Journal of Computational Physics 397 (2019) 108837

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Adaptive boundary element methods for the computation of the electrostatic capacity on complex polyhedra

Timo Betcke[a], Alexander Haberl[b,*], Dirk Praetorius[b]

[a] *University College London, Department of Mathematics, Gower Street, London WC1E 6BT, UK*
[b] *TU Wien, Institute for Analysis and Scientific Computing, Wiedner Hauptstr. 8-10 / E101 / 4, 1040 Wien, Austria*

ARTICLE INFO

Article history:
Received 24 January 2019
Received in revised form 2 July 2019
Accepted 15 July 2019
Available online 22 July 2019

Keywords:
Electrostatic capacity
Boundary integral equations
Adaptivity
Operator preconditioning

ABSTRACT

The accurate computation of the electrostatic capacity of three dimensional objects is a fascinating benchmark problem with a long and rich history. In particular, the capacity of the unit cube has widely been studied, and recent advances allow to compute its capacity to more than ten digits of accuracy. However, the accurate computation of the capacity for general three dimensional polyhedra is still an open problem. In this paper, we propose a new algorithm based on a combination of ZZ-type *a posteriori* error estimation and effective operator preconditioned boundary integral formulations to easily compute the capacity of complex three dimensional polyhedra to 5 digits and more. While this paper focuses on the capacity as a benchmark problem, it also discusses implementational issues of adaptive boundary element solvers, and we provide codes based on the boundary element package Bempp to make the underlying techniques accessible to a wide range of practical problems.

1. Introduction

1.1. The capacity problem

The capacity $\mathrm{Cap}(\Omega)$ of an isolated conductor $\Omega \subset \mathbb{R}^3$ measures its ability to store charges. It is defined as the ratio of the total surface equilibrium charge relative to its surface potential value [30]. To compute the capacity, we therefore need to solve the following exterior Laplace problem for the equilibrium potential u with unit surface value:

$$-\Delta u = 0 \qquad \text{in } \mathbb{R}^3 \backslash \overline{\Omega}, \tag{1a}$$

$$u = 1 \qquad \text{on } \Gamma := \partial\Omega, \tag{1b}$$

$$|u(x)| = \mathcal{O}\left(|x|^{-1}\right) \qquad \text{as } |x| \to \infty. \tag{1c}$$

The total surface charge of an isolated conductor is then given by Gauss' law as

$$\mathrm{Cap}^*(\Omega) = -\epsilon_0 \int\limits_{\Gamma} \frac{\partial u}{\partial \nu}(x)\, d\Gamma(x). \tag{2}$$

[*] Corresponding author.
E-mail addresses: t.betcke@ucl.ac.uk (T. Betcke), Alexander.Haberl@asc.tuwien.ac.at (A. Haberl), Dirk.Praetorius@asc.tuwien.ac.at (D. Praetorius).

https://doi.org/10.1016/j.jcp.2019.07.036
0021-9991/© 2019 Elsevier Inc. All rights reserved.

Here, $\nu(x)$ is the outward pointing normal vector for $x \in \Gamma$, and ϵ_0 is the electric constant with value $\epsilon_0 \approx 8.854 \times 10^{-12}$ F/m. In the rest of the paper, we will use the normalized capacity $\mathrm{Cap}(\Omega) = -\frac{1}{4\pi}\int_\Gamma \partial u/\partial \nu \, d\Gamma$, which is commonly used in the literature.

1.2. State of the art

Analytic expressions of the capacity in 3D are only known for very few simple domains, such as a sphere with radius r, for which $\mathrm{Cap}(\Omega) = r$. Moreover, computing the capacity to high accuracy even for simple shapes such as the unit cube is exceedingly difficult as it involves the solution of the exterior Laplace problem (1) for (possibly) non-smooth domains. This is very different from the 2D case, where techniques such as fast Schwarz-Christoffel maps [6,9] allow the computation of the logarithmic capacity to many digits of accuracy even on complex domains.

Nevertheless, there have been a range of interesting developments over the years to compute the capacity in three dimensions. A frequently used benchmark example is the capacity of the unit cube, which we will also use in this paper to compare our results to existing methods. An early bound for the normalized capacity C of the unit cube was published by Pólya already in 1947 [34] who estimated that

$$0.62033 < C < 0.71055.$$

Over the years, several improvements have been made, some of which are summarized in [27]. In that paper, techniques based on random walks are used to compute the capacity as 0.66067813 with an error believed to be in the order of $\pm 1.01 \times 10^{-7}$. In 2013, [28] improved the existing computations by using Nyström methods and multilevel solvers to 0.66067815409957 with an error in the order of 10^{-13}.

Adaptive boundary element computations for the capacity are not completely new. For example, in [35] a pre-chosen anisotropic refinement towards the edges is used together with an extrapolation technique to compute the capacity of the unit cube to around six digits of accuracy. However, the computations in that paper are simplified by exploiting the special symmetry of the cube and do not generalize to arbitrary polyhedra. Moreover, [35] briefly mentions also an adaptive refinement strategy that is based on refining elements with large charge contributions.

1.3. Contributions of the present work

In this paper, we present a black-box method for capacity computations of polyhedra in three dimensions, which achieves a similar accuracy of order 10^{-6} for the unit cube. Our method is based on an adaptive boundary element computation, which uses a ZZ-type *a posteriori* error estimator to steer the mesh-refinement in combination with a suitable operator preconditioning. Our ansatz is completely generic in that the adaptive refinement strategy works for any type of polyhedron and quickly generates meshes that compute the capacity for a given shape to several digits of accuracy.

1.4. Main results in a nutshell

A boundary integral formulation for the computation of the capacity can be derived by considering Green's representation theorem and noting that the exterior Laplace double-layer potential is zero for constant densities. We hence obtain a representation of the solution u of (1) as

$$u(x) = \int_\Gamma G(x, y)\phi(y)\, d\Gamma(y) \quad \text{for all } x \in \mathbb{R}^3 \backslash \overline{\Omega}, \tag{3}$$

where $\phi := -\partial u/\partial \nu$ is the (negative) normal derivative of the exterior solution u and $G(x, y) := \frac{1}{4\pi |x-y|}$ is the Green's function of the 3D Laplacian. Taking boundary traces, the right-hand side of (3) gives rise to the Laplace single-layer integral operator V, and we arrive at the boundary integral equation of the first kind

$$1 = \int_\Gamma G(x, y)\phi(y)\, d\Gamma(y) =: [V\phi](x) \quad \text{for all } x \in \Gamma. \tag{4}$$

Then,

$$\mathrm{Cap}(\Omega) = \frac{1}{4\pi}\int_\Gamma \phi \, d\Gamma. \tag{5}$$

The challenge is to accurately compute the solution ϕ of (4) close to edges and corners of a polyhedron. High-order collocation or Nyström methods work well for smooth obstacles, but are difficult to apply to surfaces with corners or edges. Therefore, we propose to use a Galerkin boundary element method (BEM) combined with *a posteriori* error estimation and adaptive mesh-refinement. The proposed algorithm is of the common type

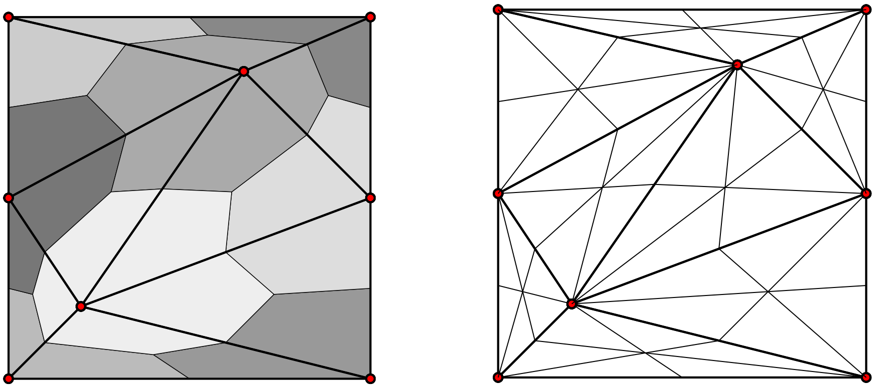

Fig. 1. Construction of the dual mesh $\mathcal{T}_\ell^{\mathrm{dual}}$ (left) and the barycentric refinement $\mathcal{T}_\ell^{\mathrm{bary}}$ (right) for a given mesh $\mathcal{T}_\ell$ indicated by the thick edges (left and right). Each (polygonal) dual element $T^{\mathrm{dual}} \in \mathcal{T}_\ell^{\mathrm{dual}}$ (left, gray) corresponds to one node of $\mathcal{T}_\ell$. Each (triangular) element $T^{\mathrm{bary}} \in \mathcal{T}_\ell^{\mathrm{bary}}$ belongs to precisely one triangle $T \in \mathcal{T}_\ell$.

$$\boxed{\texttt{solve}} \longrightarrow \boxed{\texttt{estimate}} \longrightarrow \boxed{\texttt{mark}} \longrightarrow \boxed{\texttt{refine}}$$

and generates a sequence of successively refined triangulations $\mathcal{T}_\ell$ and corresponding Galerkin approximations $\Phi_\ell \approx \phi$ such that $\mathrm{Cap}_\ell(\Omega) := \frac{1}{4\pi}\int_\Gamma \Phi_\ell \, d\Gamma$ converges to $\mathrm{Cap}(\Omega)$ at the optimal algebraic rate. The mathematical foundation of such algorithms for BEM has recently been derived in [11,12,16,18]. One novelty in the present paper is that we combine adaptivity with an effective preconditioning strategy.

We assume that $\mathcal{T}_\ell$ is a triangulation of Γ into plane surface triangles. Let $\mathcal{T}_\ell^{\mathrm{dual}}$ be the induced dual grid; see Fig. 1. Let $\mathcal{S}^1(\mathcal{T}_\ell)$ be the $\mathcal{T}_\ell$-piecewise affine globally continuous functions and $\mathcal{P}^0(\mathcal{T}_\ell^{\mathrm{dual}})$ be the $\mathcal{T}_\ell^{\mathrm{dual}}$-piecewise constant functions on Γ. It is known [25,41] that one can use the (regularized) discrete hypersingular integral operator D_ℓ^{reg} on $\mathcal{S}^1(\mathcal{T}_\ell)$ to effectively precondition the discrete weakly-singular integral operator V_ℓ^{dual} on $\mathcal{P}^0(\mathcal{T}_\ell^{\mathrm{dual}})$. The Galerkin approximation $\Phi_\ell^{\mathrm{dual}} \in \mathcal{P}^0(\mathcal{T}_\ell^{\mathrm{dual}})$ is obtained as

$$\Phi_\ell^{\mathrm{dual}} = D_\ell^{\mathrm{reg}} \widetilde{\Phi}_\ell, \quad \text{where } \widetilde{\Phi}_\ell \text{ solves the preconditioned system} \quad (V_\ell D_\ell^{\mathrm{reg}} \widetilde{\Phi}_\ell) = 1. \tag{6}$$

Having computed $\Phi_\ell^{\mathrm{dual}}$, we use that each vertex z of $\mathcal{T}_\ell$ corresponds to precisely one dual cell $T_z^{\mathrm{dual}} \in \mathcal{T}_\ell^{\mathrm{dual}}$. To steer the mesh-refinement, we compute the ZZ-type error indicators

$$\eta_\ell(T) := \mathrm{diam}(T)^{1/2} \, \|\Phi_\ell^{\mathrm{dual}} - I_\ell \Phi_\ell^{\mathrm{dual}}\|_{L^2(T)} \quad \text{for all } T \in \mathcal{T}_\ell, \tag{7}$$

where $I_\ell \Phi_\ell^{\mathrm{dual}} \in \mathcal{S}^1(\mathcal{T}_\ell)$ is the unique function with $I_\ell \Phi_\ell^{\mathrm{dual}}(z) = \Phi_\ell^{\mathrm{dual}}|_{T_z^{\mathrm{dual}}}$ for all vertices z of $\mathcal{T}_\ell$. We refer to [3,36,43] for ZZ-type estimators for FEM and to [10,13] for BEM. The local contributions $\eta_\ell(T)$ are then used to mark elements for refinement. An improved triangulation $\mathcal{T}_{\ell+1}$ is obtained from $\mathcal{T}_\ell$ by refining (essentially) these marked elements.

The mentioned results on adaptive BEM [11,12,16,18] consider residual error estimators, which provide more mathematical structure than the heuristical ZZ-error estimator (7). However, we note that the evaluation and integration of the BEM residual is usually more costly than the computation of the BEM solution (see Remark 3 below), while the ZZ-error estimator comes essentially at no cost.

The striking advantage of the proposed strategy is that both discrete integral operators D_ℓ^{reg} as well as V_ℓ^{dual} can effectively be treated as follows: Let $\mathcal{T}_\ell^{\mathrm{bary}}$ be the barycentric refinement of $\mathcal{T}_\ell$. We then build the discrete weakly-singular integral operator V_ℓ^{bary} with respect to $\mathcal{P}^0(\mathcal{T}_\ell^{\mathrm{bary}})$. Both operators D_ℓ^{reg} and V_ℓ^{dual} can be obtained by (sparse) projection operators applied to V_ℓ^{bary}; see Section 2.6. Since the computation (6) of $\Phi_\ell^{\mathrm{dual}}$ by iterative solvers relies only on matrix-vector products, we altogether assemble and store only V_ℓ^{bary}. Finally, since the number of elements satisfies $\#\mathcal{T}_\ell^{\mathrm{bary}} = 6\,\#\mathcal{T}_\ell$, preconditioning and adaptivity do not lead to any significant overhead of the overall method.

2. Preconditioned adaptive BEM for Laplace problems

In this section, we describe the numerical implementation of the preconditioned boundary integral equation (6) and the corresponding ZZ-type error estimator. This forms the basis for our adaptive capacity algorithm described in Section 2.7.

The standard way of approximating Neumann data on polyhedral boundaries is the approximation through piecewise constant basis functions on the primal grid, given the discontinuity of Neumann data cross edges. Here, we choose a different approximation space, namely the space of piecewise constant functions on the dual grid. The advantage is that this space admits a stable duality pairing with continuous, piecewise affine functions on the primal grid, allowing us to use

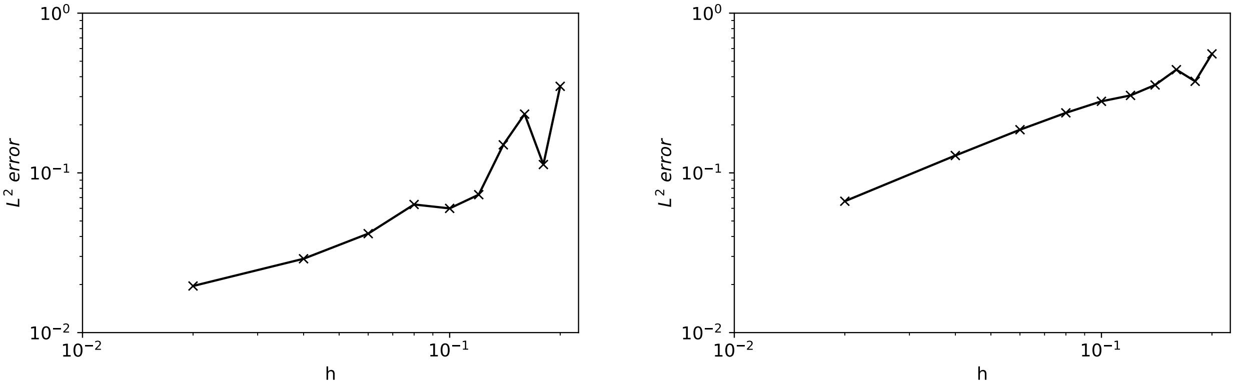

Fig. 2. Approximation error of the BEM for a Laplace–Dirichlet problem on the unit cube with continuous, piecewise linear basis functions on the primal mesh (left) and piecewise constant basis functions on the dual grid (right).

efficient operator preconditioning techniques based on the duality of the single-layer and hypersingular boundary operator, while at the same time making it possible to apply a ZZ-type error estimator as shown below.

Moreover, even though the piecewise constant basis on the dual grid is locally continuous across edges, it recovers optimal h-uniform convergence to solutions of Laplace–Dirichlet problems. An example is given in Fig. 2, which demonstrates the h-convergence in the dual basis to the Neumann data of a Laplace–Dirichlet problem on the unit cube with analytically known solution. The shown error is the relative L^2-error for the piecewise smooth Neumann data. The convergence is linear.

While a complete convergence analysis in the dual basis is beyond the merit of this paper, a simple heuristic argument is the following: The dual basis is continuous in a radius of diameter h across an edge, and discontinuous globally. Hence, locally around an edge we have an error contribution that decreases with order h, while not propagating beyond due to the discontinuity of the basis. We therefore expect to recover the optimal convergence order h for the L^2-error.

2.1. Triangulations

Throughout, let $\Omega \subset \mathbb{R}^3$ be a bounded polyhedral domain with closed boundary Γ. We consider conforming triangulations $\mathcal{T}_\ell = \{T_{\ell,1}, \dots, T_{\ell,N_\ell}\}$ of Γ into $N_\ell = \#\mathcal{T}_\ell$ plane (closed) surface triangles $T_{\ell,j} \in \mathcal{T}_\ell$. Let $\mathcal{N}_\ell = \{z_{\ell,1}, \dots, z_{\ell,M_\ell}\}$ be the set of vertices of $\mathcal{T}_\ell$, which contains the $M_\ell = \#\mathcal{N}_\ell$ vertices.

To obtain the dual mesh $\mathcal{T}_\ell^{\mathrm{dual}}$, we connect the center of gravity of each element $T_{\ell,j} \in \mathcal{T}_\ell$ with the midpoints of its edges. These lines define M_ℓ non-degenerate closed polygons $T_{\ell,j}^{\mathrm{dual}}$, which are collected in the *dual mesh* $\mathcal{T}_\ell^{\mathrm{dual}} = \{T_{\ell,1}^{\mathrm{dual}}, \dots, T_{\ell,M_\ell}^{\mathrm{dual}}\}$, where $\#\mathcal{T}_\ell^{\mathrm{dual}} = M_\ell = \#\mathcal{N}_\ell$. Note that each cell $T_{\ell,j}^{\mathrm{dual}} \in \mathcal{T}_\ell^{\mathrm{dual}}$ contains precisely one node $z_{\ell,j} \in \mathcal{N}_\ell$ in its interior, and we use the according numbering of $\mathcal{T}_\ell^{\mathrm{dual}}$ and $\mathcal{N}_\ell$, i.e., $z_{\ell,j} \in \mathrm{interior}(T_{\ell,j}^{\mathrm{dual}})$ for all $j = 1, \dots, M_\ell$; see Fig. 1 (left).

Finally, we refine each element $T_{\ell,j} \in \mathcal{T}_\ell$ into six triangles by connecting the center of gravity of $T_{\ell,j}$ with the midpoints of its edges as well as its vertices. This gives rise to the *barycentric refinement* $\mathcal{T}_\ell^{\mathrm{bary}} = \{T_{\ell,1}^{\mathrm{bary}}, \dots T_{\ell,6N_\ell}^{\mathrm{bary}}\}$. We note that $\#\mathcal{T}_\ell^{\mathrm{bary}} = 6\,\#\mathcal{T}_\ell = 6\,N_\ell$. Moreover, each element $T_{\ell,j} \in \mathcal{T}_\ell$ and $T_{\ell,j}^{\mathrm{dual}} \in \mathcal{T}_\ell^{\mathrm{dual}}$ is the union of elements in $\mathcal{T}_\ell^{\mathrm{bary}}$, i.e.,

$$T = \bigcup \left\{ T^{\mathrm{bary}} \in \mathcal{T}_\ell^{\mathrm{bary}} \,:\, T^{\mathrm{bary}} \subset T \right\} \quad \text{for all } T \in \mathcal{T}_\ell \cup \mathcal{T}_\ell^{\mathrm{dual}};$$

see Fig. 1 (right). In other words, $\mathcal{T}_\ell^{\mathrm{bary}}$ is the coarsest common refinement of $\mathcal{T}_\ell$ and $\mathcal{T}_\ell^{\mathrm{dual}}$ into plane surface triangles.

2.2. Discrete function spaces

In this section, we collect the discrete spaces of functions $\Gamma \to \mathbb{R}$, which are employed below.

On $\mathcal{T}_\ell^{\mathrm{dual}}$, we consider the space $\mathcal{P}^0(\mathcal{T}_\ell^{\mathrm{dual}})$ of $\mathcal{T}_\ell^{\mathrm{dual}}$-piecewise constant functions. We choose the basis $\{\chi_{\ell,1}^{\mathrm{dual}}, \dots, \chi_{\ell,M_\ell}^{\mathrm{dual}}\}$ consisting of characteristic functions, i.e., $\chi_{\ell,j}^{\mathrm{dual}}(x) = 1$ on $T_{\ell,j}^{\mathrm{dual}}$ and zero otherwise.

On $\mathcal{T}_\ell$, we consider the space $\mathcal{S}^1(\mathcal{T}_\ell)$ of $\mathcal{T}_\ell$-piecewise affine functions, which are globally continuous. We choose the usual nodal basis $\{\varphi_{\ell,1}, \dots, \varphi_{\ell,M_\ell}\}$, which consists of the hat functions characterized by $\varphi_{\ell,i}(z_{\ell,i}) = 1$ and $\varphi_{\ell,i}(z_{\ell,j}) = 0$ for $i \neq j$.

On $\mathcal{T}_\ell^{\mathrm{bary}}$, we consider the space $\mathcal{P}^0(\mathcal{T}_\ell^{\mathrm{bary}})$ of $\mathcal{T}_\ell^{\mathrm{bary}}$-piecewise constant functions. Again, we choose the basis $\{\chi_{\ell,1}^{\mathrm{bary}}, \dots, \chi_{\ell,6N_\ell}^{\mathrm{bary}}\}$ consisting of characteristic functions, i.e., $\chi_{\ell,j}^{\mathrm{bary}}(x) = 1$ on $T_{\ell,j}^{\mathrm{bary}}$ and zero otherwise.

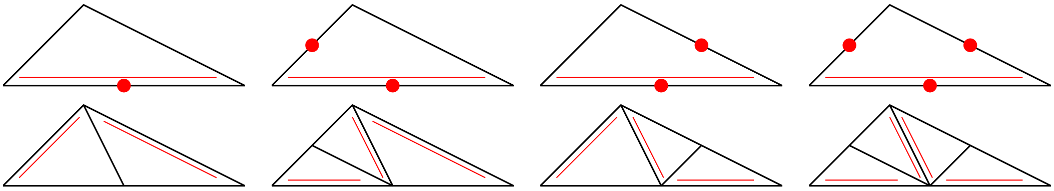

Fig. 3. For each triangle $T \in \mathcal{T}_\ell$, there is one fixed *reference edge*, indicated by the double line (left, top). Refinement of T is done by bisecting the reference edge, where its midpoint becomes a new vertex of the refined triangulation $\mathcal{T}_{\ell+1}$. The reference edges of the son triangles are opposite to this newest vertex (left, bottom). To avoid hanging nodes, one proceeds as follows: We assume that certain edges of T, but at least the reference edge, are marked for refinement (top). Using iterated newest vertex bisection, the element is then split into 2, 3, or 4 son triangles (bottom).

2.3. Galerkin discretization of the Laplace integral equation

For general $f \in H^{1/2}(\Gamma)$ (and $f = 1$ for the capacity problem), we consider the weakly-singular boundary integral equation

$$V\phi(x) := \frac{1}{4\pi} \int_\Gamma \frac{\phi(y)}{|x-y|}\, d\Gamma(y) = f(x) \quad \text{for all } x \in \Gamma, \tag{8}$$

where $V : H^{-1/2}(\Gamma) \to H^{1/2}(\Gamma)$ is the Laplace single-layer integral operator.

We refer, e.g., to the monographs [23,29,32,38,40] for the following facts on the functional analytic framework:

Let $\langle g\,,\,\psi\rangle_\Gamma := \int_\Gamma g(x)\psi(x)\,d\Gamma(x)$ be the $L^2(\Gamma)$-based duality pairing between functions $g \in H^{1/2}(\Gamma)$ and $\psi \in H^{-1/2}(\Gamma)$. We can reformulate (8) as variational problem in the form

$$\langle V\phi\,,\,\psi\rangle_\Gamma = \langle f\,,\,\psi\rangle_\Gamma \quad \text{for all } \psi \in H^{-1/2}(\Gamma). \tag{9}$$

From the ellipticity of V in $H^{-1/2}(\Gamma)$ with respect to $\langle\cdot\,,\,\cdot\rangle_\Gamma$, the unique solvability of (9) follows by the Lax–Milgram lemma. In particular, we note that $|||\,\psi\,||| := \langle V\psi\,,\,\psi\rangle_\Gamma^{1/2} \simeq \|\psi\|_{H^{-1/2}(\Gamma)}$ defines the equivalent energy norm on $H^{-1/2}(\Gamma)$. Here and throughout, $\lesssim$ abbreviates $\leq$ up to some generic multiplicative constant, and $\simeq$ abbreviates that both estimates $\lesssim$ and $\gtrsim$ hold.

Since $\mathcal{P}^0(\mathcal{T}_\ell^{\rm dual}) \subset H^{-1/2}(\Gamma)$, we can discretize (9) by a Galerkin discretization [38,40]: Find $\Phi_\ell^{\rm dual} \in \mathcal{P}^0(\mathcal{T}_\ell^{\rm dual})$ such that

$$\langle V\Phi_\ell^{\rm dual}\,,\,\Psi_\ell^{\rm dual}\rangle_\Gamma = \langle f\,,\,\Psi_\ell^{\rm dual}\rangle_\Gamma \quad \text{for all } \Psi_\ell^{\rm dual} \in \mathcal{P}^0(\mathcal{T}_\ell^{\rm dual}). \tag{10}$$

Again, unique solvability of (10) follows by the Lax–Milgram lemma. Moreover, the latter is equivalent to the linear system of equations

$$\mathbf{V}_\ell^{\rm dual}\mathbf{x}_\ell = \mathbf{f}_\ell^{\rm dual} \tag{11}$$

with the symmetric and positive definite matrix $\mathbf{V}_\ell^{\rm dual} \in \mathbb{R}_{\rm sym}^{M_\ell \times M_\ell}$ and given right-hand side $\mathbf{f}_\ell^{\rm dual} \in \mathbb{R}^{M_\ell}$ defined as $\mathbf{V}_\ell^{\rm dual}[i,j] = \langle V\chi_{\ell,j}^{\rm dual}\,,\,\chi_{\ell,i}^{\rm dual}\rangle_\Gamma$ and $\mathbf{f}_\ell^{\rm dual}[i] = \langle f\,,\,\chi_{\ell,i}^{\rm dual}\rangle_\Gamma$. Then, the unique solution $\Phi_\ell^{\rm dual} \in \mathcal{P}^0(\mathcal{T}_\ell^{\rm dual})$ of (10) satisfies $\Phi_\ell^{\rm dual} = \sum_j \mathbf{x}_\ell[j]\chi_{\ell,j}^{\rm dual}$, where $\mathbf{x}_\ell \in \mathbb{R}^{M_\ell}$ is the unique solution to the linear system (11).

2.4. ZZ-type a posteriori error estimation

Having computed the Galerkin solution $\Phi_\ell^{\rm dual} \in \mathcal{P}^0(\mathcal{T}_\ell^{\rm dual})$, we define $I_\ell\Phi_\ell^{\rm dual} := \sum_j \mathbf{x}_\ell[j]\varphi_{\ell,j} \in \mathcal{S}^1(\mathcal{T}_\ell)$ as well as the ZZ-type error indicators (7) for all $T \in \mathcal{T}_\ell$. For model problems with known solution, our numerical experiments led to the evidence that

$$\sum_{T\in\mathcal{T}_\ell} \eta_\ell(T)^2 = \|h_\ell^{1/2}(1-I_\ell)\Phi_\ell^{\rm dual}\|_{L^2(\Gamma)}^2 \simeq |||\,\phi - \Phi_\ell^{\rm dual}\,|||^2 \simeq \|\phi - \Phi_\ell^{\rm dual}\|_{H^{-1/2}(\Gamma)}^2, \tag{12}$$

where $h_\ell : \Gamma \to \mathbb{R}$ is the $\mathcal{T}_\ell$-piecewise mesh-size function $h_\ell|_T = \mathrm{diam}(T)$. A thorough mathematical proof goes beyond of the scope of the present work. We note that the efficiency estimate $\lesssim$ can be obtained with the scaling techniques from [13], while the more important reliability estimate $\gtrsim$ remains open.

2.5. A generic adaptive strategy

With the ZZ-type error estimator at hand, we consider the following generic adaptive algorithm, which locally adapts the primal mesh $\mathcal{T}_\ell$, while the Galerkin approximations $\Phi_\ell^{\rm dual}$ are computed on the corresponding dual mesh $\mathcal{T}_\ell^{\rm dual}$. For the local mesh-refinement, we employ 2D newest vertex bisection (NVB); see, e.g., [31,39] and Fig. 3 for some illustration. We note that NVB provides rich mathematical structure and, in particular, guarantees that all adaptive meshes are conforming and uniformly shape regular (i.e., generated triangles T cannot deteriorate).

Algorithm 1. Input: Initial conforming triangulation $\mathcal{T}_0$ of Γ into plane surface triangles, adaptivity parameters $0 < \theta \le 1$ and $C_{\rm mark} \ge 1$.
Adaptive loop. Iterate the following steps (i)–(iv) for all $\ell = 0, 1, 2, \dots$:

(i) Build the dual mesh $\mathcal{T}_\ell^{\rm dual}$ and compute the solution $\Phi_\ell^{\rm dual} \in \mathcal{P}^0(\mathcal{T}_\ell^{\rm dual})$ of (10).
(ii) Compute the local contributions $\eta_\ell(T)$ from (7) for all $T \in \mathcal{T}_\ell$.
(iii) Determine a set of, up to the multiplicative constant $C_{\rm mark}$, minimal cardinality, which satisfies the Dörfler marking criterion [8]

$$\theta \sum_{T \in \mathcal{T}_\ell} \eta_\ell(T)^2 \le \sum_{T \in \mathcal{M}_\ell} \eta_\ell(T)^2, \tag{13}$$

i.e., the local contributions associated with $\mathcal{M}_\ell$ control a fixed percentage of η_ℓ^2.
(vi) Use NVB to create the coarsest conforming triangulation $\mathcal{T}_{\ell+1} := \texttt{refine}(\mathcal{T}_\ell, \mathcal{M}_\ell)$, where all marked elements $T \in \mathcal{M}_\ell$ have been bisected.

Output: Sequence of successively refined triangulations $\mathcal{T}_\ell$ as well as corresponding Galerkin solutions $\Phi_\ell^{\rm dual} \in \mathcal{P}^0(\mathcal{T}_\ell^{\rm dual})$ and ZZ-type error estimators η_ℓ. □

The code of a Python implementation of Algorithm 1 based o the Bempp library is given in Appendix B. Similar algorithms driven by computationally expensive residual error estimators have been mathematically analyzed in recent years [11,12,15,16,18]. We refer to the appendix for details on available results.

Remark 2.

(i) In practice, the linear system (11) is solved iteratively. In our implementation, (the preconditioned) GMRES solver is stopped, if the iterates $\Phi_{\ell,k}^{\rm dual} \approx \Phi_\ell^{\rm dual}$ satisfy that $|||\, \Phi_{\ell,k-1}^{\rm dual} - \Phi_{\ell,k}^{\rm dual} \,||| \le \lambda \eta_\ell(\Phi_{\ell,k}^{\rm dual})$ where $\lambda \approx 10^{-3}$. Here, $\eta_\ell(\Phi_{\ell,k}^{\rm dual})$ denotes the ZZ-error estimator (7) evaluated at the inexact solution $\Phi_{\ell,k}^{\rm dual}$ instead of the exact Galerkin solution $\Phi_\ell^{\rm dual}$. We refer to the recent work [15] for a thorough mathematical analysis of the stopping criterion.
(ii) Since the BEM matrix V_ℓ^{dual} is symmetric, it would be possible to employ a preconditioned CG algorithm (PCG) for the iterative solution of (11). However, generic implementations of PCG are restricted to symmetric preconditioners. Since we employ an operator preconditioning with unsymmetric mass matrix (see Section 2.6 below) and since the BEM++ library [37] also treats non-symmetric BEM formulations, our implementation uses a generic implementation of precondioned GMRES.

Remark 3. In the numerical experiments below, we compare our adaptive algorithm (with BEM solution $\Phi_\ell^{\rm dual} \in \mathcal{P}^0(\mathcal{T}_\ell^{\rm dual})$ with a standard adaptive algorithm (with BEM solution $\Phi_\ell \in \mathcal{P}^0(\mathcal{T}_\ell)$ on the primal mesh) driven by a residual error estimator. In order to compute the weighted-residual error estimator for lowest order BEM, one can do the following: Instead of assembling up the discrete single-layer operator $\mathbf{V}_\ell^{\mathcal{P}^0}$ on $\mathcal{P}^0(\mathcal{T}_\ell)$, we assemble the operator $\mathbf{V}_\ell^{\mathcal{P}^1}$ on $\mathcal{P}^1(\mathcal{T}_\ell)$. By applying sparse projection operators onto $\mathbf{V}_\ell^{\mathcal{P}^1}$, we obtain the matrix $\mathbf{V}_\ell^{\mathcal{P}^0}$ as well as the evaluation the residual in $\mathcal{P}^1(\mathcal{T}_\ell)$ to compute the residual error estimator. Since the change of discrete spaces from $\mathcal{P}^1(\mathcal{T}_\ell)$ to $\mathcal{S}^1(\mathcal{T}_\ell)$ increases the degrees of freedom by a factor 3, the computational costs grow at least by the same factor.

On the other hand, the proposed Algorithm 1 with the ZZ-type error estimator requires to assemble the discrete single-layer operator $\mathbf{V}_\ell^{\rm bary}$ on $\mathcal{P}^0(\mathcal{T}_\ell^{\rm bary})$, which increases the degrees of freedom by a factor 6, but already includes the operator preconditioning; see Section 2.6 for further details. Hence, using a suitable implementation with FMM or $\mathcal{H}$-Matrices, Algorithm 1 with an built-in operator preconditioning just leads to double assembling costs compared to a non-preconditioned adaptive scheme based on the residual error estimator.

2.6. Operator preconditioning with the hypersingular operator

While the integral equation (11) can be solved via dense LU decomposition for smaller system sizes, larger problems require iterative methods, in particular, if BEM acceleration techniques such as $\mathcal{H}$-matrices [4,5,24] or FMM [20,22,33] are used. An efficient preconditioning strategy is based on operator preconditioning with the hypersingular operator. We recall from the literature [29,32] that the hypersingular operator $D : H^{1/2}(\Gamma) \to H^{-1/2}(\Gamma)$ is the negative normal derivative of the double-layer boundary operator, i.e.,

$$[Dv](x) := -\frac{\partial}{\partial \nu(x)} \int\limits_{\Gamma} \frac{\langle \nu(y), x-y \rangle}{4\pi\,|x-y|^3}\, v(y)\, d\Gamma(y).$$

For $v, w \in H^{1/2}(\Gamma)$, it follows from integration by parts that [23,38,40]

$$\langle Dv, w \rangle = \frac{1}{4\pi} \int\limits_{\Gamma}\int\limits_{\Gamma} \frac{\langle \underline{\mathrm{curl}}_{\Gamma} v(y), \underline{\mathrm{curl}}_{\Gamma} w(x) \rangle}{|x-y|}\, d\Gamma(y).$$

In case of the Laplace equation and Γ being connected, the hypersingular operator has a one-dimensional nullspace consisting of the constant functions on Γ. In order to obtain a suitable preconditioner, we therefore introduce the regularized operator

$$\left[D^{\mathrm{reg}} v\right](x) := [Dv](x) + \int\limits_{\Gamma} v(x)\, d\Gamma(x).$$

This operator shifts the zero eigenvalue for constant functions to the value $|\Gamma|$ and otherwise does not change the spectrum. In [41], it was shown that the regularized hypersingular operator is an effective preconditioner for the single-layer boundary operator. We use the regularized hypersingular operator as a right preconditioner in (11) and in addition, an inverse mass matrix as a left preconditioner to arrive at

$$\mathbf{M}_\ell^{-\top}\mathbf{V}_\ell^{\mathrm{dual}}\mathbf{M}_\ell^{-1}\mathbf{D}_\ell^{\mathrm{reg}}\tilde{\mathbf{x}}_\ell = \mathbf{M}_\ell^{-\top}\mathbf{f}_\ell^{\mathrm{dual}} \tag{14}$$

with $\mathbf{x}_\ell = \mathbf{M}_\ell^{-1}\mathbf{D}_\ell^{\mathrm{reg}}\tilde{\mathbf{x}}_\ell$, $\mathbf{D}_\ell^{\mathrm{reg}}[i,j] = \langle D_\ell^{\mathrm{reg}}\phi_{\ell,j}\,,\,\phi_{\ell,i}\rangle$, $\mathbf{M}_\ell[i,j]^\top = \langle \phi_{\ell,j}\,,\,\chi_{\ell,i}^{\mathrm{dual}}\rangle$, and the other quantities defined as before. The implementation of (14) only requires the assembly of the matrix $\mathbf{V}_\ell^{\mathrm{bary}} \in \mathbb{R}^{6N_\ell \times 6N_\ell}$ associated with the single-layer boundary operator on the space $\mathcal{P}^0(\mathcal{T}_\ell^{\mathrm{bary}})$, i.e., $\mathbf{V}_\ell^{\mathrm{bary}}[i,j] = \langle V\chi_{\ell,j}^{\mathrm{bary}}\,,\,\chi_{\ell,i}^{\mathrm{bary}}\rangle_\Gamma$. The matrices $\mathbf{V}_\ell^{\mathrm{dual}}$ and $\mathbf{D}_\ell^{\mathrm{reg}}$ can then be obtained through simple sparse projection operators. To obtain $\mathbf{V}_\ell^{\mathrm{dual}}$, we notice that the basis functions $\chi_{\ell,j}^{\mathrm{dual}}$ in $\mathcal{P}^0(\mathcal{T}_\ell^{\mathrm{dual}})$ can be written as a linear combinations of the basis $\chi_{\ell,j}^{\mathrm{bary}}$ in $\mathcal{P}^0(\mathcal{T}_\ell^{\mathrm{bary}})$. Let $\mathbf{P}_\ell \in \mathbb{R}^{6N_\ell \times M_\ell}$ be the sparse matrix that maps coefficients of basis functions in $\mathcal{P}^0(\mathcal{T}_\ell^{\mathrm{dual}})$ to a vector of coefficients of the associated basis functions in $\mathcal{P}^0(\mathcal{T}_\ell^{\mathrm{bary}})$. Then,

$$\mathbf{V}_\ell^{\mathrm{dual}} = \mathbf{P}_\ell^\top \mathbf{V}_\ell^{\mathrm{bary}} \mathbf{P}_\ell.$$

Similarly, we recognize that we can write

$$\mathbf{D}_\ell^{\mathrm{reg}} = \sum_{m=1}^{3} \mathbf{Q}_{\ell,m}^\top \mathbf{V}_\ell^{\mathrm{bary}} \mathbf{Q}_{\ell,m},$$

where $\mathbf{Q}_{\ell,m} \in \mathbb{R}^{6N_\ell \times M_\ell}$ maps coefficient vectors in $\mathcal{S}^1(\mathcal{T}_\ell)$ to the piecewise constant m-th component of the corresponding surface curl operator, which can be represented as a linear combination of basis functions in $\mathcal{P}^0(\mathcal{T}_\ell^{\mathrm{bary}})$.

Putting everything into (14), we obtain the discrete system

$$\mathbf{M}_\ell^{-\top}\mathbf{P}_\ell^\top\mathbf{V}_\ell^{\mathrm{bary}}\mathbf{P}_\ell\mathbf{M}_\ell^{-1}\sum_{m=1}^{3}\mathbf{Q}_{\ell,m}^\top\mathbf{V}_\ell^{\mathrm{bary}}\mathbf{Q}_{\ell,m}\tilde{\mathbf{x}}_\ell = \mathbf{M}_\ell^{-\top}\mathbf{f}_\ell^{\mathrm{dual}}.$$

Hence, we only require the assembly of the operator $\mathbf{V}_\ell^{\mathrm{bary}}$ and four matrix-vector products with this operator for each evaluation of the left-hand side. Assuming a fast method with a linear or log-linear complexity for assembly and matrix-vector product, the total cost of the assembly is therefore six times as high as that of the non-preconditioned system and each matrix-vector product about 24 times as expensive as that for a non-preconditioned system. However, we will see later that this preconditioner is very effective and only a small number of matrix-vector products will be required in our numerical experiments. Moreover, the error estimation essentially comes for free as part of this preconditioning strategy; see Section 2.4.

The associated sparse matrix operations with $\mathbf{P}_\ell$ and $\mathbf{Q}_\ell$ are cheap and negligible compared to the evaluation of the integral operators. Using a modern sparse direct solver, also the cost of the LU decomposition of the sparse mass matrix $\mathbf{M}_\ell$ and its application is reasonably cheap.

We note that the analysis of [41] is restricted to quasi-uniform meshes. However, we refer to [26] for operator preconditioning for 2D BEM on graded meshes. The extension of the latter result to 3D is beyond the scope of the present paper.

2.7. Numerical computation of electrostatic capacity

As outlined in the introduction, the capacity of Ω can be computed by

$$\operatorname{Cap}(\Omega) = \frac{1}{4\pi}\,\langle \phi\,,\,1\rangle_\Gamma,\ \text{where}\ \phi \in H^{-1/2}(\Gamma)\ \text{solves}\ V\phi = 1\ \text{on}\ \Gamma = \partial\Omega. \tag{15}$$

This yields $\operatorname{Cap}(\Omega) = \frac{1}{4\pi}\,\langle V^{-1}1\,,\,1\rangle_\Gamma$, and ellipticity of V^{-1} guarantees $\operatorname{Cap}(\Omega) > 0$. Let $\Phi_\ell^{\mathrm{dual}} \in \mathcal{P}^0(\mathcal{T}_\ell^{\mathrm{dual}})$ be the Galerkin approximation (10) of $\phi = V^{-1}1$, which is obtained by Algorithm 1 for $f = 1$. Then, a natural approximation of the capacity (15) is

$$\operatorname{Cap}_\ell(\Omega) := \frac{1}{4\pi}\,\langle \Phi_\ell^{\mathrm{dual}}\,,\,1\rangle_\Gamma. \tag{16}$$

We note that this approximation is controlled by the energy error.

Proposition 4. *There holds* $0 < \operatorname{Cap}_\ell(\Omega) \le \operatorname{Cap}(\Omega)$ *as well as*

$$4\pi\,(\operatorname{Cap}(\Omega) - \operatorname{Cap}_\ell(\Omega)) = |\!|\!|\,\phi - \Phi_\ell^{\mathrm{dual}}\,|\!|\!|^2 \simeq \|\phi - \Phi_\ell^{\mathrm{dual}}\|^2_{H^{-1/2}(\Gamma)}. \tag{17}$$

Proof. Define $\mathbf{1}_\ell^{\mathrm{dual}} \in \mathbb{R}^{M_\ell}$ by $\mathbf{1}_\ell^{\mathrm{dual}}[j] = \langle 1\,,\,\chi_{\ell,j}^{\mathrm{dual}}\rangle_\Gamma = |T_{\ell,j}^{\mathrm{dual}}| > 0$. Then, it holds that

$$\Phi_\ell^{\mathrm{dual}} = \sum_{j=1}^{M_\ell} \mathbf{x}_\ell[j]\chi_{\ell,j}^{\mathrm{dual}}, \quad \text{where} \quad \mathbf{x}_\ell := (\mathbf{V}_\ell^{\mathrm{dual}})^{-1}\mathbf{1}_\ell^{\mathrm{dual}}.$$

In particular, $4\pi\operatorname{Cap}_\ell(\Omega) = \mathbf{1}_\ell^{\mathrm{dual}} \cdot \mathbf{x}_\ell = \mathbf{1}_\ell^{\mathrm{dual}} \cdot (\mathbf{V}_\ell^{\mathrm{dual}})^{-1}\mathbf{1}_\ell^{\mathrm{dual}}$. Since the matrix $\mathbf{V}_\ell^{\mathrm{dual}}$ is symmetric and positive definite (and thus also its inverse $(\mathbf{V}_\ell^{\mathrm{dual}})^{-1}$), it follows that $\operatorname{Cap}_\ell(\Omega) > 0$. To see (17), recall that the combination of (9)–(10) yields the Galerkin orthogonality $\langle V(\phi - \Phi_\ell^{\mathrm{dual}})\,,\,\Psi_\ell^{\mathrm{dual}}\rangle_\Gamma = 0$ for all $\Psi_\ell^{\mathrm{dual}} \in \mathcal{P}^0(\mathcal{T}_\ell^{\mathrm{dual}})$. This leads to

$$\begin{aligned}
&4\pi\,\big(\operatorname{Cap}(\Omega) - \operatorname{Cap}_\ell(\Omega)\big) = \langle \phi - \Phi_\ell^{\mathrm{dual}}\,,\,1\rangle_\Gamma = \langle \phi - \Phi_\ell^{\mathrm{dual}}\,,\,V\phi\rangle_\Gamma \\
&\quad = \langle V(\phi - \Phi_\ell^{\mathrm{dual}})\,,\,\phi - \Phi_\ell^{\mathrm{dual}}\rangle_\Gamma = |\!|\!|\,\phi - \Phi_\ell^{\mathrm{dual}}\,|\!|\!|^2 \ge 0.
\end{aligned}$$

This proves (17) and, in particular, $\operatorname{Cap}_\ell(\Omega) \le \operatorname{Cap}(\Omega)$. □

3. Numerical computations

In this section, we present some numerical computations that underpin our theoretical findings. In the experiments, we compare the performance of

- Algorithm 1 (with Galerkin solution $\Phi_\ell^{\mathrm{dual}} \in \mathcal{P}^0(\mathcal{T}_\ell^{\mathrm{dual}})$ on the dual mesh and driven by the proposed ZZ-type estimator η_ℓ),
- a standard adaptive algorithm (see Appendix A) from [11,16] with Galerkin approximation $\Phi_\ell \in \mathcal{P}^0(\mathcal{T}_\ell)$ on the primal mesh $\mathcal{T}_\ell$ and driven by the weighted-residual error estimator,

for different adaptivity parameters $0 < \theta \le 1$. One particular focus is on preconditioners, where we compare

- Algorithm 1 with its built-in operator preconditioning vs. diagonal preconditioning vs. non-preconditioning;
- Algorithm 5 with multilevel additive Schwarz preconditioning (MLAS) [15] vs. diagonal preconditioning [21] vs. non-preconditioning;

We consider lowest-order BEM for different polyhedral domains Ω starting from simple shapes, e.g., the unit cube (Section 3.1), and moving to more complicated shapes, e.g., a star-shaped domain (Section 3.3). If not stated otherwise, we employ Algorithm 1 with the Dörfler parameter $\theta = 1/2$ and operator preconditioning.

We emphasize that due generic edge singularities, increasing the polynomial degree, without using anisotropic refinement, does not lead to any improved order of convergence. Usually one observes a slightly better constant, but the same overall algebraic rate; see, e.g., [14].

The numerical computations were done with help of Bempp, which is an open-source Galerkin boundary element library. We refer to [19,37,42] for details on Bempp. A possible implementation of Algorithm 1 in Python with Bempp is given in Appendix B.

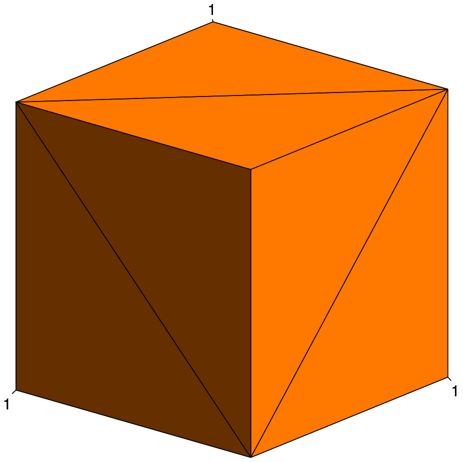

Fig. 4. Example 3.1: Initial mesh $\mathcal{T}_0$ with 12 elements.

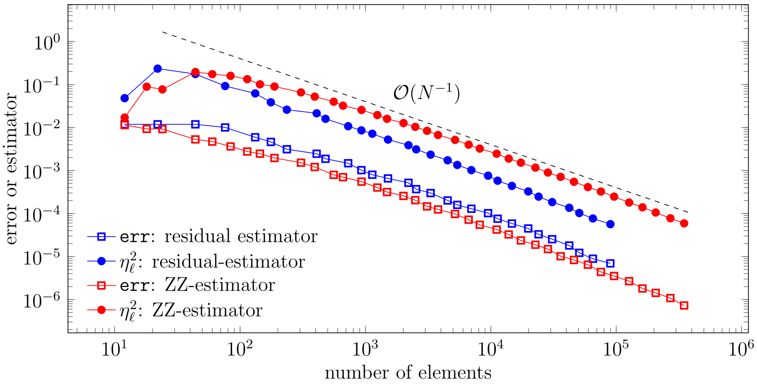

Fig. 5. Example 3.1: Convergence of the error estimator η_ℓ^2 and capacity error $\texttt{err}_\ell$. The estimator η_ℓ^2 as well as the error converge with optimal order $\mathcal{O}(N^{-1})$, if the a weighted-residual or the proposed ZZ-based error estimator is used.

3.1. Example (Unit cube)

We consider the unit cube $\Omega = (0,1)^3$, where a reference value of the capacity $\mathrm{Cap}(\Omega)$ is known; see Fig. 4. To our knowledge, the approximation $\mathrm{Cap}(\Omega) \approx 0.66067815409957$ from [28, Table 1] with an error of order 10^{-13} is the best approximation available. This value is used to compute the capacity error $\texttt{err}_\ell := |\mathrm{Cap} - \mathrm{Cap}_\ell|$.

Fig. 5 shows the convergence rates of the estimator η_ℓ^2 and the capacity error $\texttt{err}_\ell$ plotted over the number of elements $\#\mathcal{T}_\ell$. For both adaptive algorithms, the error as well as the estimator converge with rate $\mathcal{O}(N^{-1})$. This underlines empirically that both error estimators are reliable and efficient, i.e., up to multiplicative constants equivalent to the error (i.e., $\eta_\ell^2 \simeq \texttt{err} = |\!|\!|\, \phi - \Phi_\ell \,|\!|\!|^2$). Although the empirical reliability constant $\texttt{err}_\ell/\eta_\ell^2$ is slightly better for Algorithm 5 and the residual estimator, Algorithm 1 and ZZ-type estimator lead to a smaller error for the capacity approximation. Fig. 6 compares different values of the Dörfler marking parameter $\theta \in \{0.2, 0.4, 0.6, 0.8, 1\}$ in Algorithm 1, where $\theta = 1$ coincides with uniform mesh-refinement. We see that Algorithm 1 is robust and any choice of $\theta < 1$ leads to the optimal convergence behaviour. Only uniform mesh-refinement ($\theta = 1$) leads to a sub-optimal rate $\mathcal{O}(N^{-2/3})$. Table 1 displays some numerical results for the sequence of solutions produced by Algorithm 1. It shows that, e.g., Step 13 with 930 elements and 12 iterations is sufficient to reach an accuracy of $5.520 \cdot 10^{-4}$. Such a computation can be done on a usual workstation within seconds.

Fig. 7 displays the condition number of the arising linear systems (10). We emphasize that the MLAS preconditioning is optimal in the sense that the condition number of the arising systems remains bounded, independently of the number of elements; see [15]. This is confirmed for MLAS as well as empirically observed also for operator preconditioning with the hypersingular operator in Algorithm 1. On the other hand, the condition number of the non-preconditioned systems explodes with increasing number of elements resp. mesh graduation. Moreover, Fig. 7 shows that diagonal preconditioning does lead to a slower growth of the condition number, but is not optimal. Fig. 8 and Fig. 9 show the distribution of the error estimator in some of the generated adaptive meshes. Due to generic edge singularities, the adaptive al-

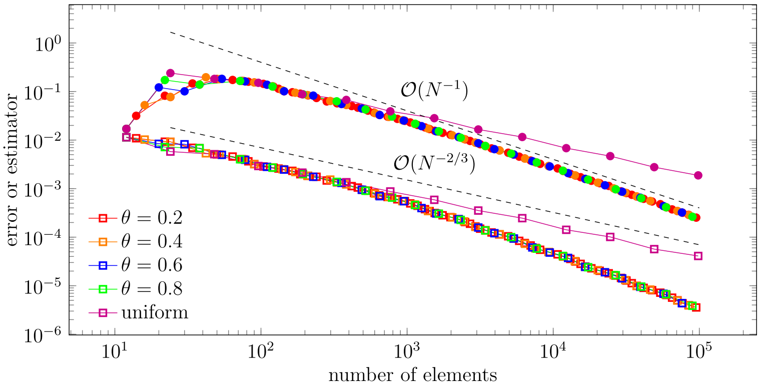

Fig. 6. Example 3.1: Convergence of the error estimator η_ℓ^2 (circles) and capacity error $\mathtt{err}_\ell$ (squares) for different value of $0 < \theta < 1$ and uniform refinement $\theta = 1$. Uniform refinement leads to a reduced rate of convergence.

Table 1
Example 3.1: Numerical results of Algorithm 1 for $\theta = 1/2$.

ℓ	$\#\mathcal{T}_\ell$	Cap_ℓ	η_ℓ^2	$\mathtt{err}_\ell = \lvert\mathrm{Cap} - \mathrm{Cap}_\ell\rvert$	$\mathtt{err}_\ell/\eta_\ell^2$	# iterations
0	12	0.6492810516	$1.702 \cdot 10^{-2}$	$1.114 \cdot 10^{-2}$	0.654	4
6	114	0.6579150471	$1.133 \cdot 10^{-1}$	$2.763 \cdot 10^{-3}$	0.024	12
13	930	0.6601261997	$2.560 \cdot 10^{-2}$	$5.520 \cdot 10^{-4}$	0.022	12
25	11132	0.6606358189	$2.460 \cdot 10^{-3}$	$4.234 \cdot 10^{-5}$	0.017	12
34	96216	0.6606746264	$2.473 \cdot 10^{-4}$	$3.528 \cdot 10^{-6}$	0.014	12
39	348152	0.6606774253	$5.920 \cdot 10^{-5}$	$7.288 \cdot 10^{-7}$	0.012	

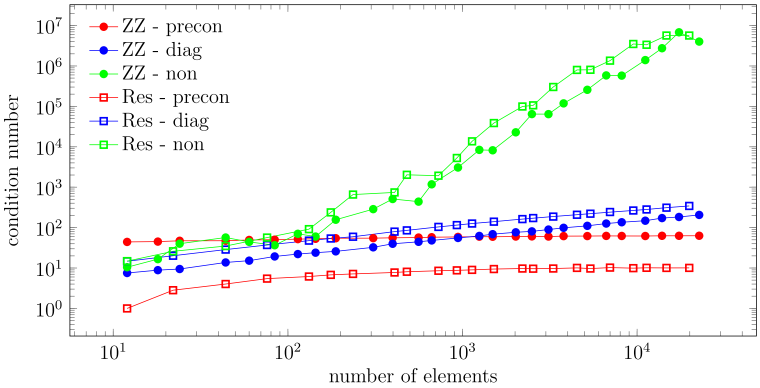

Fig. 7. Example 3.1: Condition number for different preconditioning strategies for the arising linear systems in Algorithm 1 (ZZ) with operator preconditioning with the hypersingular operators vs. Algorithm (Res) with multilevel additive Schwarz preconditioning. Additionally, diagonal preconditioning is employed in both cases.

gorithm leads to refinement along edges and corners, while elements in the interior of the cube-facets stay relatively coarse.

3.2. Example 2: Fichera cube

As second example, we consider the Fichera cube $[-1, 1]^3 \setminus [0, 1]^3$ with side length 2; see Fig. 11 (left). In contrast to the unit cube, we lack a highly accurate benchmark computation and approximate the capacity error by

$$\mathtt{err}_\ell := \lvert\mathrm{Cap} - \mathrm{Cap}_\ell\rvert \approx \lvert\mathrm{Cap}_{\ell_{\max}} - \mathrm{Cap}_\ell\rvert \quad \text{for all } 0 \le \ell < \ell_{\max}.$$

Convergence behaviour of the estimator η_ℓ^2 and of the error $\mathtt{err}_\ell$ are displayed in Fig. 10. Both, Algorithm 1 as well as Algorithm 5, lead to the optimal rate of convergence $\mathcal{O}(N^{-1})$. Table 2 shows some numerical results produced by Algorithm 1. For the fichera cube approximately 10^4 elements are sufficient to achieve an error of order 10^{-4}. This can be computed easily on a workstation within a couple of minutes. Fig. 11 and Fig. 12 show the distribution of the error estimator in some of the generated adaptive meshes. Due to generic edge singularities, the adaptive algo-

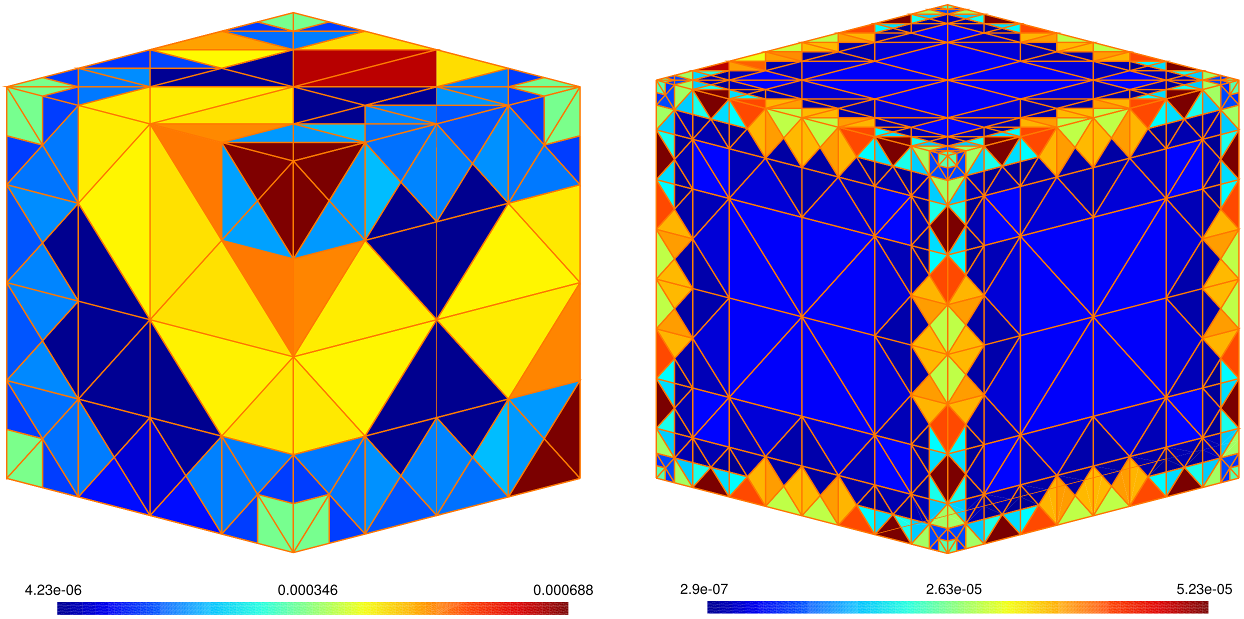

Fig. 8. Example 3.1: Mesh $\mathcal{T}_\ell$ and distribution of the error estimator η_ℓ^2 for $\ell = 10$ with 306 elements (left) and for $\ell = 15$ with 1244 elements (right).

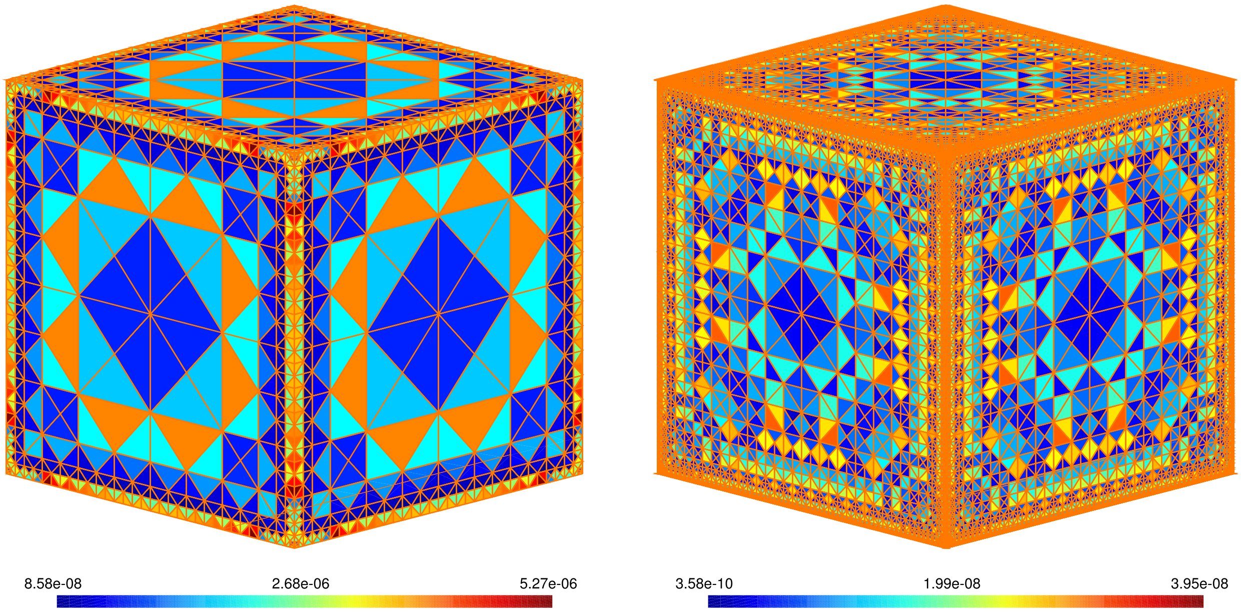

Fig. 9. Example 3.1: Mesh $\mathcal{T}_\ell$ and distribution of the error estimator η_ℓ^2 for $\ell = 20$ with 3792 elements (left) and for $\ell = 30$ with 46044 elements (right).

rithm leads to refinement along edges and corners, while elements in the interior of the cube faces stay relatively coarse.

3.3. *Example 3: Star-shaped geometry*

As third example, we consider a star-shaped geometry shown in Fig. 13. We use Algorithm 1 as black-box method. Fig. 14 shows the convergence of the estimator η_ℓ^2 and the capacity error $\texttt{err}_\ell := |\mathrm{Cap}_{\ell_{\max}} - \mathrm{Cap}_\ell| \approx |\mathrm{Cap} - \mathrm{Cap}_\ell|$. We observe the optimal convergence order $\mathcal{O}(N^{-1})$ for the estimator as well as the error for Algorithm 1 and Algorithm 5. The rate of convergence is robust with respect to the Dörfler marking parameter $0 < \theta \leq 1$. Besides uniform refinement ($\theta = 1$), all tested parameters $\theta \in \{0.2, 0.4, 0.6, 0.8\}$ lead to the same rate of $\mathcal{O}(N^{-1})$; see Fig. 15.

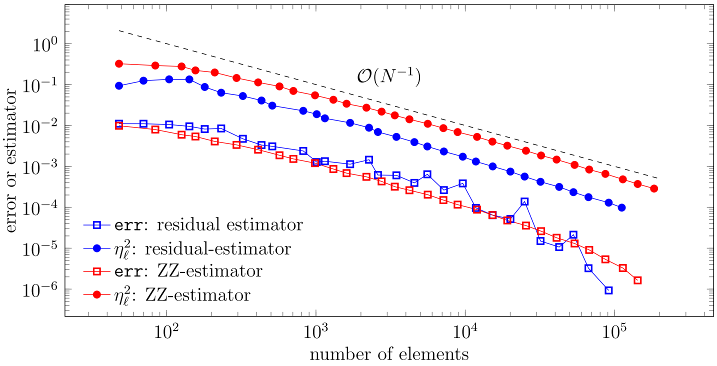

Fig. 10. Example 3.2: Convergence of the error estimator η_ℓ^2 and capacity error $\texttt{err}_\ell$. The estimator η_ℓ^2 as well as the error converge with optimal order $\mathcal{O}(N^{-1})$, if the weighted-residual or the proposed ZZ-based error estimator is used.

Table 2
Example 3.2: Numerical results of Algorithm 1 for $\theta = 1/2$.

step ℓ	number of elements $\#\mathcal{T}_\ell$	Cap_ℓ	η_ℓ^2	$\lvert\text{Cap}_{\ell_{\max}} - \text{Cap}_\ell\rvert$
0	48	1.2813985240	$3.239 \cdot 10^{-1}$	$9.858 \cdot 10^{-3}$
4	126	1.2853002501	$2.776 \cdot 10^{-1}$	$5.957 \cdot 10^{-3}$
9	986	1.2900557132	$5.470 \cdot 10^{-2}$	$1.201 \cdot 10^{-3}$
19	11992	1.2911689924	$5.270 \cdot 10^{-3}$	$8.776 \cdot 10^{-5}$
28	113484	1.2912534661	$4.829 \cdot 10^{-4}$	$3.282 \cdot 10^{-6}$
30	183762	1.2912567475	$2.868 \cdot 10^{-4}$	

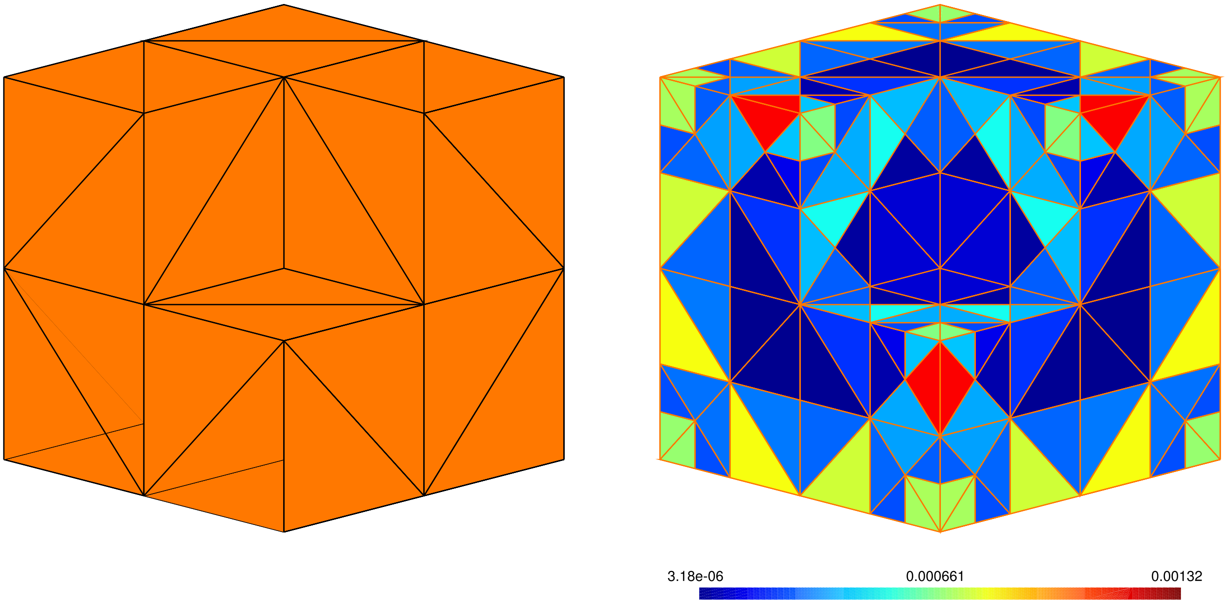

Fig. 11. Example 3.2: Initial triangulation $\mathcal{T}_0$ (left) consisting of 48 triangles. Mesh $\mathcal{T}_\ell$ and distribution of the error estimator η_ℓ^2 for $\ell = 5$ with 410 elements (right).

Fig. 16 shows the condition number of the arising linear systems (10). Operator preconditioning with the hypersingular operator in Algorithm 1 as well as the multilevel additive Schwarz preconditioner guarantee a bounded condition number. On the other hand, diagonal- and non-preconditioning lead to an increasing condition number as the number of elements grows. Similarly to previous examples, the Algorithm has to resolve generic edge singularities. Fig. 17 and Fig. 18 show the distribution of η_ℓ^2 on some meshes generated by Algorithm 1.

Table 3 shows some numerical results for the star-example. Algorithm 1 takes 22 steps and approximately 10^4 elements to reach an error of order 10^{-4}.

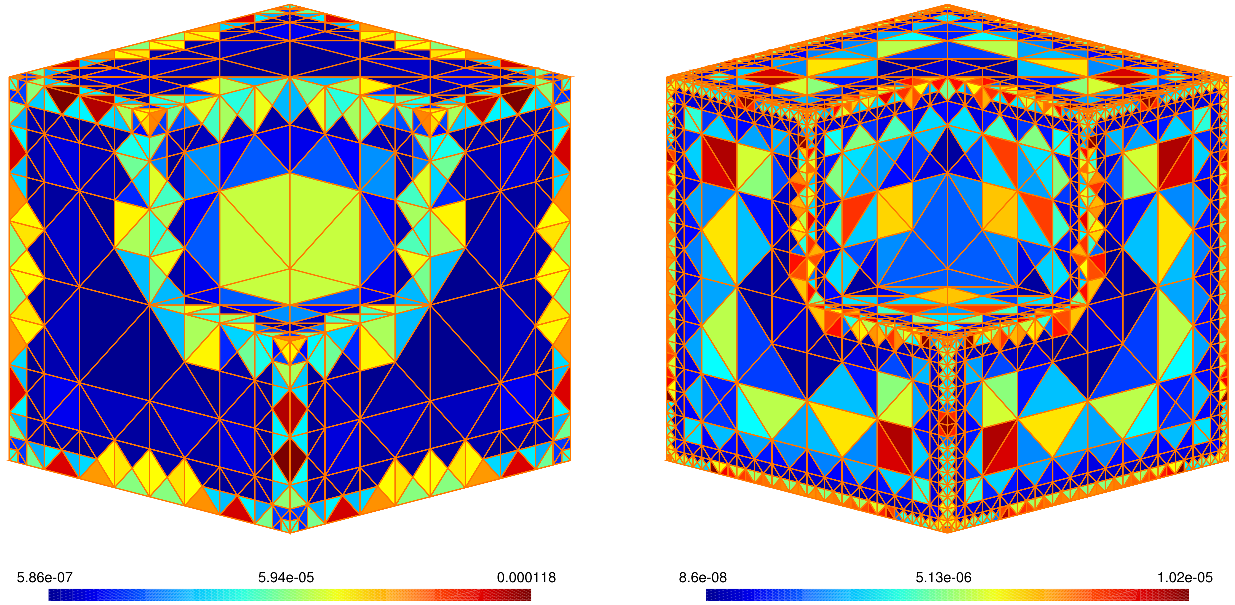

Fig. 12. Example 3.2: Mesh $\mathcal{T}_\ell$ and distribution of the error estimator η_ℓ^2 for $\ell = 10$ (left) with 1308 elements and $\ell = 15$ (right) with 4226 elements.

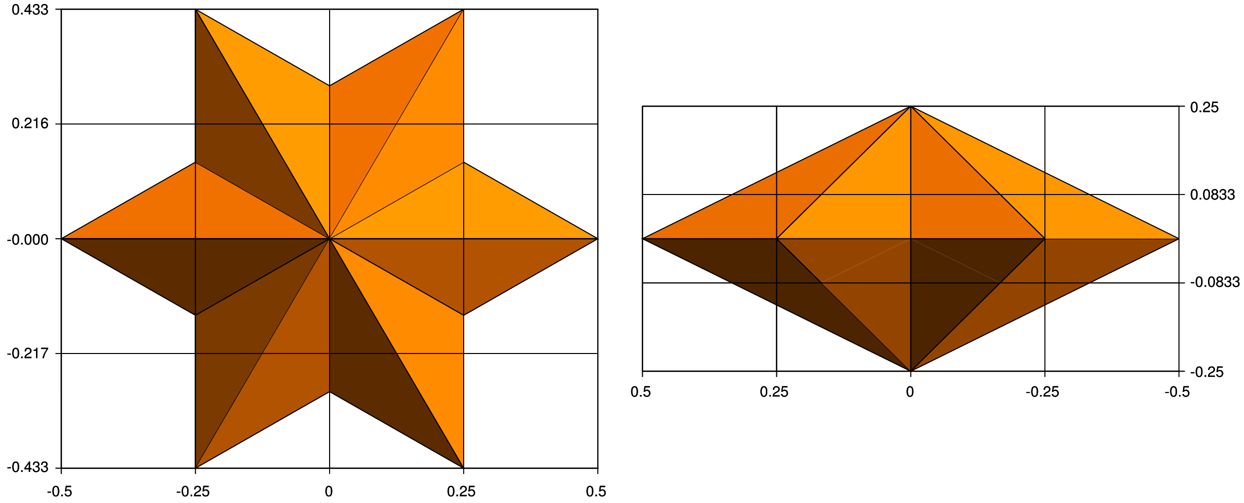

Fig. 13. Example 3.3: Geometry and initial mesh $\mathcal{T}_0$ with 24 elements of the star example. The point of view is along the z-axis (left) and along the y-axis (right).

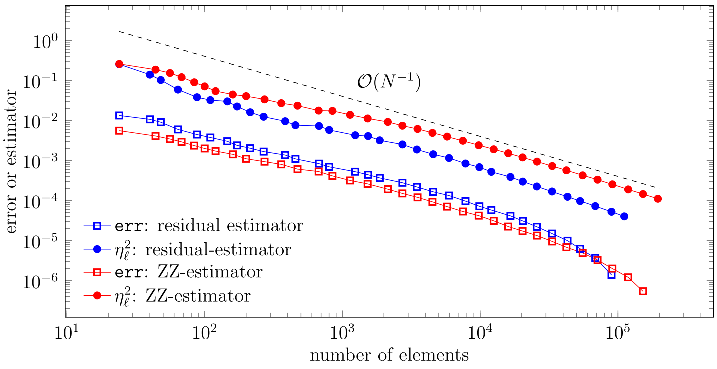

Fig. 14. Example 3.3: Convergence of the error estimator η_ℓ^2 and capacity error $\texttt{err}_\ell$. The estimator η_ℓ^2 as well as the error converge with optimal order $\mathcal{O}(N^{-1})$, if a weighted-residual or the proposed ZZ-based error estimator is used.

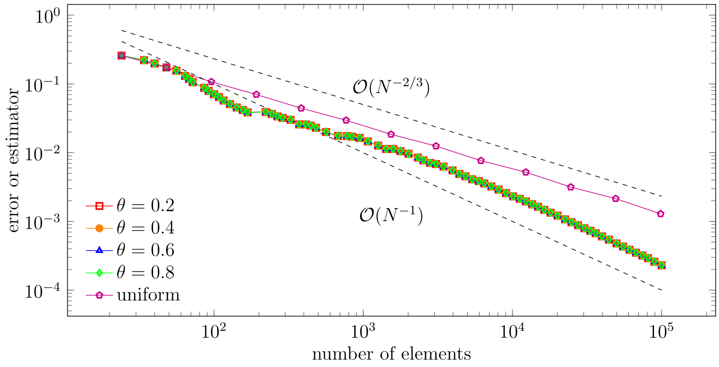

Fig. 15. Example 3.3:Convergence of the error estimator η_ℓ^2 for different values of $0 < \theta < 1$ and uniform refinement $\theta = 1$. Uniform refinement leads to a reduced rate of convergence.

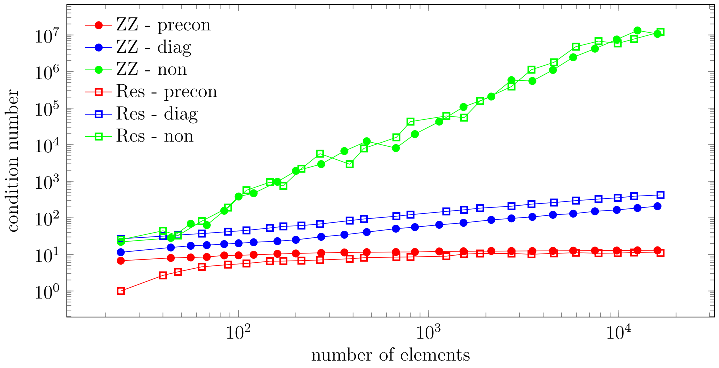

Fig. 16. Example 3.3: Condition number for different preconditioning strategies for the arising linear systems in Algorithm 1 (ZZ) with operator preconditioning with the hypersingular operators vs. Algorithm (Res) with multilevel additive Schwarz preconditioning. Additionally, diagonal preconditioning is employed in both cases.

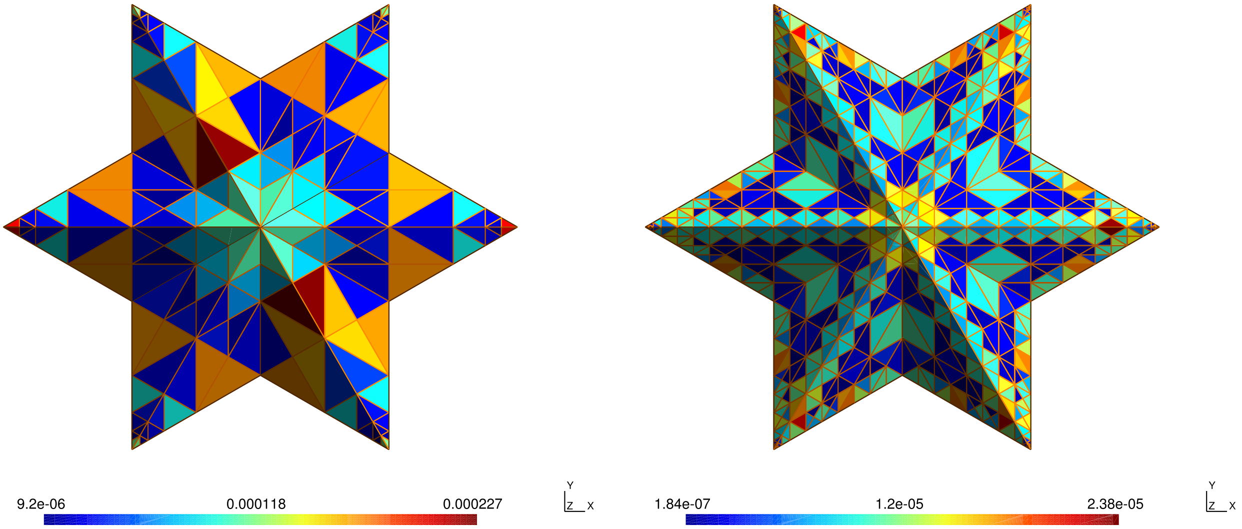

Fig. 17. Example 3.3: Mesh $\mathcal{T}_\ell$ and distribution of the error estimator η_ℓ^2 for $\ell = 10$ (left) with 360 elements and $\ell = 15$ (right) with 1524 elements.

Acknowledgements

The research of AH and DP is funded by the Austrian Science Fund (FWF) by the research project *Optimal adaptivity for BEM and FEM-BEM coupling* (grant P27005) and the special research program *Taming complexity in PDE systems* (grant SFB F65).

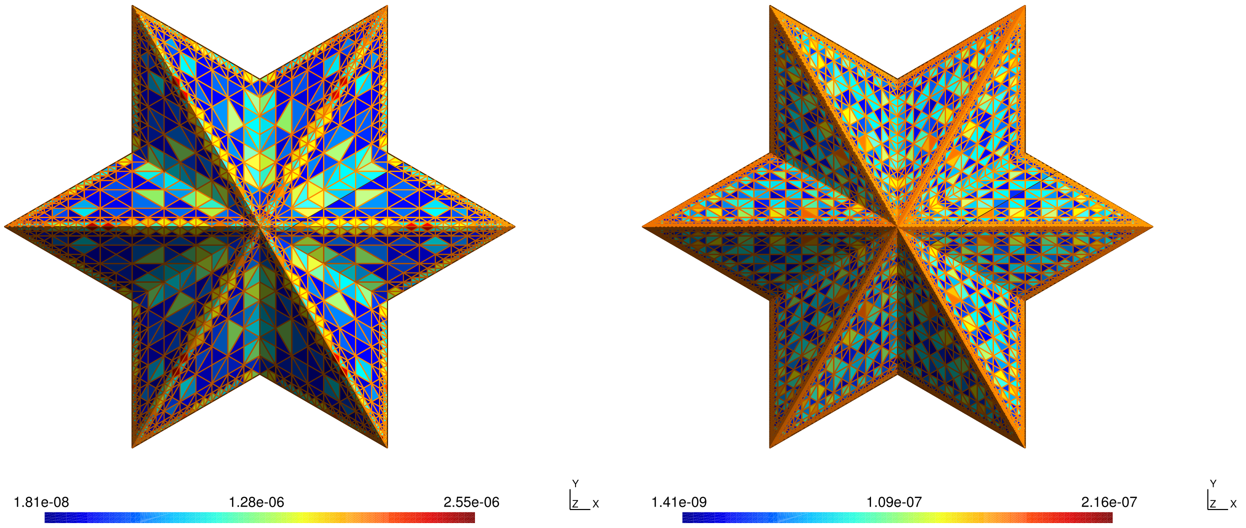

Fig. 18. Example 3.3: Mesh $\mathcal{T}_\ell$ and distribution of the error estimator η_ℓ^2 for $\ell = 20$ (left) with 5754 elements and $\ell = 25$ (right) with 20304 elements.

Table 3
Example 3.3: Numerical results of Algorithm 1 with $\theta = 1/2$.

step ℓ	number of elements $\#\mathcal{T}_\ell$	Cap_ℓ	η_ℓ^2	$\lvert\mathrm{Cap}_{\ell_{\max}} - \mathrm{Cap}_\ell\rvert$	# iterations
0	24	0.2942967843	$2.580 \cdot 10^{-1}$	$5.456 \cdot 10^{-3}$	5
5	100	0.2978622929	$7.092 \cdot 10^{-2}$	$5.456 \cdot 10^{-3}$	5
14	1136	0.2995412185	$1.384 \cdot 10^{-2}$	$1.686 \cdot 10^{-3}$	20
22	9752	0.2998567063	$2.402 \cdot 10^{-3}$	$3.223 \cdot 10^{-4}$	20
31	91046	0.2998234547	$2.540 \cdot 10^{-4}$	$2.869 \cdot 10^{-5}$	22
34	194838	0.2998587081	$1.110 \cdot 10^{-4}$		

Appendix A. Weighted-residual error estimator

In this section, we briefly sketch the state of the art for adaptive BEM driven by the weighted-residual error estimator. We consider the model problem (8) with right-hand side $f \in H^1(\Gamma)$. Note that the additional regularity of f (instead of only $f \in H^{1/2}(\Gamma)$) is required for the well-posedness of the weighted-residual error estimator introduced below. Here, $\mathcal{T}_\ell$ is a conforming triangulation of Γ into plane (closed) surface triangles. The Lax–Milgram lemma guarantees existence and uniqueness of $\Phi_\ell \in \mathcal{P}^0(\mathcal{T}_\ell)$ such that

$$\langle V\Phi_\ell\,,\,\Psi_\ell\rangle_\Gamma = \langle f\,,\,\Psi_\ell\rangle_\Gamma \quad \text{for all } \Psi_\ell \in \mathcal{P}^0(\mathcal{T}_\ell). \tag{18}$$

Unlike above, we note that the discrete solution is now computed on the primal triangulation instead of the dual triangulation. With ∇ being the surface gradient, the local contributions of the weighted-residual error estimator read

$$\mu_\ell(T) := \mathrm{diam}(T)^{1/2}\,\|\nabla(f - V\Phi_\ell)\|_{L^2(T)} \qquad \text{for all } T \in \mathcal{T}_\ell. \tag{19}$$

We note that reliability [7] and efficiency [1] hold in the sense of

$$(C'_{\mathrm{rel}})^{-1}\,\|\phi - \Phi_\ell\|_{H^{-1/2}(\Gamma)} \le \mu_\ell := \Big(\sum_{T\in\mathcal{T}_\ell} \mu_\ell(T)^2\Big)^{1/2} \le C'_{\mathrm{eff}}\,\|h_\ell^{1/2}(\phi - \Phi_\ell)\|_{L^2(\Gamma)}, \tag{20}$$

where the constants $C'_{\mathrm{rel}}, C'_{\mathrm{eff}} > 0$ depend only on Γ and γ-shape regularity of $\mathcal{T}_\ell$. As above, h_ℓ denotes the local mesh-size function $h_\ell|_T := \mathrm{diam}(T)$ for all $T \in \mathcal{T}_\ell$. Based on μ_ℓ, the standard adaptive algorithm from [11,16,18] reads as follows:

Algorithm 5. Input: Initial conforming triangulation $\mathcal{T}_0$ of Γ into plane surface triangles, adaptivity parameters $0 < \theta \le 1$ and $C_{\mathrm{mark}} \ge 1$.
Adaptive loop. Iterate the following steps (i)–(iv) for all $\ell = 0, 1, 2, \ldots$:

(i) Compute the Galerkin solution $\Phi_\ell \in \mathcal{P}^0(\mathcal{T}_\ell)$ from (18).
(ii) Compute the local contributions $\mu_\ell(T)$ from (19) for all $T \in \mathcal{T}_\ell$.

(iii) Determine a set of, up to the multiplicative constant $C_{\rm mark}$, minimal cardinality, which satisfies the Dörfler marking criterion $\theta \sum_{T\in\mathcal{M}_\ell} \mu_\ell(T)^2 \le \mu_\ell^2$.
(iv) Use NVB and refine all marked elements $T \in \mathcal{M}_\ell$ to obtain $\mathcal{T}_{\ell+1} := \texttt{refine}(\mathcal{T}_\ell, \mathcal{M}_\ell)$.

Output: Sequence of successively refined triangulations $\mathcal{T}_\ell$ as well as corresponding Galerkin solutions $\Phi_\ell \in \mathcal{P}^0(\mathcal{T}_\ell)$ and weighted-residual error estimators μ_ℓ. □

The following theorem is the main result from [11,16,18]:

Theorem 6. *For all $0 < \theta \le 1$, there exist constants $C_{\rm lin} > 0$ and $0 < q_{\rm lin} < 1$ such that the output of Algorithm 1 is linearly convergent in the sense of*

$$\eta_{\ell+n} \le C_{\rm lin} q_{\rm lin}^n \,\eta_\ell \quad \text{for all } \ell, n \in \mathbb{N}_0. \tag{21}$$

Moreover, there exists a constant $0 < \theta_{\rm opt} < 1$ such that for all $0 < \theta < \theta_{\rm opt}$, the output of Algorithm 5 converges with the optimal algebraic rate in the following sense: For all $s > 0$, define

$$\mathbb{A}_s(\phi) := \sup_{N \ge \#\mathcal{T}_0} \min_{\substack{\mathcal{T}_{\rm opt}\in\texttt{refine}(\mathcal{T}_0)\\ \#\mathcal{T}_{\rm opt}\le N}} N^s \mu_{\rm opt} \in [0, \infty] \tag{22}$$

where $\texttt{refine}(\mathcal{T}_0)$ denotes the set of all possible NVB refinements of $\mathcal{T}_0$ and $\mathcal{T}_{\rm opt}$ corresponds to the estimator $\mu_{\rm opt}$. Then, there exists constants $c_{\rm opt}, C_{\rm opt} > 0$ such that

$$c_{\rm opt}^{-1}\,\mathbb{A}_s(\phi) \le \sup_{\ell\in\mathbb{N}_0} (\#\mathcal{T}_\ell)^s \,\mu_\ell \le C_{\rm opt}\,\mathbb{A}_s(\phi),$$

i.e., Algorithm 5 asymptotically realizes (and even characterizes) each possible algebraic convergence rate $s > 0$ and therefore converges with the optimal rate. The constants $C_{\rm lin}$ and $q_{\rm lin}$ depend only on Γ, γ-shape regularity for $\mathcal{T}_0$, and θ. While $c_{\rm opt}$ is generic, $C_{\rm opt}$ depends only on $\mathcal{T}_0$, $C_{\rm lin}$, $q_{\rm lin}$, $C_{\rm mark}$, and s. □

Remark 7. The recent work [15] proves that Theorem 6 remains valid, if (18) is solved inexactly by PCG with optimal additive Schwarz preconditioner. With the PCG iterates $\Phi_{\ell,k} \approx \Phi_\ell$, the iterative solver is stopped if $|\!|\!|\, \Phi_{\ell,k-1} - \Phi_{\ell,k} \,|\!|\!| \le \lambda \mu_\ell(\Phi_{\ell,k})$, where the residual error estimator is evaluated at $\Phi_{\ell,k}$ instead of the exact Galerkin solution. Then, linear convergence (21) holds for any $\lambda > 0$, while optimal rates require $\lambda \ll 1$ to be sufficiently small.

To empirically measure the performance of Algorithm 1 in the numerical experiments of Section 3, we consider the output of Algorithm 5 for $f = 1$ as benchmark.

Corollary 8. *Consider the output of Algorithm 5 for $f = 1$ and arbitrary $0 < \theta \le 1$ and $C_{\rm mark} \ge 1$. Define the discrete capacity*

$$\mathrm{Cap}'_\ell(\Omega) := \frac{1}{4\pi}\,\langle \Phi_\ell \,,\, 1\rangle_\Gamma. \tag{23}$$

Then, there holds (17) as well as monotonicity $0 < \mathrm{Cap}'_\ell(\Omega) \le \mathrm{Cap}'_{\ell+1} \le \mathrm{Cap}(\Omega)$ for all $\ell \in \mathbb{N}_0$ and, in particular, convergence

$$0 \le \mathrm{Cap}(\Omega) - \mathrm{Cap}'_\ell(\Omega) \le C_{\rm cap}\,\mu_\ell^2 \to 0 \quad \text{as } \ell \to \infty. \tag{24}$$

Moreover, if $0 < \theta < \theta_{\rm opt}$ is sufficiently small in the sense of Theorem 6, it follows that

$$0 \le \mathrm{Cap}(\Omega) - \mathrm{Cap}'_\ell(\Omega) \le C'_{\rm cap}\,\mathbb{A}_s(\phi)^2\,(\#\mathcal{T}_\ell)^{-2s} \quad \text{for all } \ell \in \mathbb{N}_0. \tag{25}$$

The constant $C_{\rm cap}$ depends only on Γ and γ-shape regularity of $\mathcal{T}_\ell$, and $C'_{\rm cap} := C_{\rm cap}C_{\rm opt}^2$ with $C_{\rm opt}$ from Theorem 6.

Proof. Arguing as in the proof of Proposition 4, we see that

$$0 < \mathrm{Cap}'_\ell(\Omega) \le \mathrm{Cap}(\Omega) \quad \text{as well as} \quad 0 \le \mathrm{Cap}(\Omega) - \mathrm{Cap}'_\ell(\Omega) \simeq \|\phi - \Phi_\ell\|^2_{H^{-1/2}(\Gamma)}.$$

Therefore, (24)–(25) are immediate consequences of Theorem 6 and reliability (20). Overall, it only remains to prove monotonicity $\mathrm{Cap}'_\ell(\Omega) \le \mathrm{Cap}'_{\ell+1}(\Omega)$ for all $\ell \in \mathbb{N}_0$: Note that $\mathcal{P}^0(\mathcal{T}_\ell) \subseteq \mathcal{P}^0(\mathcal{T}_{\ell+1})$, which fails for the sequence of dual triangulations, i.e., $\mathcal{P}^0(\mathcal{T}_\ell^{\rm dual}) \not\subseteq \mathcal{P}^0(\mathcal{T}_{\ell+1}^{\rm dual})$. With this additional nestedness, it holds that $\Phi_{\ell+1} - \Phi_\ell \in \mathcal{P}^0(\mathcal{T}_{\ell+1})$. Therefore, there holds the discrete Galerkin orthogonality $\langle V(\Phi_{\ell+1} - \Phi_\ell)\,,\, \Psi_{\ell+1}\rangle_\Gamma = 0$ for all $\Psi_{\ell+1} \in \mathcal{P}^0(\mathcal{T}_{\ell+1})$. This implies the Pythagoras theorem

$$|||\, \phi - \Phi_\ell \,|||^2 = |||\, \phi - \Phi_{\ell+1} \,|||^2 + |||\, \Phi_{\ell+1} - \Phi_\ell \,|||^2$$

with respect to the energy norm. Arguing as for the proof of (17), we thus see that

$$4\pi \left(\mathrm{Cap}_{\ell+1}(\Omega) - \mathrm{Cap}_{\ell}(\Omega) \right) = |||\, \Phi_{\ell+1} - \Phi_\ell \,|||^2 \geq 0.$$

This concludes the proof. □

Finally, we refer to [12] for the extension of Theorem 6 to the hypersingular operator associated with the Laplace problem and to [2] for optimality in the case of the Helmholtz equation. The work [14] gives a first mathematical proof of optimal algebraic convergence rates for the point-wise approximation of the PDE-solution via BEM and potential operators. Moreover, [17] generalizes the latter approach to general goal-oriented adaptivity with linear goal functional.

Appendix B. Implementation with Bempp

The following code demonstrates a possible implementation of Algorithm 1 in Python with the Bempp library. The demonstration code can also be downloaded from the Bempp homepage https://bempp.com.

```
import bempp.api
import numpy as np

max_el = 600000 #max number elements
step_counter = 0
theta = 0.5 #marking parameter
grid = bempp.api.shapes.cube(h=0.1) #import initial grid
#initialize function for the right-hand side
def one_fun(x, n, domain_index, res):
    res[0] = -1

##Adaptive loop
while (grid.leaf_view.entity_count(0) <= max_el):
    step_counter += 1
    bary_grid = grid.barycentric_grid() #barycentric refinement

    ##initialize function spaces
    const_space = bempp.api.function_space(grid, ’DP’, 0)
    lin_space = bempp.api.function_space(grid, ’P’,1)
    bary_space = bempp.api.function_space(bary_grid, ’DP’, 0)
    bary_lin_space = bempp.api.function_space(bary_grid, ’DP’,1)

    ##initialize operators
    base_slp = bempp.api.operators.boundary.laplace.single_layer(bary_space, bary_space, bary_space)
    slp, hyp = bempp.api.operators.boundary.laplace.single_layer_and_hypersingular_pair(
            grid, spaces=’dual’, base_slp=base_slp,stabilization_factor=1)

    rank_one_op = bempp.api.RankOneBoundaryOperator(hyp.domain, hyp.range, hyp.dual_to_range)
    hyp_regularized = hyp + rank_one_op

    ##initialize right-hand side
    rhs_fun = bempp.api.GridFunction(slp.range, fun=one_fun)

    ##initialize preconditioned lhs
    lhs = slp * hyp_regularized
    discrete_op = lhs.strong_form()

    ##solve via GMRES
    from scipy.sparse.linalg import gmres
    number_of_iterations = 0
    def callback(x):
        global number_of_iterations
        number_of_iterations += 1

    phi_vec, info = gmres(discrete_op, rhs_fun.coefficients,tol=1e-10, callback=callback)
    phi_fun = hyp_regularized * bempp.api.GridFunction(hyp.domain, coefficients=phi_vec)

    print("Number of iterations: {0}".format(number_of_iterations))

    ##compute the capacity
    capacity=-phi_fun.integrate()[0,0]
    print("The capacity is {0}.".format(capacity))

    ##compute the error estimator
    bary_map = grid.barycentric_descendents_map()
    bary_index_set = bary_grid.leaf_view.index_set()

    #create sorted list of elements
    index_set = grid.leaf_view.index_set()
    elements = grid.leaf_view.entity_count(0) * [None]
    for element in grid.leaf_view.entity_iterator(0):
        elements[index_set.entity_index(element)] = element

    local_est_bary = np.zeros(bary_grid.leaf_view.entity_count(0), dtype='float64')
    local_est = np.zeros(grid.leaf_view.entity_count(0), dtype='float64')
    new_space = bempp.api.function_space(grid,"B-P",1)

    map_dual_to_bary = bempp.api.operators.boundary.sparse.identity(
            phi_fun.space,bary_lin_space,bary_lin_space)
    map_lin_to_bary_lin =  bempp.api.operators.boundary.sparse.identity(
            new_space,bary_lin_space,bary_lin_space)

    #interpret function values on dual elements as node-values in S^1
    Iphi_coefficients = phi_fun.coefficients
    Iphi = bempp.api.GridFunction(new_space, coefficients=Iphi_coefficients)

    #lift both functions to P^1(T_l^bary)
    phi = map_dual_to_bary * phi_fun
    Iphi_lin_bary = map_lin_to_bary_lin * Iphi

    zz_diff = phi - Iphi_lin_bary

    #compute the L^2-norm in each barycentric element
    for element in bary_grid.leaf_view.entity_iterator(0):
        index = bary_index_set.entity_index(element)
        local_est_bary [index] = zz_diff.l2_norm(element)**2

    #add up all barycentric contributions for each primal element
    for m in range(bary_map.shape[0]):
        element = grid.leaf_view.element_from_index(m)
        mesh_size = element.geometry.volume
        for n in range(bary_map.shape[1]):
            local_est[m] += local_est_bary[bary_map[m, n]]

        local_est[m] = local_est[m]*mesh_size**(1./2) #scale with primal mesh-size

    total_est = np.sum(local_est)

    print("Squared ZZ-Estimator: {0}".format(total_est))

    ##Doerfler-Marking and refining
    idx = np.argsort((-1)*local_est)
    ind_on_marked =0
    counter = 0

    while (ind_on_marked < theta * total_est):
        ind_on_marked += local_est[idx[counter]]
        grid.mark(elements[idx[counter]])
        counter += 1

    grid = grid.refine() # refine grid
    print("New number of elements {0}".format(grid.leaf_view.entity_count(0)))
```

References

[1] Markus Aurada, Michael Feischl, Thomas Führer, Michael Karkulik, J. Markus Melenk, Dirk Praetorius, Local inverse estimates for non-local boundary integral operators, Math. Comput. 86 (308) (2017) 2651–2686.

[2] Alex Bespalov, Timo Betcke, Alexander Haberl, Dirk Praetorius, Adaptive BEM with optimal convergence rates for the Helmholtz equation, Comput. Methods Appl. Mech. Eng. 346 (2019) 260–287.

[3] Sören Bartels, Carsten Carstensen, Each averaging technique yields reliable a posteriori error control in FEM on unstructured grids. I. Low order conforming, nonconforming, and mixed FEM, Math. Comput. 71 (239) (2002) 945–969.

[4] Mario Bebendorf, Hierarchical Matrices, Lecture Notes in Computational Science and Engineering., vol. 63, Springer-Verlag, Berlin, 2008.

[5] Steffen Börm, Efficient Numerical Methods for Non-local Operators: $\mathcal{H}^2$-Matrix Compression, Algorithms and Analysis, European Mathematical Society (EMS), Zürich, 2010.

[6] Lehel Banjai, Lloyd N. Trefethen, A multipole method for Schwarz-Christoffel mapping of polygons with thousands of sides, SIAM J. Sci. Comput. 25 (3) (2003) 1042–1065.

[7] CarstenCarstensen, Matthias Maischak, Ernst P. Stephan, A posteriori error estimate and h-adaptive algorithm on surfaces for Symm's integral equation, Numer. Math. 90 (2) (2001) 197–213.

[8] Willy Dörfler, A convergent adaptive algorithm for Poisson's equation, SIAM J. Numer. Anal. 33 (3) (1996) 1106–1124.

[9] Tobin A. Driscoll, Lloyd N. Trefethen, Schwarz-Christoffel Mapping, Cambridge University Press, Cambridge, 2002.

[10] Michael Feischl, Thomas Führer, Norbert Heuer, Michael Karkulik, Dirk Praetorius, Adaptive boundary element methods, Arch. Comput. Methods Eng. 22 (3) (Jul 2015) 309–389.

[11] Michael Feischl, Thomas Führer, Michael Karkulik, Jens Markus Melenk, Dirk Praetorius, Quasi-optimal convergence rates for adaptive boundary element methods with data approximation, part I: weakly-singular integral equation, Calcolo 51 (4) (2014) 531–562.

[12] Michael Feischl, Thomas Führer, Michael Karkulik, J. Markus Melenk, Dirk Praetorius, Quasi-optimal convergence rates for adaptive boundary element methods with data approximation. Part II: hyper-singular integral equation, Electron. Trans. Numer. Anal. 44 (2015) 153–176.

[13] Michael Feischl, Thomas Führer, Michael Karkulik, Dirk Praetorius, ZZ-type a posteriori error estimators for adaptive boundary element methods on a curve, Eng. Anal. Bound. Elem. 38 (2014) 49–60.

[14] Michael Feischl, Gregor Gantner, Alexander Haberl, Dirk Praetorius, Thomas Führer, Adaptive boundary element methods for optimal convergence of point errors, Numer. Math. 132 (3) (2016) 541–567.

[15] Thomas Führer, Alexander Haberl, Dirk Praetorius, Stefan Schimanko, Adaptive BEM with inexact PCG solver yields almost optimal computational costs, Numer. Math. 141 (2019) 967–1008.

[16] Michael Feischl, Michael Karkulik, Jens Markus Melenk, Dirk Praetorius, Quasi-optimal convergence rate for an adaptive boundary element method, SIAM J. Numer. Anal. 51 (2) (2013) 1327–1348.

[17] Michael Feischl, Dirk Praetorius, Kristoffer G. van der Zee, An abstract analysis of optimal goal-oriented adaptivity, SIAM J. Numer. Anal. 54 (3) (2016) 1423–1448.
[18] Tsogtgerel Gantumur, Adaptive boundary element methods with convergence rates, Numer. Math. 124 (3) (2013) 471–516.
[19] Samuel P. Groth, Anthony J. Baran, Timo Betcke, Stephan Havemann, Wojciech Śmigaj, The boundary element method for light scattering by ice crystals and its implementation in bem++, J. Quant. Spectrosc. Radiat. Transf. 167 (Supplement C) (2015) 40–52.
[20] Leslie Greengard, Denis Gueyffier, Per-Gunnar Martinsson, Vladimir Rokhlin, Fast direct solvers for integral equations in complex three-dimensional domains, Acta Numer. 18 (2009) 243–275.
[21] Ivan G. Graham, William McLean, Anisotropic mesh refinement: the conditioning of Galerkin boundary element matrices and simple preconditioners, SIAM J. Numer. Anal. 44 (4) (2006) 1487–1513.
[22] Leslie Greengard, Vladimir Rokhlin, A new version of the fast multipole method for the Laplace equation in three dimensions, in: Acta Numerica, in: Acta Numer, vol. 6, Cambridge Univ. Press, Cambridge, 1997, pp. 229–269.
[23] Joachim Gwinner, Ernst Peter Stephan, Advanced Boundary Element Methods, Springer, Cham, 2018.
[24] Wolfgang Hackbusch, Hierarchical Matrices: Algorithms and Analysis, Springer, Heidelberg, 2015.
[25] Ralf Hiptmair, Operator preconditioning, Comput. Math. Appl. 52 (5) (2006) 699–706.
[26] Ralf Hiptmair, Carlos Jerez-Hanckes, Carolina Urzúa-Torres, Mesh-independent operator preconditioning for boundary elements on open curves, SIAM J. Numer. Anal. 52 (5) (2014) 2295–2314.
[27] Chi-Ok Hwang, Michael Mascagni, Taeyoung Won Monte, Carlo methods for computing the capacitance of the unit cube, Math. Comput. Simul. 80 (6) (2010) 1089–1095.
[28] Johan Helsing, Karl-Mikael Perfekt, On the polarizability and capacitance of the cube, Appl. Comput. Harmon. Anal. 34 (3) (2013) 445–468.
[29] George C. Hsiao, Wolfgang L. Wendland, Boundary Integral Equations, Springer, Berlin, 2008.
[30] Oliver Dimon Kellogg, Foundations of Potential Theory, Reprint from the first edition of 1929. Springer, Berlin, 1967.
[31] Michael Karkulik, David Pavlicek, Dirk Praetorius, On 2D newest vertex bisection: optimality of mesh-closure and H^1-stability of L_2-projection, Constr. Approx. 38 (2) (2013) 213–234.
[32] William McLean, Strongly Elliptic Systems and Boundary Integral Equations, Cambridge University Press, Cambridge, 2000.
[33] Günther Of, Olaf Steinbach, Wolfgang L. Wendland, The fast multipole method for the symmetric boundary integral formulation, IMA J. Numer. Anal. 26 (2) (2006) 272–296.
[34] George Pólya, Estimating electrostatic capacity, Am. Math. Mon. 54 (4) (1947) 201–206.
[35] Frank H. Read, Improved extrapolation technique in the boundary element method to find the capacitances of the unit square and cube, J. Comp. Physiol. 133 (1) (1997) 1–5.
[36] Rodolfo Rodriguez, Some remarks on Zienkiewicz–Zhu estimator, Numer. Methods Partial Differ. Equ. 10 (5) (1994) 625–635.
[37] Wojciech Śmigaj, Timo Betcke, Simon Arridge, Joel Phillips, Martin Schweiger, Solving boundary integral problems with BEM++, ACM Trans. Math. Softw. 41 (2) (2015), Art. 6, 40.
[38] Stefan A. Sauter, Christoph Schwab, Boundary Element Methods, Springer, Berlin, 2011.
[39] Rob Stevenson, Optimality of a standard adaptive finite element method, Found. Comput. Math. 7 (2) (2007) 245–269.
[40] Olaf Steinbach, Numerical Approximation Methods for Elliptic Boundary Value Problems, Springer, New York, 2008.
[41] Olaf Steinbach, Wolfgang L. Wendland, The construction of some efficient preconditioners in the boundary element method, Adv. Comput. Math. 9 (1–2) (1998) 191–216.
[42] Elwin van 't Wout, Pierre Gélat, Timo Betcke, Simon Arridge, A fast boundary element method for the scattering analysis of high-intensity focused ultrasound, J. Acoust. Soc. Am. 138 (5) (2015) 2726–2737.
[43] Oleg Zienkiewicz, Jian Z. Zhu, A simple error estimator and adaptive procedure for practical engineering analysis, Int. J. Numer. Methods Eng. 24 (2) (1987) 337–357.