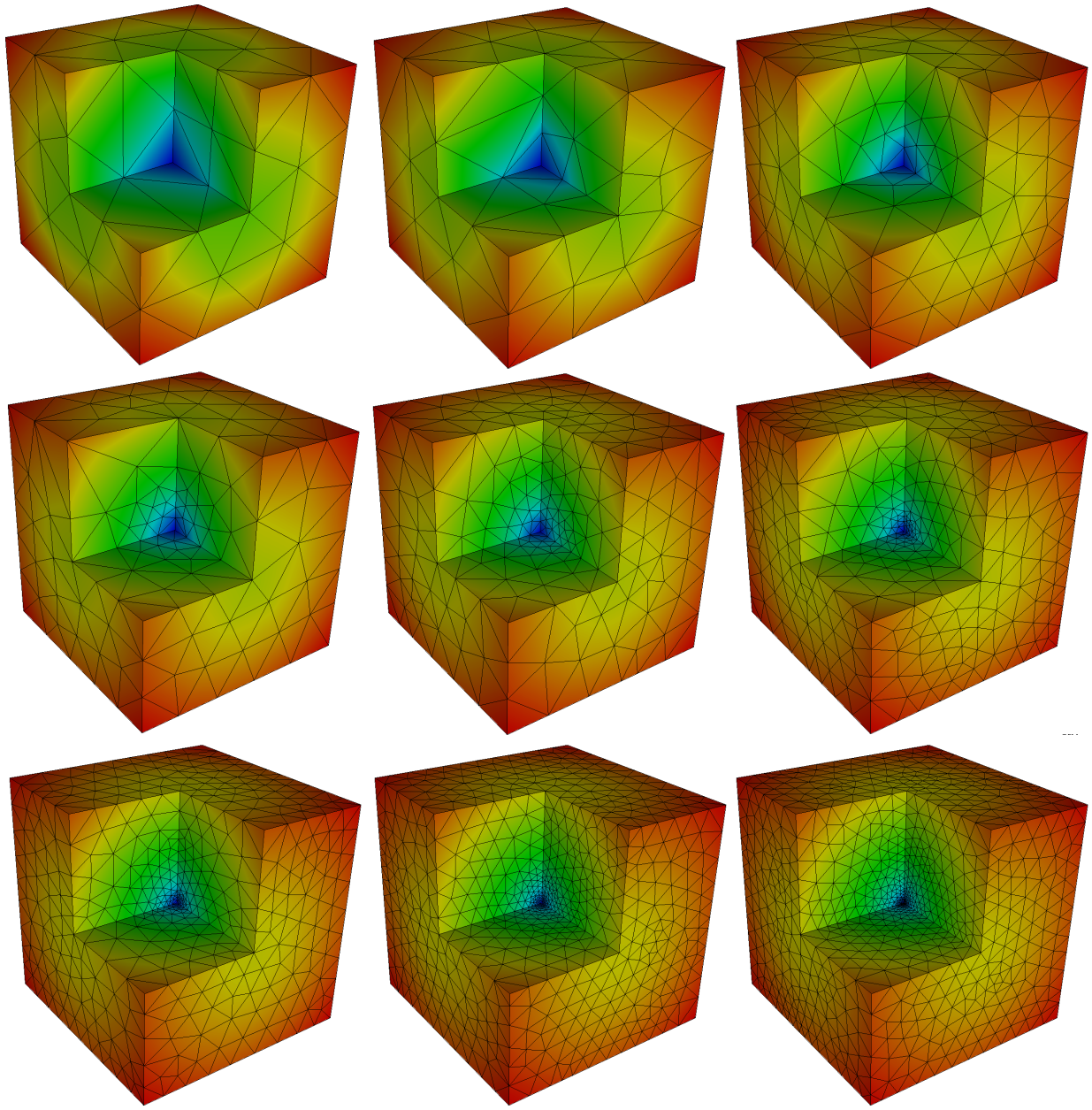
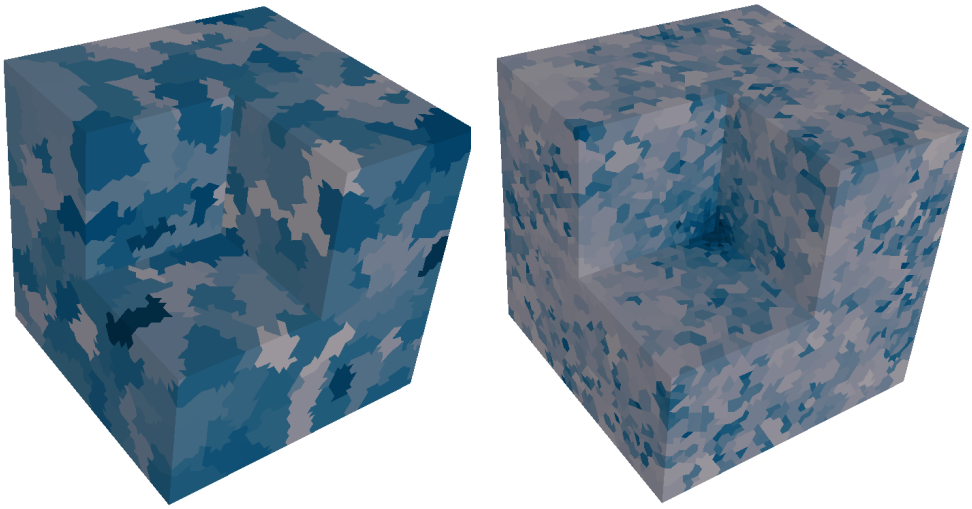
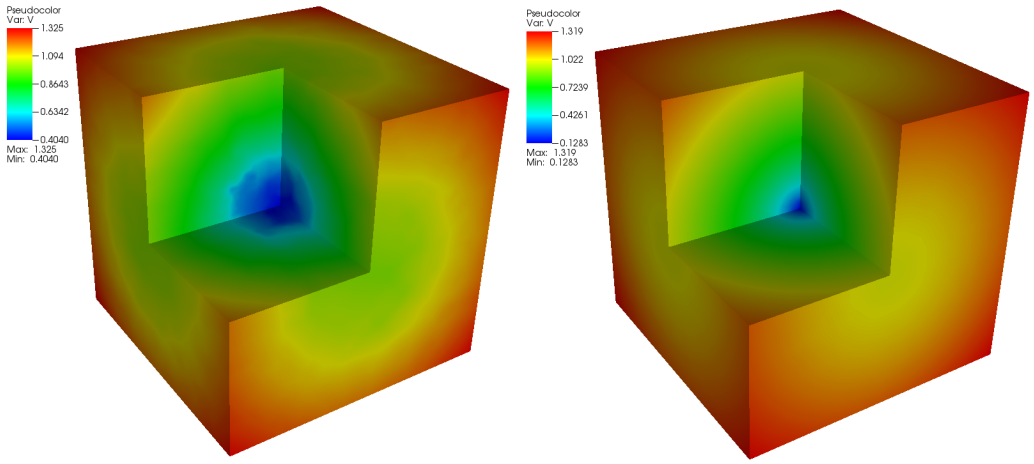


Figure 6: Example of eight meshes refined adaptively for the Fichera corner benchmark described in Section 4.1.2.

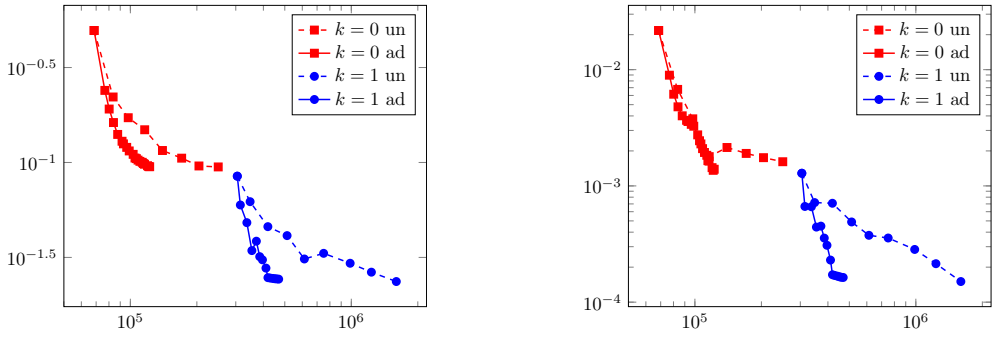


(a) On the left: Initial coarse mesh with agglomerated elements containing about 1,024 tetrahedral elements each. On the right: The mesh obtained after running the adaptivity refinement iterations.



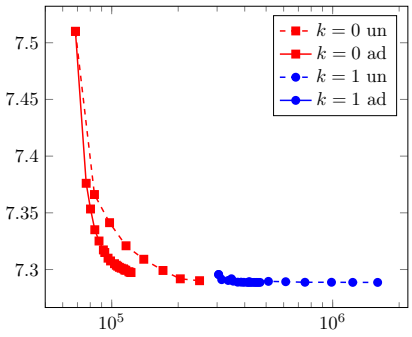
(b) On the left: Interpolated potential u_h^* (cf. Remark 3) on the first mesh of Figure 7a. On the right: Potential u_h^* on the second mesh of Figure 7a.

Figure 7: Results for the Fichera corner benchmark with adaptive mesh coarsening and $k = 0$, cf. Section 4.1.3.

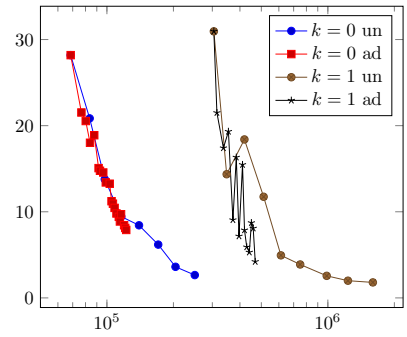


(a) e_{en} vs. N_{dof}

(b) e_{L^2} vs. N_{dof}



(c) E_h vs. N_{dof}



(d) Efficiency index vs. N_{dof}

Figure 8: Results for the Fichera corner benchmark (see Section 4.1.2) with agglomerated polyhedral elements.

the initial mesh) is achieved. In order to go further with the error reduction, one would need in this case to adaptively refine the initial mesh. Finally, we observe that the efficiency index depicted in Figure 8d follows a decreasing path and tends to 1 when refining.

4.2. Modeling of industrial devices

We next assess the performance of our adaptive method on two industrial applications: the modeling of a comb drive and of a MEMS device. The electric permittivity is assumed to be homogeneous and isotropic, i.e., $\boldsymbol{\varepsilon} = \varepsilon \boldsymbol{I}_d$, and we take for ε its value in the vacuum, $8.85 \cdot 10^{-12}$F/m.

4.2.1. Comb drive benchmark

Comb drives are capacitive actuators that exploit electrostatic forces between two electrically conductive combs. We solve the electrostatic problem described in [41], in which the potentials on two comb electrodes are set to 0V and 1V, respectively, see Figure 9. The two conductors are placed in the center of a cube of 1mm of edge. On the boundary of the cube a potential of 0V is set.

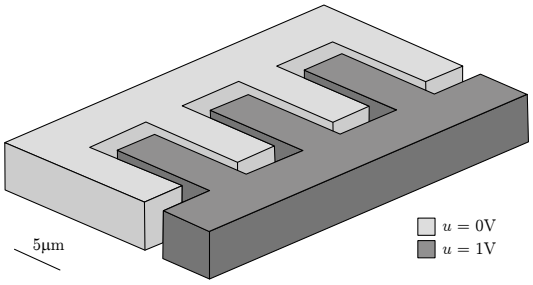
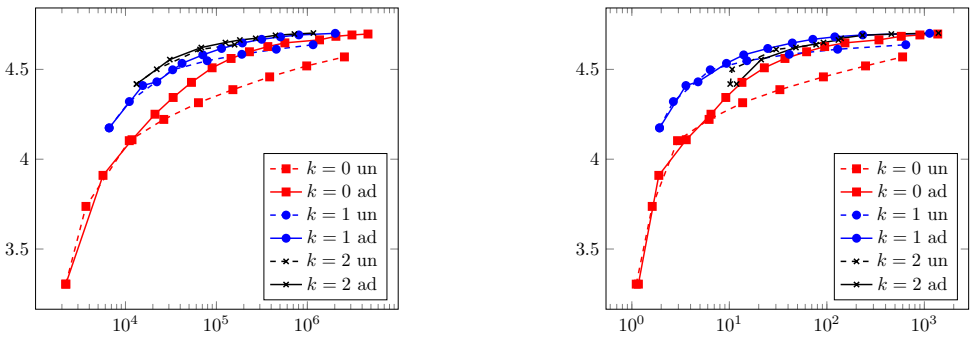


Figure 9: The geometry of the comb drive benchmark described in Section 4.2.1. Additional data to reproduce this benchmark can be found in [41].



(a) Capacitance vs. N_{dof}

(b) Capacitance vs. computing time

Figure 10: Results for the comb drive benchmark described in Section 4.2.1.

We monitor the numerical approximation of capacitance, i.e., the ability of the system to store electrostatic energy in the form of an electrical charge. In many applications arising in physics and engineering practice this is the most important quantity that the simulation estimates, see for example [3].

The total capacitance C is computed as $2\,E_h$, given that the voltage imposed between the two comb electrodes is 1V. Figures 10a–b show the convergence of the capacitance with respect to N_{dof} and the total computational wall time, respectively. Our results are in accordance with the ones presented in [41], and the accuracy has been largely improved thanks to the use of high-order approximations combined with adaptivity.

Figure 11 shows the discrete scalar potential and the corresponding local size of the mesh on a slice of the computational domain. Notice that refinement mainly happens around the corners, where singularities of the exact solution can be expected.

Finally, Figure 12 shows how the different components (pre-processing, assembly, solution and post-processing) contribute to the final computational wall time for $0 \leqslant k \leqslant 2$. We observe that for $k = 0$ the pre-processing is the dominating component, whereas for $k \geqslant 1$ most of the time is spent solving the global linear system.

4.2.2. Electrostatic MEMS switch benchmark

The second industrial test case is the capacitive MEMS switch benchmark described in [16], see Figure 13. The MEMS is composed by a perforated top plate (475 μm height, 275 μm width, thickness 4 μm) suspended by a set of beams over a bottom plate (485 μm height, 285 μm width, thickness 0.5 μm, gap 3 μm). The dimension of the holes in the top plate is $25 \times 25\mu$m with a pitch of 50 μm. The MEMS is placed in the center of a cube with edge length of 20 mm on which a potential of 0V is set. We solve the electrostatic problem in which the potentials on the two electrodes are set to 0V and 1V, respectively, see again Figure 13.

Figures 15a–b show the convergence of the capacitance with respect to N_{dof} and the total computational wall time, respectively. The obtained results are in accordance with the ones presented in [16], and, for a given precision, a gain in terms of computational wall-time is reported increasing the approximation order.

Figure 14 shows the discrete scalar potential and the corresponding local size of the mesh on a slice of the computational domain. Finally, Figure 16 shows how the different components (pre-processing, assembly, solution and post-processing) contribute to the final computational wall time for $0 \leqslant k \leqslant 2$. In this case, the algebraic multigrid solver displays a better scaling with respect to N_{dof}.

5. A posteriori error analysis

This section contains the proof of Theorem 2 preceeded by the required auxiliary results.

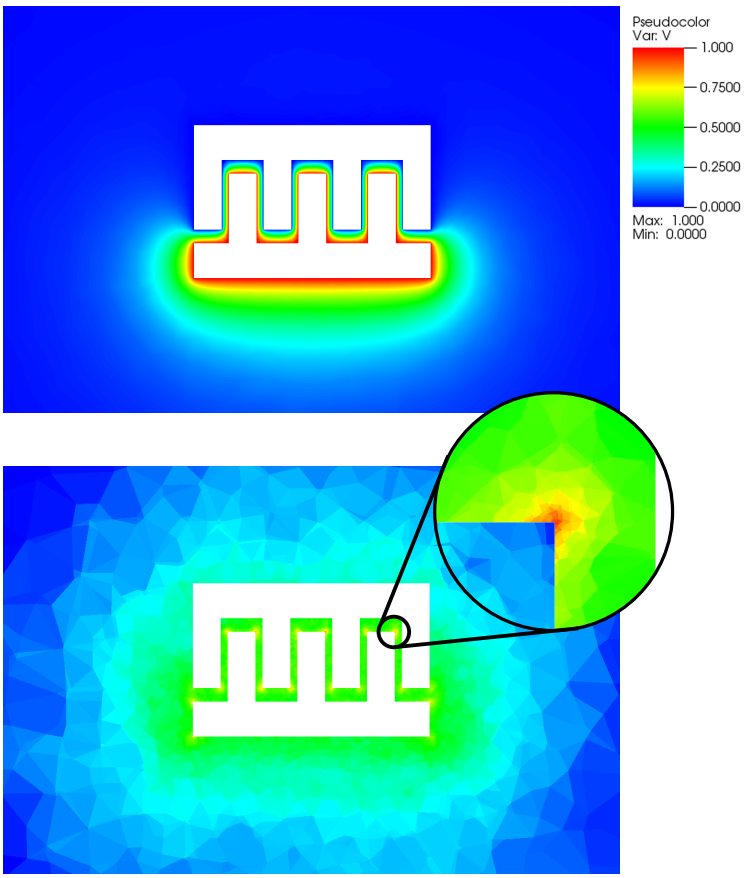
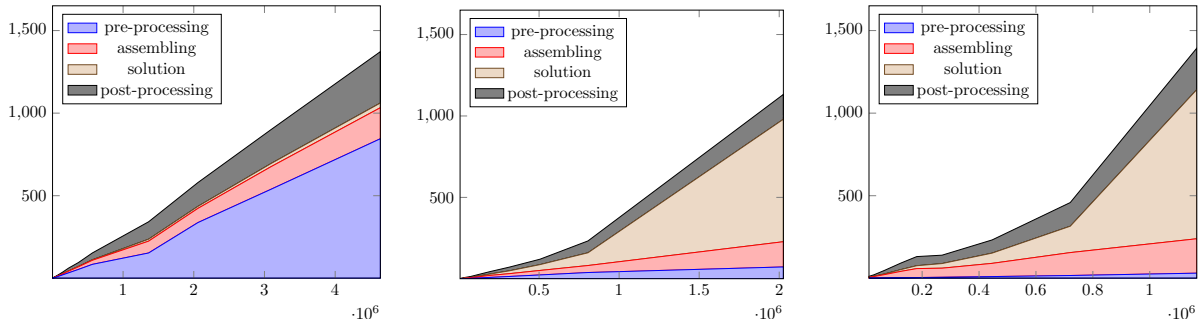


Figure 11: Results for the comb drive benchmark described in Section 4.2.1. Top: The potential in a slice of the computational domain. Bottom: The mesh is automatically adapted on suitable corners. The colour represents h_T^{-1} (arbitrary units).



(a) $k = 0$ (b) $k = 1$ (c) $k = 2$

Figure 12: Computing wall time vs N_{dof} for the comb drive benchmark described in Section 4.2.1.

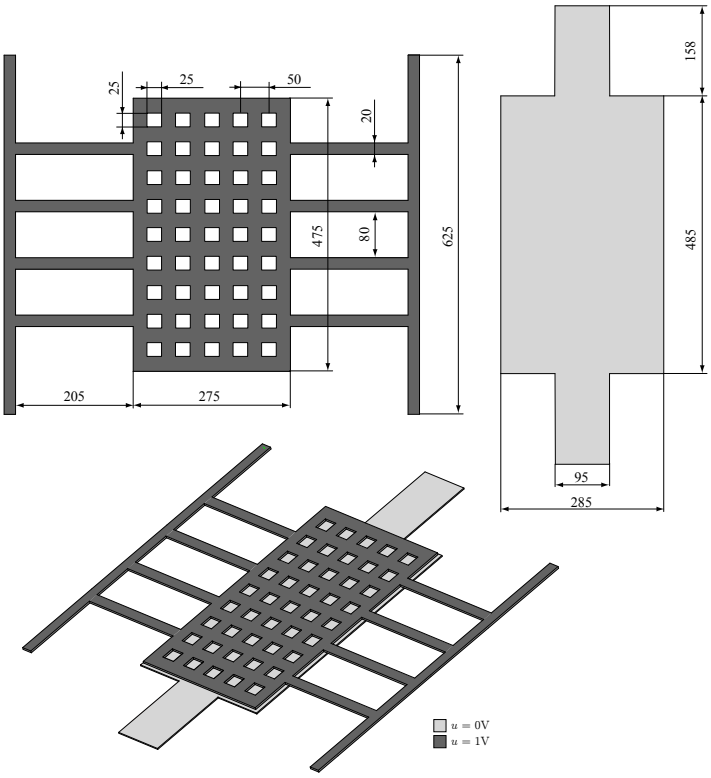


Figure 13: The geometry of the MEMS switch benchmark described in Section 4.2.2 (all lengths are in μm). Additional data to reproduce this benchmark can be found in [16].

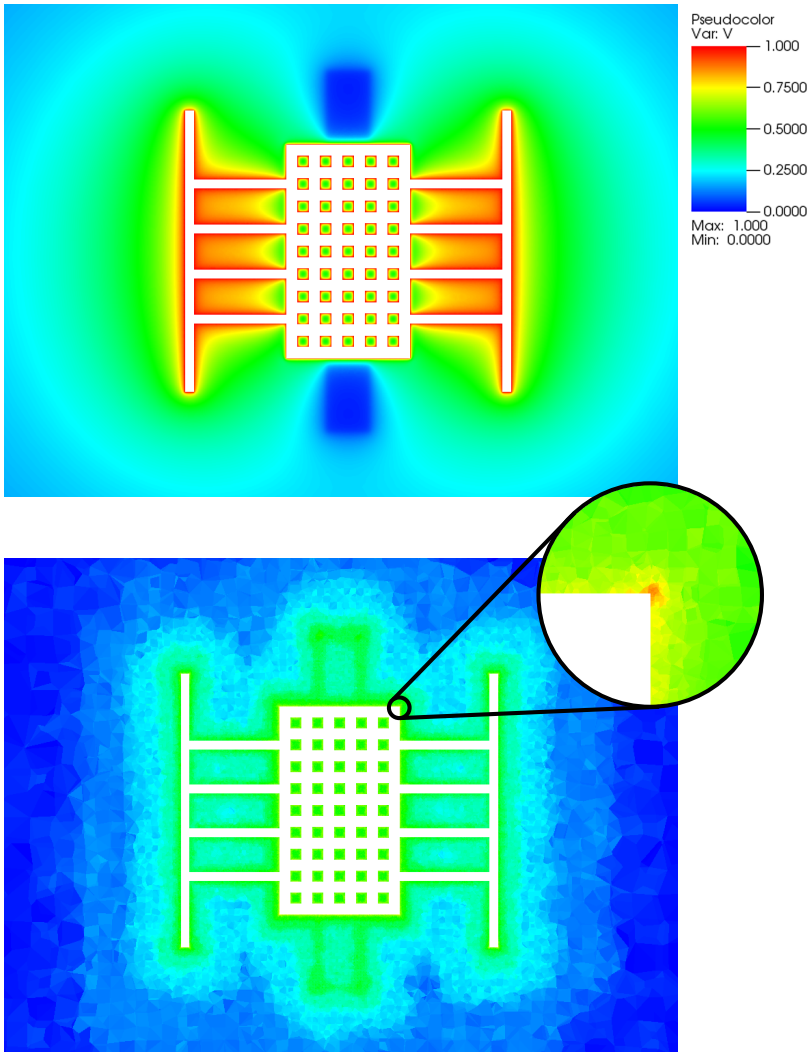


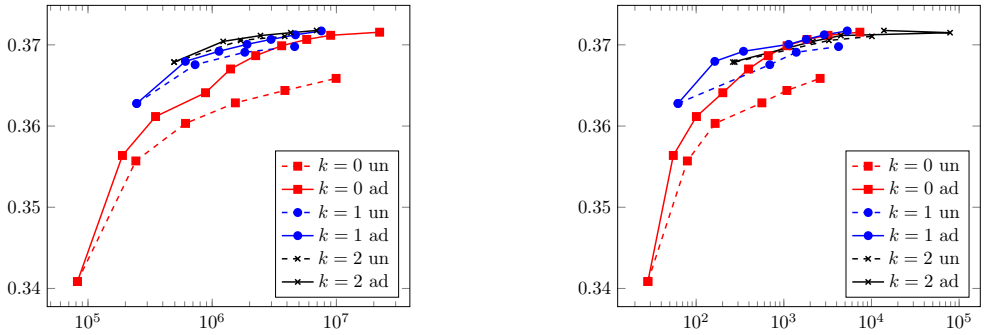
Figure 14: Results for the MEMS switch benchmark described in Section 4.2.2. Top: The potential in a slice of the computational domain. Bottom: The mesh is automatically adapted on suitable corners. The colour represents h_T^{-1} (arbitrary units).

5.1. Abstract estimate

We extend the bilinear form a defined by (12) to $H^1(\mathcal{T}_h) \times H^1(\mathcal{T}_h)$ by replacing the standard gradient operator $\boldsymbol{\nabla}$ by its broken counterpart $\boldsymbol{\nabla}_h$ defined in Section 3.3. Denoting by $u \in V$ the unique solution of the continuous primal problem (12), for a generic broken polynomial function $v_h \in \mathbb{P}^l(\mathcal{T}_h)$, $l \geqslant 1$, we define the residual $\mathcal{R}(v_h) \in H^{-1}(\Omega)$ such that, for all $\varphi \in H^1(\Omega)$,

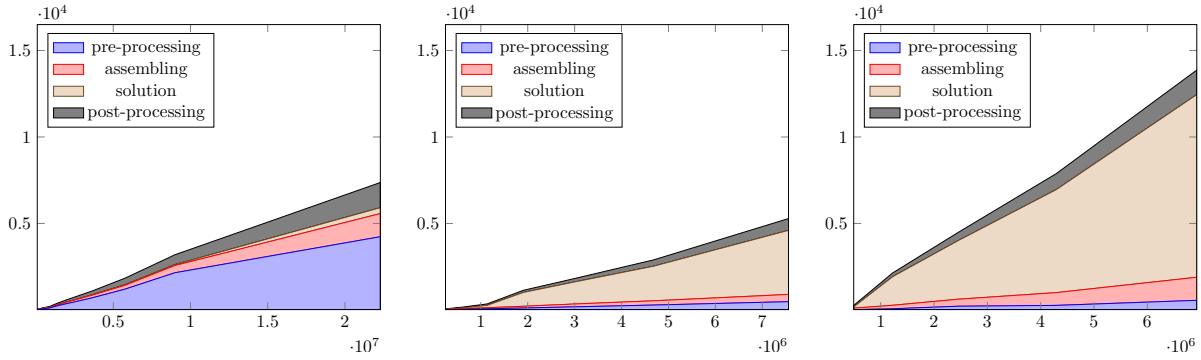
$$\langle \mathcal{R}(v_h), \varphi \rangle_{-1,1} := \mathrm{a}(u - v_h, \varphi) = (f, \varphi) - \mathrm{a}(v_h, \varphi), \tag{25}$$

where $\langle \cdot, \cdot \rangle_{-1,1}$ denotes the duality product between $H^{-1}(\Omega)$ and $H^1(\Omega)$. The following estimate is inspired by [22, Lemma 4.4]; cf. also [31, Lemma 5.44] and [42, Theorem 7.6].



(a) Capacitance vs. N_{dof} (b) Capacitance vs. computing time

Figure 15: Results for the MEMS switch benchmark described in Section 4.2.2.



(a) $k = 0$ (b) $k = 1$ (c) $k = 2$

Figure 16: Computing wall time vs N_{dof} for the MEMS switch benchmark described in Section 4.2.2.

Lemma 7 (Abstract estimate)**.** *Let $u \in V$ solve the continuous primal problem* (12)*. Then, for any $v_h \in \mathbb{P}^l(\mathcal{T}_h)$, $l \geqslant 1$, it holds*

$$\|\boldsymbol{K}^{1/2}\boldsymbol{\nabla}_h(u - v_h)\|^2 \leqslant \inf_{\varphi \in V} \|\boldsymbol{K}^{1/2}\boldsymbol{\nabla}_h(\varphi - v_h)\|^2 + \left\{ \sup_{\varphi \in V,\ \|\boldsymbol{K}^{1/2}\boldsymbol{\nabla}\varphi\| = 1} \langle \mathcal{R}(v_h), \varphi \rangle_{-1,1} \right\}^2. \tag{26}$$

Proof. Let $\psi \in V$ be such that

$$\mathrm{a}(\psi, z) = \mathrm{a}(v_h, z) \qquad \forall z \in V.$$

We observe that ψ is well-defined since the bilinear form a defined by (12) is V-coercive, and we classically have

$$\|\boldsymbol{K}^{1/2}\boldsymbol{\nabla}_h(v_h - \psi)\|^2 = \inf_{\varphi \in V} \|\boldsymbol{K}^{1/2}\boldsymbol{\nabla}_h(v_h - \varphi)\|^2. \tag{27}$$

Additionally,

$$\|\boldsymbol{K}^{1/2}\boldsymbol{\nabla}(u - \psi)\| \leqslant \sup_{\varphi \in V,\ \|\boldsymbol{K}^{1/2}\boldsymbol{\nabla}\varphi\| = 1} \mathrm{a}(u - \psi, \varphi) = \sup_{\varphi \in V,\ \|\boldsymbol{K}^{1/2}\boldsymbol{\nabla}\varphi\| = 1} \langle \mathcal{R}(v_h), \varphi \rangle_{-1,1}, \tag{28}$$

where the conclusion follows using the definition of ψ followed by (25). Finally, since $(v_h - \psi)$ is by definition a-orthogonal to functions in V, using the Pythagorean theorem, we have

$$\|\boldsymbol{K}^{1/2}\boldsymbol{\nabla}_h(u - v_h)\|^2 = \|\boldsymbol{K}^{1/2}\boldsymbol{\nabla}_h(v_h - \psi)\|^2 + \|\boldsymbol{K}^{1/2}\boldsymbol{\nabla}(u - \psi)\|^2. \tag{29}$$

To conclude, it suffices to use (27) and (28), respectively, to bound the terms in the right-hand side of (29). □

5.2. Reformulation of the stabilization bilinear form

To treat the stabilization term, we need a reformulation and a bound for the local stabilization bilinear form $s_{\boldsymbol{\Sigma},T}$ defined by (9). For use in the proof, we observe that, recalling the definition (2) of π_T^k and plugging the definition (6) of D_T^k into the definition (7) of $\boldsymbol{S}_T^k$, it holds for all $\underline{\boldsymbol{\tau}}_T \in \underline{\boldsymbol{\Sigma}}_T^k$ and all $w \in \mathbb{P}^{k+1}(T)$,

$$(\boldsymbol{S}_T^k\underline{\boldsymbol{\tau}}_T, \boldsymbol{\nabla}w)_T = (\boldsymbol{\tau}_T, \boldsymbol{\nabla}\pi_T^k w)_T + (\tau_{\partial T}, \pi_{\partial T}^k w - \pi_T^k w)_{\partial T}. \tag{30}$$

Additionally, we note the following characterization of p_T^{k+1} proved as in [21, Section 3.2.2]: For all $\underline{v}_T \in \underline{V}_T^k$ and all $w \in \mathbb{P}^{k+1}(T)$,

$$(\boldsymbol{K}_T\boldsymbol{\nabla}p_T^{k+1}\underline{v}_T, \boldsymbol{\nabla}w)_T = -(v_T, \boldsymbol{\nabla}\cdot(\boldsymbol{K}_T\boldsymbol{\nabla}w))_T + (v_{\partial T}, \boldsymbol{K}_T\boldsymbol{\nabla}w\cdot\boldsymbol{n}_{\partial T})_{\partial T}. \tag{31a}$$

$$= (\boldsymbol{K}_T\boldsymbol{\nabla}v_T, \boldsymbol{\nabla}w)_T + (v_{\partial T} - v_T, \boldsymbol{K}_T\boldsymbol{\nabla}w\cdot\boldsymbol{n}_{\partial T})_{\partial T}. \tag{31b}$$

We are now ready to state the following lemma.

Lemma 8 (Reformulation of $\mathrm{s}_{\boldsymbol{\Sigma},T}$)**.** *For all $\underline{\boldsymbol{\tau}}_T \in \underline{\boldsymbol{\Sigma}}_T^k$ and all $\underline{v}_T \in \underline{V}_T^k$, the bilinear form $\mathrm{s}_{\boldsymbol{\Sigma},T}$ defined by* (9) *satisfies*

$$\mathrm{s}_{\boldsymbol{\Sigma},T}(\underline{\boldsymbol{\tau}}_T, \underline{\boldsymbol{\varsigma}}_T^k\underline{v}_T) = (r_{\partial T}^{*,k}(\tau_{\partial T} - \boldsymbol{\tau}_T\cdot\boldsymbol{n}_{\partial T}), v_{\partial T} - v_T)_{\partial T}, \tag{32}$$

with scalar-to-vector variable operator $\underline{\boldsymbol{\varsigma}}_T^k$ defined by (13) *and adjoint boundary residual operator $r_{\partial T}^{*,k}$ defined by* (19)*. Additionally, it holds*

$$\|r_{\partial T}^{*,k}(\tau_{\partial T} - \boldsymbol{\tau}_T\cdot\boldsymbol{n}_{\partial T})\|_{\partial T} \leqslant \|\tau_{\partial T} - \boldsymbol{S}_T^k\underline{\boldsymbol{\tau}}_T\cdot\boldsymbol{n}_{\partial T}\|_{\partial T}. \tag{33}$$

Proof. Let $\underline{\boldsymbol{\tau}}_T \in \underline{\boldsymbol{\Sigma}}_T^k$ and $\underline{v}_T \in \underline{V}_T^k$ and set, for the sake of brevity,

$$\breve{v}_T := p_T^{k+1}\underline{v}_T, \qquad \widehat{v}_T := v_T + (\breve{v}_T - \pi_T^k\breve{v}_T). \tag{34}$$

Using the definitions (8) of the symmetric bilinear form m_T (with $\underline{\boldsymbol{\sigma}}_T = \underline{\boldsymbol{\varsigma}}_T^k\underline{v}_T$) and (13b) of the scalar-to-vector variable operator $\underline{\boldsymbol{\varsigma}}_T^k$, we get

$$(\boldsymbol{S}_T^k\underline{\boldsymbol{\tau}}_T, \boldsymbol{K}_T^{-1}\boldsymbol{S}_T^k(\underline{\boldsymbol{\varsigma}}_T^k\underline{v}_T))_T + \mathrm{s}_{\boldsymbol{\Sigma},T}(\underline{\boldsymbol{\tau}}_T, \underline{\boldsymbol{\varsigma}}_T^k\underline{v}_T) = (\boldsymbol{\tau}_T, \boldsymbol{\nabla}v_T)_T + (\tau_{\partial T}, v_{\partial T} - v_T)_{\partial T}. \tag{35}$$

Hence, rearranging the terms and recalling that, by virtue of (14) and (34),

$$\boldsymbol{K}_T^{-1}\boldsymbol{S}_T^k(\underline{\boldsymbol{\varsigma}}_T^k\underline{v}_T) = \boldsymbol{\nabla}p_T^{k+1}\underline{v}_T = \boldsymbol{\nabla}\breve{v}_T,$$

it is inferred

$$\begin{aligned}\mathrm{s}_{\boldsymbol{\Sigma},T}(\underline{\boldsymbol{\tau}}_T,\underline{\boldsymbol{\varsigma}}_T^k\underline{v}_T) &= (\boldsymbol{\tau}_T,\boldsymbol{\nabla}v_T)_T + (\tau_{\partial T},v_{\partial T}-v_T)_{\partial T} - (\boldsymbol{S}_T^k\underline{\boldsymbol{\tau}}_T,\boldsymbol{\nabla}\breve{v}_T)_T\\ &= (\boldsymbol{\tau}_T,\boldsymbol{\nabla}(v_T-\pi_T^k\breve{v}_T))_T + (\tau_{\partial T},v_{\partial T}-\pi_{\partial T}^k\breve{v}_T-v_T+\pi_T^k\breve{v}_T)_{\partial T}\\ &= (\boldsymbol{\tau}_T,\boldsymbol{\nabla}(\widehat{v}_T-\breve{v}_T))_T + (\tau_{\partial T},\pi_{\partial T}^k(v_{\partial T}-\widehat{v}_T))_{\partial T}\\ &= (\boldsymbol{\nabla}\cdot\boldsymbol{\tau}_T,\widehat{v}_T-v_T)_T + (\tau_{\partial T}-\boldsymbol{\tau}_T\cdot\boldsymbol{n}_{\partial T},\pi_{\partial T}^k(v_{\partial T}-\widehat{v}_T))_{\partial T},\end{aligned} \tag{36}$$

where we have used (30) with $w=\breve{v}_T$ to pass to the second line, equation (34) together with the fact that $\tau_{\partial T}\in\mathbb{P}^k(\mathcal{F}_T)$ to pass to the third, and the characterization (31a) of $\breve{v}_T=p_T^{k+1}\underline{v}_T$ with $\boldsymbol{K}_T\boldsymbol{\nabla}w=\boldsymbol{\tau}_T$ together with integration by parts for the term $(\widehat{v}_T,\boldsymbol{\nabla}\cdot\boldsymbol{\tau}_T)_T$ to conclude. Recalling (34) and observing that $\boldsymbol{\nabla}\cdot\boldsymbol{\tau}_T\in\mathbb{P}^{k-2}(T)$, it is readily inferred that $(\boldsymbol{\nabla}\cdot\boldsymbol{\tau}_T,\widehat{v}_T-v_T)_T=(\boldsymbol{\nabla}\cdot\boldsymbol{\tau}_T,\breve{v}_T-\pi_T^k\breve{v}_T)_T=0$ by definition (2) of π_T^k. Hence, (36) becomes

$$\mathrm{s}_{\boldsymbol{\Sigma},T}(\underline{\boldsymbol{\tau}}_T,\underline{\boldsymbol{\varsigma}}_T^k\underline{v}_T) = (\tau_{\partial T}-\boldsymbol{\tau}_T\cdot\boldsymbol{n}_{\partial T},\pi_{\partial T}^k(v_{\partial T}-\widehat{v}_T))_{\partial T}. \tag{37}$$

We now observe that

$$\begin{aligned}v_{\partial T}-\widehat{v}_T &= v_{\partial T}-v_T-p_T^{k+1}(v_T,v_{\partial T})+\pi_T^kp_T^{k+1}(v_T,v_{\partial T})\\ &= v_{\partial T}-v_T-\cancel{p_T^{k+1}(v_T,v_{T|\partial T})}-p_T^{k+1}(0,v_{\partial T}-v_{T|\partial T})+\cancel{\pi_T^kp_T^{k+1}(v_T,v_{T|\partial T})}\\ &\quad+\pi_T^kp_T^{k+1}(0,v_{\partial T}-v_{T|\partial T})\\ &= (v_{\partial T}-v_T)-p_T^{k+1}(0,v_{\partial T}-v_{T|\partial T})+\pi_T^kp_T^{k+1}(0,v_{\partial T}-v_{T|\partial T}),\end{aligned} \tag{38}$$

where we have used the definition (34) of $\widehat{v}_T$ in the first line and (31b) to infer that $p_T^{k+1}(v_T,v_{T|\partial T})=v_T$, hence also $p_T^{k+1}(v_T,v_{T|\partial T})-\pi_T^kp_T^{k+1}(v_T,v_{T|\partial T})=v_T-v_T=0$, so that the third and fifth terms in the second line cancel. Plugging (38) into the right-hand side of (37), and recalling the definitions (18) of the boundary residual operator $r_{\partial T}^k$ and (19) of its adjoint $r_{\partial T}^{*,k}$, we conclude that

$$\mathrm{s}_{\boldsymbol{\Sigma},T}(\underline{\boldsymbol{\tau}}_T,\underline{\boldsymbol{\varsigma}}_T^k\underline{v}_T) = (\tau_{\partial T}-\boldsymbol{\tau}_T\cdot\boldsymbol{n}_{\partial T},r_{\partial T}^k(v_{\partial T}-v_T))_{\partial T} = (r_{\partial T}^{*,k}(\tau_{\partial T}-\boldsymbol{\tau}_T\cdot\boldsymbol{n}_{\partial T}),v_{\partial T}-v_T)_{\partial T},$$

which proves (32).

To prove (33), we combine (35) with (32) and use (31b) for $\breve{v}_T=p_T^{k+1}\underline{v}_T$ (with $\boldsymbol{K}_T\boldsymbol{\nabla}w=\boldsymbol{S}_T^k\underline{\boldsymbol{\tau}}_T$) to infer

$$\begin{aligned}(r_{\partial T}^{*,k}(\tau_{\partial T}-\boldsymbol{\tau}_T\cdot\boldsymbol{n}_{\partial T}),v_{\partial T}-v_T)_{\partial T} &= (\boldsymbol{\tau}_T,\boldsymbol{\nabla}v_T)_T+(\tau_{\partial T},v_{\partial T}-v_T)_{\partial T}-(\boldsymbol{S}_T^k\underline{\boldsymbol{\tau}}_T,\boldsymbol{\nabla}\breve{v}_T)_T\\ &= (\boldsymbol{\tau}_T-\boldsymbol{S}_T^k\underline{\boldsymbol{\tau}}_T,\boldsymbol{\nabla}v_T)_T+(\tau_{\partial T}-\boldsymbol{S}_T^k\underline{\boldsymbol{\tau}}_T\cdot\boldsymbol{n}_{\partial T},v_{\partial T}-v_T)_{\partial T},\end{aligned}$$

which, letting $\underline{v}_T$ be such that $v_T=0$ and $v_{\partial T}=r_{\partial T}^{*,k}(\tau_{\partial T}-\boldsymbol{\tau}_T\cdot\boldsymbol{n}_{\partial T})$, yields

$$\|r_{\partial T}^{*,k}(\tau_{\partial T}-\boldsymbol{\tau}_T\cdot\boldsymbol{n}_{\partial T})\|_{\partial T}^2 = (\tau_{\partial T}-\boldsymbol{S}_T^k\underline{\boldsymbol{\tau}}_T\cdot\boldsymbol{n}_{\partial T},r_{\partial T}^{*,k}(\tau_{\partial T}-\boldsymbol{\tau}_T\cdot\boldsymbol{n}_{\partial T}))_{\partial T},$$

and (33) follows using the Cauchy–Schwarz inequality. □

5.3. Proof of Theorem 2

We are now ready to prove our main result.

Proof of Theorem 2. We invoke the abstract estimate (26) with $v_h = \breve{u}_h$ to infer

$$\|\boldsymbol{K}^{-1/2}(\boldsymbol{\sigma} - \breve{\boldsymbol{\sigma}}_h)\|^2 = \|\boldsymbol{K}^{1/2}\boldsymbol{\nabla}_h(u - \breve{u}_h)\|^2 \leqslant \sum_{T\in\mathcal{T}_h} \eta_{\mathrm{nc},T}^2 + \left\{ \sup_{\varphi\in V,\, \|\boldsymbol{K}^{1/2}\boldsymbol{\nabla}\varphi\|=1} \langle \mathcal{R}(\breve{u}_h), \varphi\rangle_{-1,1} \right\}^2. \tag{39}$$

To conclude, it suffices to estimate the last term. Let $\varphi \in V$. By definition (25) of the residual, we have,

$$\langle \mathcal{R}(\breve{u}_h), \varphi\rangle_{-1,1} = (f,\varphi) - \mathrm{a}(\breve{u}_h,\varphi) = (f,\varphi) - (\boldsymbol{K}\boldsymbol{\nabla}_h\breve{u}_h, \boldsymbol{\nabla}\varphi) = (f,\varphi) - (\breve{\boldsymbol{\sigma}}_h, \boldsymbol{\nabla}\varphi). \tag{40}$$

We proceed to find an upper bound for the last term in the right-hand side. We start by observing that

$$(f,\varphi) - (\breve{\boldsymbol{\sigma}}_h, \boldsymbol{\nabla}\varphi) = \sum_{T\in\mathcal{T}_h} \{(f,\varphi)_T - (\breve{\boldsymbol{\sigma}}_T, \boldsymbol{\nabla}\varphi)_T\} = \sum_{T\in\mathcal{T}_h} \{(f + \boldsymbol{\nabla}\cdot\breve{\boldsymbol{\sigma}}_T, \varphi)_T - (\breve{\boldsymbol{\sigma}}_T\boldsymbol{n}_{\partial T}, \varphi)_{\partial T}\}, \tag{41}$$

where we have used element-by-element integration by parts to conclude. Let now $\underline{\varphi}_T \in \underline{V}_{h,0}^k$ be such that, for all $T \in \mathcal{T}_h$,

$$\varphi_T := \pi_T^0\varphi, \qquad \varphi_{\partial T} := \pi_{\partial T}^k\varphi. \tag{42}$$

We have that

$$\begin{aligned}
\sum_{T\in\mathcal{T}_h} (\pi_T^0(f + \boldsymbol{\nabla}\cdot\breve{\boldsymbol{\sigma}}_T), \varphi)_T &= \sum_{T\in\mathcal{T}_h} \{(f,\varphi_T)_T + (\boldsymbol{\nabla}\cdot\breve{\boldsymbol{\sigma}}_T, \varphi_T)_T\} \\
&= \sum_{T\in\mathcal{T}_h} \left\{ \mathrm{a}_T(\underline{u}_T, \underline{\varphi}_T) - \cancel{(\breve{\boldsymbol{\sigma}}_T, \boldsymbol{\nabla}\varphi_T)_T} + (\breve{\boldsymbol{\sigma}}_T\boldsymbol{n}_{\partial T}, \varphi_T)_{\partial T} \right\} \\
&= \sum_{T\in\mathcal{T}_h} \left\{ (\boldsymbol{K}_T\boldsymbol{\nabla}\breve{u}_T, \boldsymbol{\nabla}p_T^{k+1}\underline{\varphi}_T)_T + \mathrm{s}_{U,T}(\underline{u}_T, \underline{\varphi}_T) + (\breve{\boldsymbol{\sigma}}_T\boldsymbol{n}_{\partial T}, \varphi_T)_{\partial T} \right\} \\
&= \sum_{T\in\mathcal{T}_h} \left\{ \cancel{(\boldsymbol{K}_T\boldsymbol{\nabla}\breve{u}_T, \boldsymbol{\nabla}\varphi_T)_T} + (\breve{\boldsymbol{\sigma}}_T\boldsymbol{n}_{\partial T}, \varphi_{\partial T})_{\partial T} + \mathrm{s}_{\boldsymbol{\Sigma},T}(\underline{\boldsymbol{\sigma}}_T, \underline{\boldsymbol{\varsigma}}_T^k\underline{\varphi}_T) \right\},
\end{aligned}$$

where we have used the primal reformulation (15) of problem (11) together with integration by parts and $\boldsymbol{\nabla}\varphi_T = \boldsymbol{\nabla}\pi_T^0\varphi = 0$ in the second line, the definition (16) of the bilinear form a_T in the third line, and the characterization (31b) of p_T^{k+1} with $\underline{v}_T = \underline{\varphi}_T$ and $w = \breve{u}_T$ together with $\boldsymbol{K}_T\boldsymbol{\nabla}\breve{u}_T = \breve{\boldsymbol{\sigma}}_T$ and the definition (16) of $\mathrm{s}_{U,T}$ and $\underline{\boldsymbol{\sigma}}_T = \underline{\boldsymbol{\varsigma}}_T^k\underline{u}_T$ (cf. (15a)) to conclude. Hence, observing that $\breve{\boldsymbol{\sigma}}_{T|\partial T}\boldsymbol{n}_{\partial T} \in \mathbb{P}^k(\mathcal{F}_T)$ (so that $(\breve{\boldsymbol{\sigma}}_T\boldsymbol{n}_{\partial T}, \varphi_{\partial T})_{\partial T} = (\breve{\boldsymbol{\sigma}}_T\boldsymbol{n}_{\partial T}, \varphi)_{\partial T}$ by (42) and the definition of $\pi_{\partial T}^k$) and rearranging, we have

$$0 = \sum_{T\in\mathcal{T}_h} (\pi_T^0(f + \boldsymbol{\nabla}\cdot\breve{\boldsymbol{\sigma}}_T), \varphi)_T - \sum_{T\in\mathcal{T}_h} \left\{ (\breve{\boldsymbol{\sigma}}_T\boldsymbol{n}_{\partial T}, \varphi)_{\partial T} + \mathrm{s}_{\boldsymbol{\Sigma},T}(\underline{\boldsymbol{\sigma}}_T, \underline{\boldsymbol{\varsigma}}_T^k\underline{\varphi}_T) \right\}. \tag{43}$$

Subtracting (43) from (41), and using formula (32) proved in Lemma 8 with $\underline{v}_T = \underline{\varphi}_T$ and $\underline{\boldsymbol{\tau}}_T = \underline{\boldsymbol{\sigma}}_T$ for the term involving $\mathrm{s}_{\boldsymbol{\Sigma},T}$, it is inferred

$$(\boldsymbol{\sigma} - \breve{\boldsymbol{\sigma}}_h, \boldsymbol{\nabla}\varphi) = \sum_{T\in\mathcal{T}_h} \left\{ ((f + \boldsymbol{\nabla}\cdot\breve{\boldsymbol{\sigma}}_T) - \pi_T^0(f + \boldsymbol{\nabla}\cdot\breve{\boldsymbol{\sigma}}_T), \varphi - \pi_T^0\varphi)_T + (r_{\partial T}^{*,k}(\sigma_{\partial T} - \boldsymbol{\sigma}_T\boldsymbol{n}_{\partial T}), \varphi - \pi_T^0\varphi)_{\partial T} \right\},$$

where we have used the fact that $r_{\partial T}^{*,k}(\sigma_{\partial T} - \boldsymbol{\sigma}_T \cdot \boldsymbol{n}_{\partial T}) \in \mathbb{P}^k(\mathcal{F}_T)$ and (42) to write φ instead of $\varphi_{\partial T}$ in the last term. Therefore, using the Cauchy–Schwarz inequality followed by the local Poincaré (20) and Friedrichs (21) inequalities for the first and second terms, respectively, it is inferred

$$|(\boldsymbol{\sigma} - \breve{\boldsymbol{\sigma}}_h, \boldsymbol{\nabla}\varphi)| \leqslant \left\{ \sum_{T \in \mathcal{T}_h} \left(\eta_{\mathrm{res},T} + \eta_{\mathrm{sta},T}\right)^2 \right\}^{1/2} \times \|\boldsymbol{K}^{1/2}\boldsymbol{\nabla}\varphi\|. \tag{44}$$

To conclude, use (44) to estimate the right-hand side of (40) and plug the resulting bound into (39). □

Acknowledgements. The authors are grateful to Lorenzo Botti (Università di Bergamo) for providing the modified version of the MGridGen library used in the test cases of Section 4.1.3.

References

[1] D. A. Di Pietro, A. Ern, A family of arbitrary-order mixed methods for heterogeneous anisotropic diffusion on general meshes, IMA J. Numer. Anal.Published online. doi:10.1093/imanum/drw003.

[2] D. A. Di Pietro, A. Ern, S. Lemaire, Building bridges: Connections and challenges in modern approaches to numerical partial differential equations, Springer, 2016, Ch. A review of Hybrid High-Order methods: formulations, computational aspects, comparison with other methods, to appear.

[3] R. Specogna, Extraction of vlsi multiconductor transmission line parameters by complementarity, IEEE Trans. Very Large Scale Integr. (VLSI) Syst. 22 (1) (2014) 146–154. doi:10.1109/TVLSI.2012.2232320.

[4] X. Xu, Z. Ren, H. Qu, D. Ren, 3-d ic interconnect capacitance extraction using dual discrete geometric methods with prism elements, IEEE Transactions on Very Large Scale Integration (VLSI) Systems 24 (4) (2016) 1524–1534. doi:10.1109/TVLSI.2015.2459043.

[5] M. Soleimani, W. Lionheart, Nonlinear image reconstruction for electrical capacitance tomography using experimental data, Meas. Sci. Technol. 16 (10) (2005) 1987.

[6] A. De Lorenzi, L. Grando, A. Pesce, P. Bettini, R. Specogna, Modeling of epoxy resin spacers for the 1 mv dc gas insulated line of iter neutral beam injector system, IEEE Trans. Dielectr. Electr. Insul. 16 (1) (2009) 77–87. doi:10.1109/TDEI.2009.4784554.

[7] P. Bettini, E. Brusa, M. Munteanu, R. Specogna, F. Trevisan, Innovative numerical methods for non-linear MEMS: the non-incremental FEM vs. the discrete geometric approach., CMES, Comput. Model. Eng. Sci. 33 (3) (2008) 215–242.

[8] A. Paussa, R. Specogna, D. Esseni, F. Trevisan, Discrete geometric approach for modelling quantization effects in nanoscale electron devices, J. Comput. Electron. 13 (1) (2014) 287–299. doi:10.1007/s10825-013-0523-2.

[9] D. Xie, New solution decomposition and minimization schemes for Poisson–Boltzmann equation in calculation of biomolecular electrostatics, J. Comput. Phys. 275 (2014) 294 – 309. doi:10.1016/j.jcp.2014.07.012.

[10] J. Ying, D. Xie, A new finite element and finite difference hybrid method for computing electrostatics of ionic solvated biomolecule, J. Comput. Phys. 298 (2015) 636 – 651. doi:10.1016/j.jcp.2015.06.016.

[11] M. Cheney, D. Isaacson, J. Newell, Electrical impedance tomography, SIAM Rev. 41 (1) (1999) 85–101. doi:10.1137/S0036144598333613.

[12] P. Bettini, R. Specogna, Computation of stationary 3d halo currents in fusion devices with accuracy control, J. Comput. Phys. 273 (2014) 100 – 117. doi:10.1016/j.jcp.2014.04.060.

[13] J. Jin, The Finite Element Method in Electromagnetics, 3rd Edition, Wiley-IEEE Press, 2014.

[14] M. Marrone, Properties of constitutive matrices for electrostatic and magnetostatic problems, IEEE Trans. Magn. 40 (3) (2004) 1516–1520. doi:10.1109/TMAG.2004.827175.

[15] U. van Rienen, Numerical Methods in Computational Electrodynamics - Linear Systems in Practical Applications., Vol. 12 of Lecture Notes in Computational Science and Engineering, Springer, 2001.

[16] R. Specogna, Complementary geometric formulations for electrostatics, Int. J. Numer. Meth. Eng. 86 (8) (2011) 1041–1068. doi:10.1002/nme.3089.

[17] E. Tonti, The Mathematical Structure of Classical and Relativistic Physics: A General Classification Diagram, Modeling and Simulation in Science, Engineering and Technology, Springer New York, 2013.

[18] Y. Cao, Y. Chu, X. He, T. Lin, An iterative immersed finite element method for an electric potential interface problem based on given surface electric quantity, J. Comput. Phys. 281 (2015) 82 – 95. doi:10.1016/j.jcp.2014.10.014.

[19] A. Bardazzi, C. Lugni, M. Antuono, G. Graziani, O. Faltinsen, Generalized {HPC} method for the poisson equation, J. Comput. Phys. 299 (2015) 630 – 648. doi:10.1016/j.jcp.2015.07.026.

[20] L. Codecasa, R. Specogna, F. Trevisan, A new set of basis functions for the discrete geometric approach, J. Comput. Phys. 19 (299) (2010) 7401–7410.

[21] J. Aghili, S. Boyaval, D. A. Di Pietro, Hybridization of mixed high-order methods on general meshes and application to the Stokes equations, Comput. Meth. Appl. Math. 15 (2) (2015) 111–134. doi:10.1515/cmam-2015-0004.

[22] K. Y. Kim, A posteriori error estimators for locally conservative methods of nonlinear elliptic problems, Appl. Numer. Math. 57 (9) (2007) 1065–1080.

[23] M. Ainsworth, J. T. Oden, A posteriori error estimation in finite element analysis, Pure and Applied Mathematics (New York), Wiley-Interscience [John Wiley & Sons], New York, 2000.

[24] M. Vohralík, A posteriori error estimates for lowest-order mixed finite element discretizations of convection-diffusion-reaction equations, SIAM J. Numer. Anal. 45 (4) (2007) 1570–1599.

[25] D. A. Di Pietro, A. Ern, S. Lemaire, An arbitrary-order and compact-stencil discretization of diffusion on general meshes based on local reconstruction operators, Comput. Meth. Appl. Math. 14 (4) (2014) 461–472. doi:10.1515/cmam-2014-0018.

[26] D. A. Di Pietro, A. Ern, A hybrid high-order locking-free method for linear elasticity on general meshes, Comput. Meth. Appl. Mech. Engrg. 283 (2015) 1–21. doi:10.1016/j.cma.2014.09.009.

[27] B. Cockburn, W. Zhang, A posteriori error analysis for Hybridizable Discontinuous Galerkin methods for second order elliptic problems, SIAM J. Numer. Anal. 51 (1) (2013) 676–693.

[28] L. Beirão da Veiga, M. Manzini, Residual *a posteriori* error estimation for the virtual element method for elliptic problems, ESAIM Math. Model. Numer. Anal. 49 (2) (2015) 577–599.

[29] A. Cangiani, E. H. Georgoulis, T. Pryer, O. J. Sutton, A posteriori error estimates for the virtual element method, submitted (2016).
URL http://arxiv.org/abs/1603.05855

[30] F. Bassi, L. Botti, A. Colombo, D. A. Di Pietro, P. Tesini, On the flexibility of agglomeration based physical space discontinuous Galerkin discretizations, J. Comput. Phys. 231 (1) (2012) 45–65. doi:10.1016/j.jcp.2011.08.018.

[31] D. A. Di Pietro, A. Ern, Mathematical Aspects of Discontinuous Galerkin Methods, Vol. 69 of Mathématiques & Applications, Springer, Berlin Heidelberg, 2012.

[32] B. Cockburn, D. A. Di Pietro, A. Ern, Bridging the Hybrid High-Order and Hybridizable Discontinuous Galerkin methods, ESAIM: Math. Model. Numer. Anal. (M2AN) 50 (3) (2016) 635–650. doi:10.1051/m2an/2015051.

[33] L. E. Payne, H. F. Weinberger, An optimal Poincaré inequality for convex domains, Arch. Rational Mech. Anal. 5 (1960) 286–292.

[34] M. Bebendorf, A note on the Poincaré inequality for convex domains, Z. Anal. Anwendungen 22 (4) (2003) 751–756.

[35] R. Verfürth, A review of a posteriori error estimation and adaptive mesh-refinement techniques, Teubner-Wiley, Stuttgart, 1996.

[36] J. Schöberl, Netgen an advancing front 2d/3d-mesh generator based on abstract rules, Comput. Vis. Sci. 1 (1) 41–52. doi:10.1007/s007910050004.

[37] Y. Notay, An aggregation-based algebraic multigrid method, Electron. Trans. Numer. Anal. 37 (6) (2010) 123–146.

[38] G. Fichera, Asymptotic behaviour of the electric field and density of the electric charge in the neighbourhood of singular points of a conducting surface, Russian Math. Surveys 30 (3) (1975) 107.

[39] P. F. Antonietti, S. Giani, P. Houston, *hp*-version composite discontinuous Galerkin methods for elliptic problems on complicated domains, SIAM J. Sci. Comput 35 (3) (2013) A1417–A1439.

[40] I. Moulitsas, G. Karypis, MGridGen/ParmGridGen, Serial/Parallel library for generating coase meshes for multigrid methods, Technical Report Version 1.0, University of Minnesota, Department of Computer Science/Army HPC Research Center (2001).

[41] Z. Ren, X. Xu, Dual discrete geometric methods in terms of scalar potential on unstructured mesh in electrostatics, IEEE Trans. Magn. 50 (2) (2014) 37–40. doi:10.1109/TMAG.2013.2280452.

[42] M. Vohralík, A posteriori error estimates for efficiency and control in numerical simulations, université Pierre et Marie Curie. Lecture notes from the course NM497 (February 2015).