ASC Report No. 24/2011

HILBERT-A MATLAB Implementation of Adaptive 2D-BEM

Markus Aurada, Michael Ebner, Michael Feischl, Samuel Ferraz-Leite, Petra Goldenits, Michael Karkulik, Markus Mayr, Dirk Praetorius

Institute for Analysis and Scientific Computing
Vienna University of Technology — TU Wien
www.asc.tuwien.ac.at ISBN 978-3-902627-04-9

Most recent ASC Reports

23/2011 *JinMyong Kim, Anton Arnold, Xiaohua Yao*
Estimates for a class of oscillatory integrals and decay rates for wave-type equations

22/2011 *Markus Aurada, Michael Feischl, Michael Karkulik, Dirk Praetorius*
Adaptive coupling of FEM and BEM: Simple error estimators and convergence

21/2011 *Michael Feischl, Michael Karkulik, Jens Markus Melenk, Dirk Praetorius*
Residual a-posteriori error stimates in BEM: Convergence of h-adaptive algorithms

20/2011 *Markus Aurada, Michael Feischl, Michael Karkulik, Dirk Praetorius*
Adaptive coupling of FEM and BEM: Simple error estimators and convergence

19/2011 *Petra Goldenits, Dirk Praetorius, Dieter Suess*
Convergent geometric integrator for the Landau-Lifshitz-Gilbert equation in micromagnetics

18/2011 *M. Aurada, M. Feischl, M. Karkulik, D. Praetorius*
A Posteriori Error Estimates for the Johnson-Nédélec FEM-BEM Coupling

17/2011 *Michael Feischl, Marcus Page, Dirk Praetorius*
Convergence of adaptive FEM for elliptic obstacle problems

16/2011 *Michael Feischl, Marcus Page, Dirk Praetorius*
Convergence and quasi-optimality of adaptive FEM with inhomogeneous Dirichlet data

15/2011 *M. Huber, A. Pechstein and J. Schöberl*
Hybrid domain decomposition solvers for scalar and vectorial wave equation

14/2011 *Ansgar Jüngel, José Luis López, Jesús Montejo-Gámez*
A new derivation of the quantum Navier-Stokes equations in the Wigner-Fokker-Planck approach

Institute for Analysis and Scientific Computing
Vienna University of Technology
Wiedner Hauptstraße 8–10
1040 Wien, Austria

E-Mail: admin@asc.tuwien.ac.at
WWW: http://www.asc.tuwien.ac.at
FAX: +43-1-58801-10196

ISBN 978-3-902627-04-9

ASC
TU WIEN

HILBERT — A MATLAB Implementation of Adaptive 2D-BEM

HILBERT Is a Lovely Boundary Element Research Tool

Markus Aurada[1] Michael Ebner[1] Michael Feischl[1] Samuel Ferraz-Leite[2]
Petra Goldenits[1] Michael Karkulik[1] Markus Mayr[1] Dirk Praetorius[1]

[1] Viennat University of Technology
Wiedner Hauptstraße 8-10, A-1040 Vienna, Austria
hilbert@asc.tuwien.ac.at http://www.asc.tuwien.ac.at/abem

[2] Max-Planck-Institute for Mathematics in the Sciences
Inselstraße 22, 04103 Leipzig, Germany
samuel.ferraz-leite@mis.mpg.de

June 9, 2011

Abstract

We report on the Matlab program package HILBERT. It provides an easy-accessible implementation of lowest order adaptive Galerkin boundary element methods for the numerical solution of the Poisson equation in 2D. The library was designed to serve several purposes: The stable implementation of the integral operators may be used in research code. The framework of Matlab ensures usability in lectures on boundary element methods or scientific computing. Finally, we aim at emphasizing the use of adaptivity as general concept and for boundary element methods in particular.

In this work we summarize recent analytical results on adaptivity in the context of BEM and give sample code to illustrate the use of HILBERT. Various benchmarks are performed to empirically analyze the performance of the proposed adaptive algorithms and to compare adaptive and uniform mesh-refinement. In particular, we do not only focus on mathematical convergence behavior but also on the usage of critical system resources such as memory consumption and computational time. In any case, the superiority of the proposed adaptive approach is empirically supported.

Keywords: boundary element methods, adaptive mesh-refinement, a posteriori error estimation, Matlab implementation

Mathematics Subject Classification (2000): 65N38, 65Y20, 65N50

1 Introduction

In many applications, the (Galerkin) boundary element method (BEM) has established itself as possible strategy for the numerical solution of a certain class of partial differential equations. One of the strengths of BEM is its potentially high order of convergence: For smooth analytical solutions, the error of the lowest-order approximation behaves like $\mathcal{O}(h^{3/2})$ with h denoting the global mesh-size of a partition of the boundary. However, due to generic singularities of stresses and/or fluxes on the boundary, this superlinear convergence is usually not observed with uniform meshes in practice.

The Matlab program package HILBERT has been designed to make *adaptive* BEM (ABEM) more accessible to a broader audience. It provides stable implementations of the discrete boundary integral operators corresponding to the Laplace operator in 2D. Various $(h-h/2)$-type error estimators as well as mesh-refining algorithms, functions for the discretization of given boundary data, visualization routines, and many other functions necessary for the implementation and use of ABEM for the Poisson problem are provided.

Several benchmarks demonstrate and compare the behavior of adaptive and uniform BEM in presence of different types of singularities. In our experiments, we observe that our adaptive algorithm resolves

the singularities of both, analytical solution and given data. Throughout, it reveals the optimal order of convergence. Moreover, the reachable accuracy – usually limited by system resources such as time and memory consumption – is increased dramatically by use of adaptivity.

1.1 Model problem

As model problem, we consider the Poisson equation on some bounded Lipschitz domain $\Omega \subset \mathbb{R}^2$ with polygonal boundary $\Gamma = \partial\Omega$. Each solution $u \in H^1(\Omega) := \{v \in L^2(\Omega) \,|\, \nabla v \in L^2(\Omega)\}$ of

$$-\Delta u = f \quad \text{in } \Omega \tag{1}$$

can explicitly be written in the form

$$u(x) = \widetilde{N} f(x) + \widetilde{V}\phi(x) - \widetilde{K} g(x) \quad \text{for all } x \in \Omega, \tag{2}$$

where $g := u|_\Gamma \in H^{1/2}(\Gamma) := \{v \in L^2(\Gamma) \,|\, \text{there is an extension } \widetilde{v} \in H^1(\Omega)\}$ is the trace of u and $\phi := \partial_n u \in H^{-1/2}(\Gamma) := H^{1/2}(\Gamma)^*$ is the normal derivative of u on Γ, see e.g. [17, 19, 20]. The involved linear integral operators read

$$\widetilde{N} f(x) := -\frac{1}{2\pi} \int_\Omega \log|x-y|\, f(y)\, dy, \tag{3}$$

$$\widetilde{V}\phi(x) := -\frac{1}{2\pi} \int_\Gamma \log|x-y|\, \phi(y)\, d\Gamma(y), \tag{4}$$

$$\widetilde{K} g(x) := -\frac{1}{2\pi} \int_\Gamma \frac{(y-x)\cdot n_y}{|y-x|^2}\, g(y)\, d\Gamma(y), \tag{5}$$

where n_y denotes the outer unit vector of Ω at some point $y \in \Gamma$ and where $\int_\Gamma d\Gamma(\cdot)$ denotes integration along the boundary. Put differently, the solution u of (1) is known as soon as the Cauchy data $(u|_\Gamma, \partial_n u)$ are known on the entire boundary Γ.

If one considers the trace and the normal derivative of u, the so-called *representation formula* (2) becomes the *Calderón system*

$$\begin{aligned} g &= N_0 f + V\phi - (K - 1/2)g, \\ \phi &= N_1 f + (K' + 1/2)\phi + W g. \end{aligned} \tag{6}$$

It involves six linear integral operators acting only on Γ: The simple-layer potential V, the double-layer potential K with adjoint operator K', the hypersingular integral operator W, as well as the trace N_0 and the normal derivative N_1 of the Newtonian potential $\widetilde{N}$.

For symmetric BEM, the Poisson equation with given boundary data is equivalently stated in terms of the Calderón system (6), which leads to a boundary integral equation (BIE) on Γ. In order to solve this BIE, we discretize a variational formulation by a Galerkin method. To that end, let $E_1, \dots, E_n$ be a partition of the boundary Γ. We then use piecewise constant functions to discretize fluxes and piecewise linear and globally continuous functions for stresses. In a post-processing step, the computed Cauchy data are then plugged into the representation formula (2) to obtain an approximation of the solution u of the differential equation (1).

1.2 Outline of the paper

First, Section 2 deals with an abstract formulation of the proposed adaptive algorithm. Preliminaries of practical relevance, such as mesh administration with `HILBERT` and comments on the stable implementation of the discrete boundary integral operators, are given in Section 3. In the following Sections 4–5, we give possible choices of error indicators for the boundary integral formulation of the Dirichlet and the Neumann problem. Finally, Section 6 deals with a mixed boundary value problem and non-homogeneous volume forces $f \neq 0$. Sections 4–6 are supported each by its own set of numerical experiments. Some conclusions and a summary of the functionality of `HILBERT` finally conclude the paper in Section 7.

2 Adaptive algorithm

Using a lowest order ansatz, BEM is known to converge optimally with order $\mathcal{O}(h^{3/2})$ if the data and unknown solution are sufficiently smooth, e.g., piecewise regularity $\phi \in H^1, g \in H^2, f \in H^1$ [19]. However, fluxes and stresses on the boundary usually lack the regularity necessary to observe the optimal order with uniform meshes. We therefore aim at providing a fully adaptive mesh-refining algorithm that leads to more efficient use of available system resources.

2.1 $(h - h/2)$-type error estimators

Let $\mathcal{H}$ be a Hilbert space with corresponding norm $|||\cdot|||$. Furthermore, given some discrete subspace X_ℓ corresponding to a mesh $\mathcal{E}_\ell$ of Γ, let $U_\ell \in X_\ell$ denote the computed discrete solution approximating the unknown solution $u \in \mathcal{H}$. Let $\widehat{\mathcal{E}}_\ell$ denote the uniform refinement of $\mathcal{E}_\ell$ which is obtained by splitting all segments $E \in \mathcal{E}_\ell$ at their midpoint. The associated discrete space is denoted by $\widehat{X}_\ell$. Clearly, the Galerkin solution $\widehat{U}_\ell$ corresponding to the space $\widehat{X}_\ell$ is a better approximation to the unknown solution $u \in \mathcal{H}$. In [13], we first analyzed the simple $(h - h/2)$-type error estimator

$$\eta_\ell := |||\widehat{U}_\ell - U_\ell||| \tag{7}$$

in the context of BEM. It always provides a lower bound for the Galerkin error

$$\eta_\ell \leq |||u - U_\ell|||.$$

Moreover, an upper bound (up to some multiplicative constant indicated by the symbol $\lesssim$)

$$|||u - U_\ell||| \lesssim \eta_\ell$$

is equivalent to the so-called saturation assumption

$$|||u - \widehat{U}_\ell||| \leq q\,|||u - U_\ell||| \tag{8}$$

with some uniform constant $q \in (0, 1)$. I.e., uniform mesh-refinement leads to a uniform improvement of the discretization error. From a heuristic point of view, the saturation assumption states that the BEM error exhibits asymptotics $\mathcal{O}(h^\alpha)$, see the discussion in [13].

We stress that this assumption is in some sense natural and can, for instance, be proven for the finite element method [9], see also [12, Section 2.3]. For the boundary element method, however, (8) still remains mathematically open although observed throughout academic experiments.

2.2 Formulation of adaptive algorithm

Due to the non-local nature of the integral operators involved, the energy norm $|||\cdot|||$ cannot be easily written as sum of local contributions, as is, e.g., the case for the L^2-norm $\|\cdot\|_{L^2}^2 = \sum_{E\in\mathcal{E}_\ell} \|\cdot\|_{L^2(E)}^2$. Based on localization techniques from [6, 7, 11, 13], we give suitable error estimators $\widetilde{\mu}_\ell^2 = \sum_{E\in\mathcal{E}_\ell} \widetilde{\mu}_\ell(E)^2$ which are equivalent to η_ℓ, but whose contributions $\widetilde{\mu}_\ell(E)$, at least heuristically, measure the local error. Moreover, based on recent results from [4, 2], a compound error estimator ρ_ℓ which additionally controls data oscillations, will be proposed for all problem settings discussed in this paper. Given an initial mesh $\mathcal{E}_0$ and an adaptivity parameter $0 < \theta < 1$, our adaptive algorithm then reads as follows:

Algorithm 1.

1. *compute discrete solution* $U_\ell \in X_\ell$
2. *compute error indicators* $\rho_\ell(E)$ *for all* $E \in \mathcal{E}_\ell$
3. *find (minimal) set* $\mathcal{M}_\ell$ *such that*

$$\theta \sum_{E\in\mathcal{E}_\ell} \rho_\ell(E)^2 \leq \sum_{E\in\mathcal{M}_\ell} \rho_\ell(E)^2 \tag{9}$$

4. *refine (at least) elements* $E \in \mathcal{M}_\ell$ *and obtain* $\mathcal{E}_{\ell+1}$
5. *update counter* $\ell \mapsto \ell + 1$ *and goto (1)*

2.3 Benchmarks

In order to empirically study the performance of our adaptive approach compared to a uniform mesh-refining strategy, we propose two benchmarks. The first example covers the Laplace equation, i.e. $f = 0$, and is constructed in such a way, that Dirichlet and Neumann data, both, have singularities. In particular, they lack smoothness properties sufficient to reveal optimal convergence with uniform mesh refinement. The second example covers the case of nontrivial volume forces $f \neq 0$ and is designed such, that our adaptive algorithm needs to resolve singularities of Cauchy and volume data.

Below, the results of all numerical experiments are visualized within three figures. Since we prescribe the analytical solution, a reliable error bound *err* can be computed and is shown for reference along with the error estimators and data oscillations. In the first figure, we plot all quantities over the number of boundary elements $\#\mathcal{E}_\ell$. We recall that the optimal rate of convergence of lowest-order BEM is $\mathcal{O}(\#\mathcal{E}_\ell^{-3/2})$, since $h_\ell \sim (\#\mathcal{E}_\ell)^{-1}$ for uniform meshes. The examples are chosen in such a way that uniform mesh-refinement can be predicted to yield a reduced order of convergence $\mathcal{O}(\#\mathcal{E}_\ell^{-\alpha})$ with $\alpha < 3/2$.

Second, we plot the quantities over the computational time. Since an adaptively generated solution depends on the entire history of solutions, whereas this is not the case for uniform meshes, the time consumption is measured differently for uniform and adaptive approach. We define the computational time as follows:

- For uniform mesh-refinement, $t_{\ell,\mathrm{unif}}$ is the time elapsed for ℓ uniform mesh-refinements of the initial mesh $\mathcal{E}_0$, the assembly of the Galerkin data, and the computation of the Galerkin solution.

For adaptive mesh-refinement, the computational time is defined cumulatively:

- We define $t_{-1,\mathrm{adap}} = 0$.
- For $\ell \geq 0$, we set $t_{\ell,\mathrm{adap}} = t_{\ell-1,\mathrm{adap}}$ plus the time elapsed for the assembly of the Galerkin data, the computation of the Galerkin solution, the computation of the error indicators, and the adaptive refinement of the mesh to obtain $\mathcal{E}_{\ell+1}$.

Finally, we plot the quantities over the memory consumption which is understood as follows:

- For uniform mesh-refinement, we count the memory which is occupied by the data structure for the mesh, the discrete integral operators, and the solution vector.
- For the adaptive version, we count the memory which is occupied by the data structure for the mesh, the integral operators, the solution vector, the error estimators, and the data oscillations which are needed for the adaptive mesh-refinement.

With these three different figures, we empirically evaluate the quality of a performed computation with respect to both, mathematical order of convergence and computational effectivity.

2.3.1 Geometry

We choose Ω to be a rotated L-shaped domain as shown in Figure 1. The initial boundary mesh $\mathcal{E}_0$ consist of $\#\mathcal{E}_0 = 8$ elements. In the case of non-vanishing volume forces, we discretize the given data $f \in L^2(\Omega)$. To that end, we use a regular triangulation $\mathcal{T}_\ell$ of Ω that satisfies $\mathcal{T}_\ell|_\Gamma = \mathcal{E}_\ell$. The initial triangulation $\mathcal{T}_0$ with $\#\mathcal{T}_0 = 6$ elements is also plotted in Figure 1.

2.3.2 Example 1

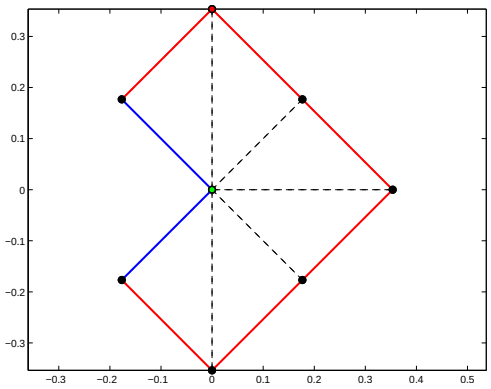
For given $\delta > 0$, we first define a function in polar coordinates $v_\delta(r, \varphi) = r^\delta \cos(\delta\varphi)$. The analytical solution u of the Laplace equation

$$\Delta u = 0 \quad \text{in} \quad \Omega$$

is prescribed by

$$u(x) = v_{2/3}(x) + v_{7/8}(x - z),$$

where z denotes the uppermost corner of Γ. This choice of u effects that the Dirichlet data g have a weak singularity at the uppermost corner, whereas the Neumann data ϕ have a generic singularity at the reentrant corner.



coordinates.dat		elements.dat	
0.3536	0	1	2
0.1768	0.1768	2	3
0	0.3536	3	4
-0.1768	0.1768	4	5
0	0	5	6
-0.1768	-0.1768	6	7
0	-0.3536	7	8
0.1768	-0.1768	8	1

dirichlet.dat		Neumann.dat	
4	5	6	7
5	6	7	8
		8	1
		1	2
		2	3
		3	4

Figure 1: Rotated L-shaped domain Ω. The initial boundary mesh $\mathcal{E}_0$ consists of 8 boundary elements. In case of non-trivial volume forces, the initial triangulation $\mathcal{T}_0$ (dashed) of Ω consists of 6 triangles. In Example 1, the solution is prescribed such that the Neumann data $\phi = \partial_n u$ has a generic singularity at the reentrant corner, the Dirichlet data $g = u|_\Gamma$ has a singularity at the uppermost corner. For the mixed boundary value problem covered by Example 2, the Dirichlet boundary consist of the two boundary elements that share the reentrant corner as common point. The Neumann boundary is the remaining part of Γ. The corresponding data structure is presented to the right of the picture.

2.3.3 Example 2

Let $z = (0.14, 0.14) \in \Omega$ denote a point inside the domain. We then prescribe the analytical solution

$$u(x) = |x - z|^{9/5} + v_{2/3}(x).$$

The volume force $f = -\Delta u$ has a weak singularity at the point z and the normal derivative of the Newtonian potential $\partial_n \widetilde{N} f$ as well as the Neumann data $\phi = \partial_n u$ have a generic singularity at the reentrant corner.

We aim at solving some mixed boundary value problem. To that end we split the boundary into Dirichlet and Neumann part as shown in Figure 1.

3 Implementation of mesh-refinement and integral operators

3.1 Discretization of the boundary

Throughout, $\Gamma = \partial\Omega$ is the piecewise affine boundary of a polygonal Lipschitz domain $\Omega \subset \mathbb{R}^2$. If necessary, Γ is partitioned into finitely many relatively open and disjoint boundary pieces, e.g. in a Dirichlet boundary Γ_D and a Neumann boundary Γ_N, i.e., $\Gamma = \overline{\Gamma}_D \cup \overline{\Gamma}_N$ and $\Gamma_D \cap \Gamma_N = \emptyset$.

A triangulation or mesh of Γ is a finite set $\mathcal{E}_\ell = \{E_1, \dots, E_N\}$ such that the elements $E_j \in \mathcal{E}_\ell$ are affine line segments and their intersection has vanishing measure, i.e., it holds that $E_j = [a_j, b_j] := \operatorname{conv}\{a_j, b_j\}$ as well as

$$|E_i \cap E_j| = 0 \text{ for } i \neq j \text{ and } \Gamma = \bigcup_{E_j \in \mathcal{E}_\ell} E_j. \tag{10}$$

Furthermore, if Γ is partitioned into Γ_D and Γ_N, one usually assumes that this partition is resolved by $\mathcal{E}_\ell$, i.e., $E_j \in \mathcal{E}_\ell$ satisfies either $E_j \subseteq \overline{\Gamma}_D$ or $E_j \subseteq \overline{\Gamma}_N$. With $\mathcal{K}_\ell = \{z_1, \dots, z_N\}$ the set of all nodes of the triangulation $\mathcal{E}_\ell$, it holds that $\#\mathcal{E}_\ell = \#\mathcal{K}_\ell$ for the closed boundary Γ.

3.1.1 Data structure and mesh-refinement

The set of nodes $\mathcal{K}_\ell = \{z_1, \dots, z_N\}$ of the triangulation $\mathcal{E}_\ell = \{E_1, \dots, E_N\}$ is represented by an $N \times 2$ array `coordinates`. The j-th row of `coordinates` stores the coordinates of the j-th node $z_j = (x_j, y_j) \in \mathbb{R}^2$ as

`coordinates(`j`,:)` $= [\ x_j \ \ y_j\ ]$.

If Γ is not split into several parts, the triangulation $\mathcal{E}_\ell$ is represented by an $N \times 2$ array `elements`. The i-th boundary element $E_i = [z_j, z_k]$ with nodes $z_j, z_k \in \mathcal{K}_\ell$ is stored as

`elements(`i`,:)` $= [\ j \ \ k\ ]$,

where the nodes are given in counterclockwise order, i.e., the parametrization of the boundary element $E_i \subset \Gamma$ is mathematically positive.

If Γ is split into Dirichlet boundary Γ_D and Neumann boundary Γ_N, the triangulation $\mathcal{E}_\ell$ is represented by an $N_D \times 2$ array `dirichlet` and an $N_N \times 2$ array `neumann` which describe the elements $E_j \subseteq \overline{\Gamma}_D$ and $E_k \subseteq \overline{\Gamma}_N$ as before. Then, `elements = [dirichlet;neumann]` with $N = N_D + N_N$.

For boundary element meshes, `HILBERT` provides an efficient local mesh-refinement algorithm implemented in the function `refineBoundaryMesh`. Let `marked` be an $(M \times 1)$-column vector containing the indices of marked elements, i.e. the set $\mathcal{M}_\ell$ in step (3) of Algorithm 1. Then,

```
[coordinates_fine,elements_fine,father2son] ...
 = refineBoundaryMesh(coordinates,elements,marked)
```

provides a mesh $\mathcal{E}_{\ell+1}$ which is only refined locally in the sense that all elements of $\mathcal{M}_\ell$ are refined. A marked element $E_i \in \mathcal{E}_\ell$ is bisected to certain sons $e_j, e_k \in \mathcal{E}_{\ell+1}$. The $(N \times 2)$-matrix `father2son` provides a link between the element indices in the sense that

`father2son`$(i,:) =$ `[` j, k `]` for $E_i = e_j \cup e_k$.

If an element $E_i \in \mathcal{E}_\ell$ is not refined, one has $E_i = e_j \in \mathcal{E}_{\ell+1}$, where the link between these indices is given by

`father2son`$(i,:) =$ `[` j, j `]` for $E_i = e_j$.

If the optional parameter `marked` is ommited, the uniformly refined mesh $\widehat{\mathcal{E}}_\ell = \{e_1, \dots, e_{2N}\}$ is returned. If Γ is split into a Dirichlet boundary Γ_D and a Neumann boundary Γ_N, the function call

```
[coordinates_fine,dirichlet_fine,neumann_fine, ...
dir2son,neu2son] ...
  = refineBoundaryMesh(coordinates,dirichlet, ...
        neumann,marked_dirichlet,marked_neumann)
```

provides an accordingly refined mesh. The parameters `marked_dirichlet` and `marked_neumann` are optional and may again be ommited to obtain the uniform refinement.

3.1.2 Boundedness of K-mesh constant

Many estimates in numerical analysis depend on local quantities of the mesh, e.g., on an upper bound of the K-mesh constant

$$\kappa(\mathcal{E}_\ell) := \sup\{\text{length}(E_j)/\text{length}(E_k) \,|\, E_j, E_k \in \mathcal{E}_\ell \text{ with } E_j \cap E_k \neq \emptyset\} \geq 1 \tag{11}$$

which is the maximal ratio of the element widths of neighboring elements. To avoid the blow-up of the K-mesh constant, the mesh-refinement algorithm implemented in `refineBoundaryMesh` guarantees

$$\sup_{\ell \in \mathbb{N}} \kappa(\mathcal{E}_\ell) \leq 2\, \kappa(\mathcal{E}_0) \tag{12}$$

by refinement of all elements in a certain superset $\mathcal{R}_\ell \supseteq \mathcal{M}_\ell$.

As is proven in [4, Theorem 2.5], our mesh-refinement algorithm ensures that the number of refined elements does not exceed the number of marked elements arbitrarily. For high-quality initial meshes with $\kappa(\mathcal{E}_0) < 2$, our refinement rule guarantees

$$\#\mathcal{E}_\ell - \#\mathcal{E}_0 \lesssim \sum_{j=0}^{\ell-1} \#\mathcal{M}_j, \tag{13}$$

i.e. the set $\mathcal{R}_j$ is generically of the same size as $\mathcal{M}_j$. The constant hidden in the symbol $\lesssim$ only depends on the initial mesh $\mathcal{E}_0$.

3.2 Integral operators

3.2.1 Discrete function spaces

Let $\mathcal{P}^p(\mathcal{E}_\ell)$ be the space of all $\mathcal{E}_\ell$-piecewise polynomials of degree $p \in \mathbb{N}_0$ with respect to the arc-length. Note that functions $f_\ell \in \mathcal{P}^p(\mathcal{E}_\ell)$ are, in general, not continuous, but have jumps at the nodes of $\mathcal{E}_\ell$. In particular, $\mathcal{P}^0(\mathcal{E}_\ell)$ denotes the space of all $\mathcal{E}_\ell$-piecewise constant functions. If $\chi_j \in \mathcal{P}^0(\mathcal{E}_\ell)$ denotes the characteristic function of $E_j \in \mathcal{E}_\ell$, the set $\{\chi_1, \dots, \chi_N\}$ is the basis of $\mathcal{P}^0(\mathcal{E}_\ell)$ which is used throughout our implementation.

One particular example for a function in $\mathcal{P}^0(\mathcal{E}_\ell)$ is the local mesh-width $h_\ell \in \mathcal{P}^0(\mathcal{E}_\ell)$ which is defined $\mathcal{E}_\ell$-elementwise by

$$h_\ell|_E := \text{length}(E) \quad \text{for all } E \in \mathcal{E}_\ell. \tag{14}$$

Let $\mathcal{S}^1(\mathcal{E}_\ell) := \mathcal{P}^1(\mathcal{E}_\ell) \cap C(\Gamma)$ denote the set of all continuous and (with respect to the arc length) $\mathcal{E}_\ell$-piecewise affine functions. For each node $z_j \in \mathcal{K}_\ell$ of $\mathcal{E}_\ell$, let $\zeta_j \in \mathcal{S}^1(\mathcal{E}_\ell)$ be the hat function associated with the node $z_j \in \mathcal{K}_\ell$, i.e., $\zeta_j(z_k) = \delta_{jk}$. Then, the set $\{\zeta_1, \dots, \zeta_N\}$ is a basis of $\mathcal{S}^1(\mathcal{E}_\ell)$, which is used throughout our implementation.

In the following, we only consider the lowest-order BEM, where the spaces $\mathcal{P}^0(\mathcal{E}_\ell)$ and $\mathcal{S}^1(\mathcal{E}_\ell)$ are used to discretize fluxes and stresses.

3.2.2 Discrete integral operators

The Calderón-System (6) essentially involves the boundary integral operators V, K, W, and N_0. The remaining operators K' and N_1 can then be expressed in terms of the first four. Following our lowest-order ansatz, `HILBERT` provides a C-implementation of integral operators for discrete fluxes and stresses:

- the simple-layer potential matrix $\mathbf{V} : \mathcal{P}^0(\mathcal{E}_\ell) \to \mathcal{P}^0(\mathcal{E}_\ell)$ given by

$$\mathbf{V}_{ij} = -\frac{1}{2\pi} \int_{E_j} \int_{E_i} \log|x-y| \, d\Gamma(y) \, d\Gamma(x),$$

- the double-layer potential matrix $\mathbf{K} : \mathcal{S}^1(\mathcal{E}_\ell) \to \mathcal{P}^0(\mathcal{E}_\ell)$ given by

$$\mathbf{K}_{ij} = -\frac{1}{2\pi} \int_{E_i} \int_{\text{supp}(\zeta_j)} \zeta_j(y) \frac{(y-x)\cdot n_y}{|x-y|^2} \, d\Gamma(y) \, d\Gamma(x),$$

- the hypersingular integral operator matrix $\mathbf{W} : \mathcal{S}^1(\mathcal{E}_\ell) \to \mathcal{S}^1(\mathcal{E}_\ell)$ given by

$$\mathbf{W}_{ij} = \frac{1}{2\pi} \partial n_x \int_{\text{supp}(\zeta_j)} \int_{\text{supp}(\zeta_i)} \zeta_j(y) \frac{(y-x)\cdot n_y}{|x-y|^2} \zeta_i(x) \, d\Gamma(y) \, d\Gamma(x),$$

- Given a regular triangulation $\mathcal{T}_\ell$ of the domain Ω, the trace of the Newtonian potential $\mathbf{N} : \mathcal{P}^0(\mathcal{T}_\ell) \to \mathcal{P}^0(\mathcal{E}_\ell)$ reads

$$\mathbf{N}_{ij} = -\frac{1}{2\pi} \int_{E_i} \int_{T_j} \log|x-y| \, dy \, d\Gamma(x).$$

Antiderivatives for the integrals of $\mathbf{V}$ and $\mathbf{K}$ are taken from [15]. With similar techniques, we developed antiderivatives for the computation $\mathbf{N}$. The computation of $\mathbf{W}$ is implemented by use of Nédélec's formula

$$\langle Wu, v\rangle = \langle Vu', v'\rangle,$$

where $(\cdot)'$ denotes the arc-length derivative. However, for adaptive meshes, the analytical computation leads to instabilities due to cancellation effects. Given some fixed parameter $\eta > 0$, we call two boundary elements E_i, E_j *admissible* if they satisfy

$$\min\{\text{length}(E_i), \text{length}(E_j)\} \le \eta \, \text{distance}(E_i, E_j).$$

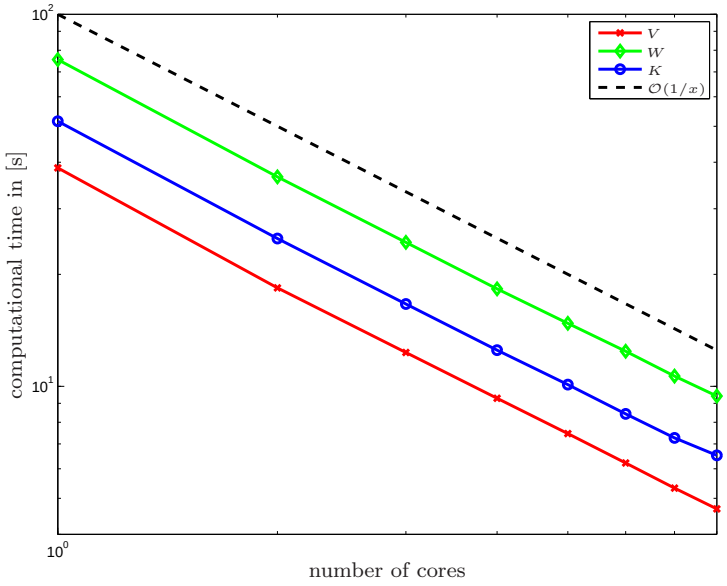


Figure 2: Speed up of operators using parallelization. The computation time is plotted in a double logarithmic scale over the number of cores. Computations were performed for a uniform mesh with 4096 number of boundary elements. For reference, the function $100/x$ with x representing the number of cores is plotted.

In this case, the outer integration is replaced by Gaussian quadrature. Moreover, in case of $\operatorname{diam}(E_j) < \operatorname{diam}(E_i)$, the order of integration is swapped ensuring that the outer integration is performed over the smaller boundary part. For admissible elements E_i, E_j, there holds exponential convergence of the semi-analytic quadrature used with respect to the order of the Gaussian quadrature rule chosen, see [16] or [14] in the context of hierarchical matrices.

In `HILBERT`, the discrete integral operators are provided by the following MEX-functions:

- `V = buildV(coordinates,elements[,eta])`
- `K = buildK(coordinates,elements[,eta])`
- `W = bulldW(coordinates,elements[,eta])`
- `N = buildN(coordinates,elements,vertices,triangles[,eta])`

The optional parameter `eta` may be omitted, a sane default value is used in that case. The assembly of the discrete boundary integral operators is significantly time consuming. Therefore, `HILBERT` uses simple parallelization paradigms to increase efficiency on multi-core systems. In our experiments, runtime scales properly with number of cores, see Figure 2.

4 The Dirichlet problem

4.1 Continuous model problem

In this section, we discuss the Dirichlet problem with vanishing volume forces

$$-\Delta u = 0 \text{ in } \Omega \quad \text{with} \quad u = g \text{ on } \Gamma. \tag{15}$$

With the simple-layer potential

$$V\phi(x) = -\frac{1}{2\pi} \int_\Gamma \phi(y) \log(|x-y|)\, d\Gamma(y) \quad \text{for } x \in \Gamma, \tag{16}$$

and the double-layer potential

$$Kg(x) = -\frac{1}{2\pi} \int_\Gamma g(y) \frac{(y-x)\cdot n_y}{|y-x|^2}\, d\Gamma(y) \quad \text{for } x \in \Gamma, \tag{17}$$

the first equation of the Calderón system (6) yields *Symm's integral equation*

$$V\phi = (K + 1/2)g \quad \text{on } \Gamma, \tag{18}$$

which is an equivalent formulation of (15). The exact solution $\phi \in H^{-1/2}(\Gamma)$ of (18) is the normal derivative $\phi = \partial_n u$ of the solution $u \in H^1(\Omega)$ of (15).

Formally, the Dirichlet data satisfy $g \in H^{1/2}(\Gamma)$. However, we assume additional regularity $g \in H^1(\Gamma) \subset H^{1/2}(\Gamma)$ so that g is, in particular, continuous.

Note that (18) can equivalently be written in variational form

$$\langle V\phi\,,\, \psi\rangle_\Gamma = \langle (K+1/2)g\,,\, \psi\rangle_\Gamma \quad \text{for all } \psi \in H^{-1/2}(\Gamma), \tag{19}$$

where $\langle\cdot\,,\,\cdot\rangle_\Gamma$ denotes the extended $L^2(\Gamma)$-scalar product $\langle\phi\,,\,\psi\rangle_\Gamma = \int_\Gamma \phi\psi\, d\Gamma$ for $\phi, \psi \in L^2(\Gamma)$. Provided that $\operatorname{diam}(\Omega) < 1$, one can show that the left-hand side

$$\langle\!\langle \phi\,,\psi\rangle\!\rangle_V := \langle V\phi\,,\,\psi\rangle_\Gamma \quad \text{for } \phi, \psi \in H^{-1/2}(\Gamma) \tag{20}$$

of (19) defines a scalar product and that the induced norm $|\!|\!|\phi|\!|\!|_V := \langle\!\langle \phi\,,\phi\rangle\!\rangle_V^{1/2}$ is an equivalent norm on $H^{-1/2}(\Gamma)$. In particular, the variational form (19) has a unique solution $\phi \in H^{-1/2}(\Gamma)$ which depends continuously on the data g with respect to the $H^{1/2}(\Gamma)$-norm.

4.2 Galerkin discretization

To discretize (19), we first replace the continuous Dirichlet data $g \in H^1(\Gamma)$ by its nodal interpolant

$$G_\ell := \sum_{j=1}^N g(z_j)\zeta_j \in \mathcal{S}^1(\mathcal{E}_\ell) \subset H^1(\Gamma). \tag{21}$$

Second, we replace the function space $H^{-1/2}(\Gamma)$ in (19) by the finite-dimensional space $\mathcal{P}^0(\mathcal{E}_\ell)$. Since the discrete space $\mathcal{P}^0(\mathcal{E}_\ell)$ is a subspace of $H^{-1/2}(\Gamma)$, $\langle\!\langle\cdot\,,\cdot\rangle\!\rangle_V$ from (20) is also a scalar product on $\mathcal{P}^0(\mathcal{E}_\ell)$. Consequently, there is a unique Galerkin solution $\Phi_\ell \in \mathcal{P}^0(\mathcal{E}_\ell)$ of

$$\langle V\Phi_\ell\,,\,\Psi_\ell\rangle_\Gamma = \langle (K+1/2)G_\ell\,,\,\Psi_\ell\rangle_\Gamma \quad \text{for all } \Psi_\ell \in \mathcal{P}^0(\mathcal{E}_\ell). \tag{22}$$

Let $\mathbf{x} \in \mathbb{R}^N$ denote the coefficient vector of the ansatz

$$\Phi_\ell = \sum_{j=1}^N \mathbf{x}_j \chi_j \tag{23}$$

and let $\mathbf{g} \in \mathbb{R}^N$ be defined by $\mathbf{g}_j := g(z_j)$ for all $z_j \in \mathcal{K}_\ell$. With the matrices $\mathbf{V}, \mathbf{K} \in \mathbb{R}^{N\times N}$ defined in Section 3.2.2 and the mass matrix $\mathbf{M} \in \mathbb{R}^{N\times N}$ defined by

$$\mathbf{M}_{kj} = \langle \zeta_j\,,\,\chi_k\rangle_\Gamma \quad \text{for all } j,k = 1,\dots,N,$$

the Galerkin formulation (22) is equivalently stated in terms of the linear system

$$\mathbf{V}\mathbf{x} = \mathbf{K}\mathbf{g} + \frac{1}{2}\mathbf{M}\mathbf{g}. \tag{24}$$

We stress that $\mathbf{V}$ is symmetric and positive definite since it stems from a scalar product.

4.3 Error indicators

Instead of discretizing the correct variational form (19), in fact, we solve

$$\langle V\phi_\ell\,,\,\psi\rangle_\Gamma = \langle (K+1/2)G_\ell\,,\,\psi\rangle_\Gamma \quad \text{for all } \psi\in H^{-1/2}(\Gamma) \tag{25}$$

with perturbed right-hand side, where we use the approximation $G_\ell \approx g$. For nodal interpolation, it is an analytical observation that the error between the exact solution $\phi \in H^{-1/2}(\Gamma)$ of (19) and the exact solution $\phi_\ell \in H^{-1/2}(\Gamma)$ of the perturbed formulation (25) is controlled by

$$|||\phi-\phi_\ell|||_V \lesssim \|h_\ell^{1/2}(g-G_\ell)'\|_{L^2(\Gamma)} =: \operatorname{osc}_{D,\ell}, \tag{26}$$

where $(\cdot)'$ denotes the arc-length derivative, cf. [4, Theorem 3.4].

With $\phi_\ell \in H^{-1/2}(\Gamma)$ the exact solution of (25), the $(h-h/2)$-error estimator η_ℓ from (7) now reads

$$\eta_\ell = |||\widehat{\Phi}_\ell - \Phi_\ell|||_V \simeq |||\phi_\ell - \Phi_\ell|||_V \tag{27}$$

and controls the discretization error. Altogether, (26)–(27) result in

$$|||\phi-\Phi_\ell|||_V \le |||\phi-\phi_\ell|||_V + |||\phi_\ell-\Phi_\ell|||_V \lesssim \operatorname{osc}_{D,\ell} + \eta_\ell \tag{28}$$

according to the triangle inequality.

As already mentioned, the non-locality of the integral operators leads to difficulties measuring the local contribution of a function to its norm. In [13, Theorem 3.4], we prove that

$$\eta_\ell \simeq \widetilde{\mu}_{D,\ell} := \|h_\ell^{1/2}(\widehat{\Phi}_\ell - \Pi_\ell\widehat{\Phi}_\ell)\|_{L^2(\Gamma)}, \tag{29}$$

where $\simeq$ indicates equivalence up to general multiplicative constants. Here, $\Pi_\ell : L^2(\Gamma) \to \mathcal{P}^0(\mathcal{E}_\ell)$ is the L^2-orthogonal projection onto the space of piecewise constants, which is just the piecewise integral mean

$$\Pi_\ell\psi|_{E_j} = \frac{1}{|E_j|}\int_{E_j}\psi\,d\Gamma.$$

Moreover, the estimator $\widetilde{\mu}_{D,\ell}$ is stated in a weighted L^2-norm and may thus be used to steer the local mesh-refinement.

If we plot η_ℓ and $\widetilde{\mu}_{D,\ell}$ over the number of elements, from the equivalence (29) of estimators, one can predict that the corresponding curves, for a sequence of arbitrarily refined meshes, are parallel, cf. [10, 13].

The equivalence (29) as well as the error control (28) lead to the choice of $\rho_\ell(E_j)^2 := \widetilde{\mu}_{D,\ell}(E_j)^2 + \operatorname{osc}_{D,\ell}(E_j)^2$ as error indicator to steer Algorithm 1. Based on results from [12, 3], one can prove that this choice of ρ_ℓ and the Dörfler marking criterion (9) guarantee $\lim_\ell \rho_\ell = 0$, see [4, Theorem 4.2]. Therefore, if the saturation assumption (8) holds (at least in infinitely many steps), we obtain convergence of Φ_ℓ to ϕ.

Remark. *We originally stated the saturation assumption* (8) *for the non-perturbed problem* (19)*. We stress that this already implies* (28)*, i.e., no additional assumption on the perturbed problem is posed, cf. [4, Theorem 3.4].* □ □

Listing 1: Implementation of Adaptive Algorithm

```
% adaptiveSymm provides the implementation of an adaptive
% algorithm for Symm's integral equation.

%*** maximal number of elements
nEmax = 100;

%*** adaptivity parameter
theta = 0.25;
percentage = 0.25;
```

```
10
11 %*** adaptive mesh-refining algorithm
12 while size(elements,1) < nEmax
13     %*** build uniformly refined mesh
14     [coordinates_fine,elements_fine,father2son] ...
15         = refineBoundaryMesh(coordinates,elements);
16     %*** discretize Dirichlet data and compute data oscillations
17     [uDh_fine,osc_fine] = discretizeDirichletData(coordinates_fine,...
18                                                   elements_fine,@g);
19     osc = osc_fine(father2son(:,1)) + osc_fine(father2son(:,2));
20     %*** compute fine-mesh solution
21     V_fine = buildV(coordinates_fine,elements_fine);
22     b_fine = buildSymmRHS(coordinates_fine,elements_fine,uDh_fine);
23     x_fine = V_fine\b_fine;
24     %*** compute (h-h/2)-error estimator tilde-mu
25     mu_tilde = computeEstSlpMuTilde(coordinates,elements,...
26                                     father2son,x_fine);
27     %*** mark elements for refinement
28     marked = markElements(theta,percentage,mu_tilde + osc);
29     %*** generate new mesh
30     [coordinates,elements] = refineBoundaryMesh(coordinates,...
31                                                 elements,marked);
32 end
```

4.4 Implementation of adaptive algorithm (Listing 1)

The given Dirichlet data g is provided by a `Matlab`-function `g.m`. Recall that we discretize $g \approx G_\ell$ in order to provide a black-box algorithm for the computation of the right-hand side $(K+1/2)G_\ell$.

The MATLAB script of Listing 1 realizes the adaptive algorithm.

- We use the adaptivity parameter $\theta = 1/4$ in (9) and mark at least the 25% of elements with the largest indicators (Line 8–9).
- Inside the adaptive loop, we first build a uniformly refined mesh for the computation of $\widehat{\Phi}_\ell$ (Line 14–15).
- Then, we discretize the given boundary data (Line 17–18).
- The function `buildSymmRHS` computes the right-hand side $(K+1/2)G_\ell$ in (24). Therefore, the lines 21–23 compute the coefficient vector corresponding to the fine mesh solution $\widehat{\Phi}_\ell$.
- The return values of the functions `computeEstSlpMuTilde` and `discretizeDirichletData` are the quadratic error estimator contributions $\widetilde{\mu}_{D,\ell}(E)^2$ and $\mathrm{osc}_{D,\ell}(E)^2$, respectively. The marking criterion (9) is provided by means of the function `markElements` (Line 28).

4.5 Numerical experiment

We perform Example 1 as benchmark and compare the results obtained by the proposed adaptive algorithm with those from a uniform approach. The Dirichlet data $g = u|_\Gamma$ is given, and the missing information $\phi = \partial_n u$ is computed by solving Symm's integral equation. Figure 3 shows error, error estimators, and data oscillations with respect to the number of boundary elements, the computational time, and the memory consumption. The singularities of the analytical solution lead to a reduced order of convergence $\mathcal{O}(\#\mathcal{E}_\ell^{-2/3})$ when using uniform meshes. The adaptive algorithm, however, resolves the singularities of the solution ϕ as well as of the given data g and reveals the optimal convergence behavior. The more interesting information for practitioners is effectivity with respect to computational resources. Figure 3 clearly shows that, in our experiment, the overhead introduced by the adaptive algorithm is soon overcome due to higher order of convergence.

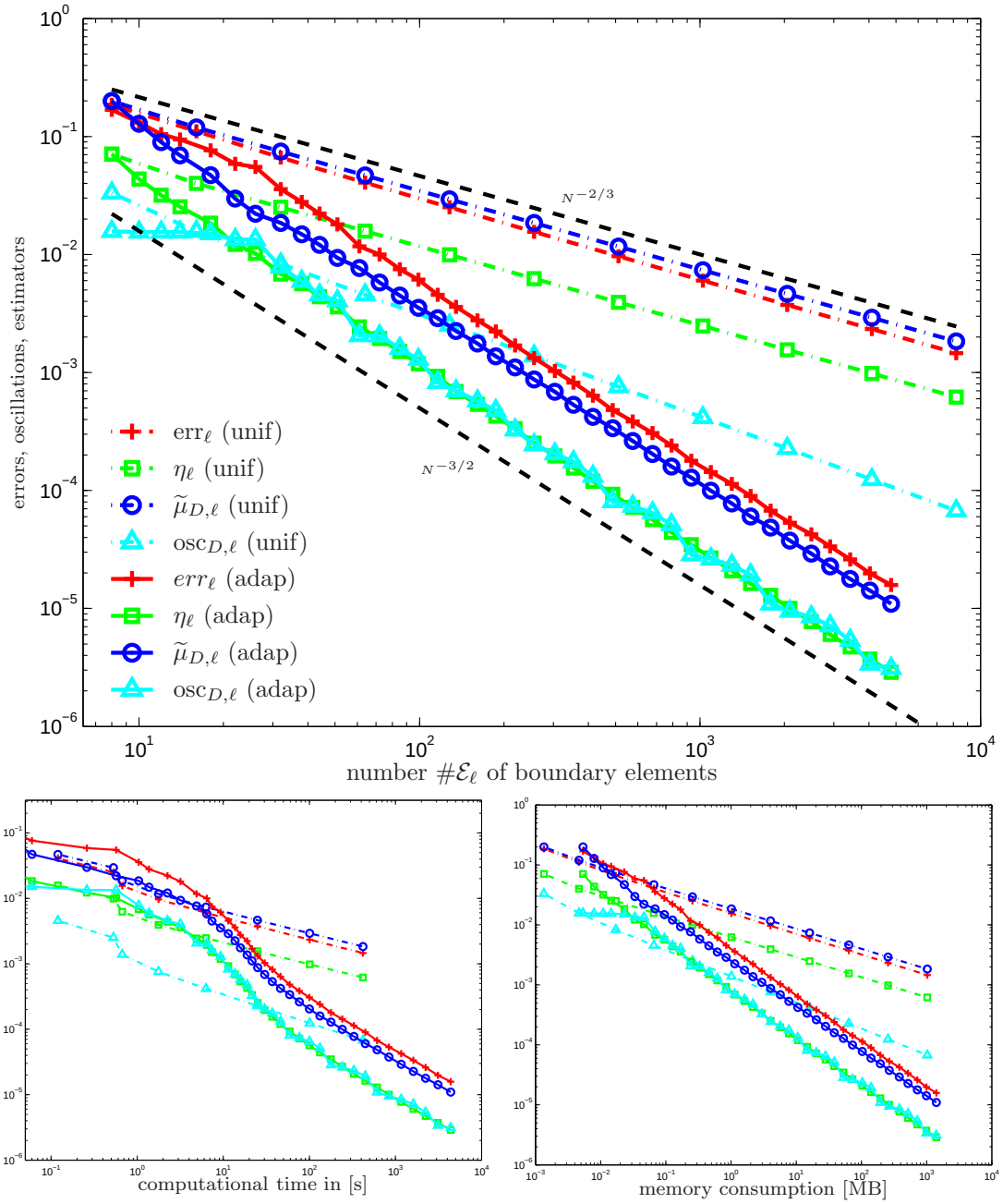


Figure 3: Example 1 computed with given Dirichlet and unknown Neumann data. For uniform mesh-refinement, the singularity of ϕ leads to a reduced order of convergence $\mathcal{O}(\#\mathcal{E}_\ell^{-2/3})$, whereas the adaptive strategy recovers the optimal order of convergence $\mathcal{O}(\#\mathcal{E}_\ell^{-3/2})$. Moreover, the adaptive scheme is also superior with respect to computational time and memory consumption.

5 The neumann problem

5.1 Continuous model problem

In this section, we discuss the Neumann problem with vanishing volume forces

$$-\Delta u = 0 \text{ in } \Omega \quad \text{with} \quad \partial_n u = \phi \text{ on } \Gamma. \tag{30}$$

We assume that Γ is connected, i.e. Ω das no holes. Note that due to the Gauss divergence theorem there holds

$$\int_\Gamma \phi \, d\Gamma = \int_{\partial\Omega} \partial_n u \, d\Gamma = \int_\Omega \Delta u \, dx = 0.$$

The second equation of the Calderón system (6) yields the *hypersingular integral equation*

$$Wg = (1/2 - K')\phi \quad \text{on } \Gamma, \tag{31}$$

with

$$Wg(x) = \frac{1}{2\pi} \partial_{n_x} \int_\Gamma \frac{(y-x)\cdot n_y}{|x-y|^2} g(y) \, dy \, dx$$

the hypersingular integral operator and K' the adjoint double layer potential defined by $\langle K\phi, \psi\rangle = \langle \phi, K'\psi\rangle$. We stress that (31) is an equivalent formulation of (30).

Formally, the Neumann data satisfy $\phi \in H^{-1/2}_*(\Gamma)$, where the subscript abbreviates the constraint $\langle \phi \, , \, 1\rangle_\Gamma = 0$. We will, however, assume additional regularity $\phi \in L^2(\Gamma)$ and piecewise continuity $\phi \in \mathcal{C}(E_j)$ for all $E_j \in \mathcal{E}_\ell$. The exact solution $g \in H^{1/2}(\Gamma)$ of the integral formulation (31) are the Dirichlet data $u|_\Gamma$ of the solution $u \in H^1(\Omega)$ of (30).

Due to $Wc = 0$ for all constant functions $c \in \mathbb{R}$, the solutions of (30) and (31) are only unique up to additive constants. To fix the additive constant, we consider the bilinear form

$$\langle\!\langle g \, , v \rangle\!\rangle_{W+S} := \langle Wg \, , \, v\rangle_\Gamma + \Big(\int_\Gamma g \, d\Gamma \Big) \Big(\int_\Gamma v \, d\Gamma \Big) \quad \text{for all } g, v \in H^{1/2}(\Gamma), \tag{32}$$

which leads to the following variational form: Find $g \in H^{1/2}(\Gamma)$ such that

$$\langle\!\langle g \, , v \rangle\!\rangle_{W+S} = \langle (1/2 - K')\phi \, , \, v\rangle_\Gamma \quad \text{for all } v \in H^{1/2}(\Gamma). \tag{33}$$

One can prove that $\langle\!\langle \cdot \, , \cdot \rangle\!\rangle_{W+S}$ from (32) defines a scalar product such that the induced norm $|\!|\!| \cdot |\!|\!|_{W+S}$ is an equivalent norm on $H^{1/2}(\Gamma)$. Consequently, (33) has a unique solution g which depends continuously on the Neumann data ϕ. Moreover, the solution automatically satisfies $g \in H^{1/2}_*(\Gamma)$.

5.2 Galerkin discretization

To discretize (33), we first replace the Neumann data $\phi \in L^2(\Gamma)$ by its L^2-projection $\Phi_\ell \in \mathcal{P}^0(\mathcal{E}_\ell)$,

$$\Phi_\ell|_{E_j} = \frac{1}{\text{length}(E_j)} \int_{E_j} \phi \, d\Gamma =: \mathbf{p}_j \quad \text{for all } E_j \in \mathcal{E}_\ell. \tag{34}$$

According to this definition,

$$\int_\Gamma \Phi_\ell \, d\Gamma = \sum_{E\in\mathcal{E}_\ell} \int_E \Phi_\ell \, d\Gamma = \sum_{E\in\mathcal{E}_\ell} \int_E \phi \, d\Gamma = \int_\Gamma \phi \, d\Gamma = 0,$$

i.e. $\Phi_\ell \in H^{-1/2}_*(\Gamma)$. Second, we replace the function space $H^{1/2}(\Gamma)$ in (33) by the finite-dimensional space $\mathcal{S}^1(\mathcal{E}_\ell)$. Since $\mathcal{S}^1(\mathcal{E}_\ell)$ is a subspace of $H^{1/2}(\Gamma)$, consequently, there exists a unique Galerkin solution $G_\ell \in \mathcal{S}^1(\mathcal{E}_\ell)$ of the discretized problem

$$\langle\!\langle G_\ell \, , V_\ell \rangle\!\rangle_{W+S} = \langle (1/2 - K')\Phi_\ell \, , \, V_\ell\rangle_\Gamma \quad \text{for all } V_\ell \in \mathcal{S}^1(\mathcal{E}_\ell). \tag{35}$$

As in the continuous case, the discrete solution G_ℓ automatically satisfies $\int_\Gamma G_\ell \, d\Gamma = 0$.

Recall the definition of $\mathbf{p} \in \mathbb{R}^N$ from (34) and let $\mathbf{x} \in \mathbb{R}^N$ denote the coefficient vector of the ansatz

$$G_\ell = \sum_{j=1}^N \mathbf{x}_j \zeta_j. \tag{36}$$

We recall the definition of the mass matrix $\mathbf{M} \in \mathbb{R}^{N\times N}$ from Section 4 as well as the definition of the matrices $\mathbf{K}, \mathbf{W} \in \mathbb{R}^{N\times N}$ from Section 3.2.2. With the rank-1 stabilization matrix $\mathbf{S} \in \mathbb{R}^{N\times N}$ defined by

$$\mathbf{S}_{kj} = \Big(\int_\Gamma \zeta_j \, d\Gamma\Big)\Big(\int_\Gamma \zeta_k \, d\Gamma\Big) \quad \text{for all } j,k = 1,\dots,N,$$

the Galerkin system (35) is equivalently stated in terms of the linear system

$$(\mathbf{W}+\mathbf{S})\mathbf{x} = \frac{1}{2}\mathbf{M}^T\mathbf{p} - \mathbf{K}^T\mathbf{p}. \tag{37}$$

Finally, we stress that the matrix $\mathbf{W}+\mathbf{S}$ from (37) is symmetric and positive definite since it stems from a scalar product.

5.3 Error indicators

Instead of solving the correct variational form (33), we solve

$$\langle\!\langle g_\ell \, , v\rangle\!\rangle_{W+S} = \langle (1/2 - K')\Phi_\ell \, , \, v\rangle_\Gamma \quad \text{for all } v \in H^{1/2}(\Gamma) \tag{38}$$

with perturbed right-hand side, where we use the approximation $\Phi_\ell \approx \phi$. Analytically, the error between the exact solution $g \in H^{1/2}(\Gamma)$ of (33) and the exact solution $g_\ell \in H^{1/2}(\Gamma)$ of the perturbed formulation (38) is controlled by the data oscillations

$$|\!|\!| g - g_\ell |\!|\!|_{W+S} \lesssim \|h_\ell^{1/2}(\phi - \Phi_\ell)\|_{L^2(\Gamma)} =: \operatorname{osc}_{N,\ell}, \tag{39}$$

see [2].

As for Symm's integral equation, the error estimator η_ℓ from (7) controls the discretization error by

$$|\!|\!| g_\ell - G_\ell |\!|\!|_{W+S} \approx |\!|\!| \widehat{G}_\ell - G_\ell |\!|\!|_{W+S} = |\!|\!| \widehat{G}_\ell - G_\ell |\!|\!|_{W} =: \eta_\ell. \tag{40}$$

This again leads to

$$|\!|\!| g - G_\ell |\!|\!|_{W+S} \le |\!|\!| g - g_\ell |\!|\!|_{W+S} + |\!|\!| g_\ell - G_\ell |\!|\!|_{W+S} \lesssim \operatorname{osc}_{N,\ell} + \eta_\ell \tag{41}$$

according to the triangle inequality and (39).

We stress that we are now dealing with different norms than in the case of the Dirichlet problem. In [11], it is proven that

$$\eta_\ell \sim \widetilde{\mu}_{N,\ell} := \|h_\ell^{1/2}(\widehat{G}_\ell - I_\ell \widehat{G}_\ell)'\|_{L^2(\Gamma)}. \tag{42}$$

The nodal interpolation operator $I_\ell : C(\Gamma) \to \mathcal{S}^1(\mathcal{E}_\ell)$ is defined as in (21).

We need to estimate both, the Galerkin error and the data-discretization error which leads to the choice of $\rho_\ell(E_j)^2 := \widetilde{\mu}_{N,\ell}(E_j)^2 + osc_{N,\ell}(E_j)^2$ as error indicator to steer Algorithm 1. As discussed in Section 4.4, and similar to [4, Theorem 4.2], we prove that our adaptive algorithm guarantees $\lim_\ell \rho_\ell = 0$ for the hypersingular integral equation. Therefore, if the saturation assumption holds (at least in infinitely many steps), we obtain convergence of G_ℓ to g.

Listing 2: Implementation of Adaptive Algorithm

```
% adaptiveHypsing provides the implementation of an adaptive
% algorithm for the hypersingular integral equation.

%*** maximal number of elements
nEmax = 100;

%*** adaptivity parameter
theta = 0.25;
percentage = 0.25;

%*** adaptive mesh–refining algorithm
```

```
12 while size(elements,1) < nEmax
13     %*** build uniformly refined mesh
14     [coordinates_fine,elements_fine,father2son] ...
15         = refineBoundaryMesh(coordinates,elements);
16     %*** discretize Neumann data and compute data oscillations
17     [phih_fine,osc_fine] ...
18         = discretizeNeumannData(coordinates_fine,elements_fine,@phi);
19     osc = osc_fine(father2son(:,1)) + osc_fine(father2son(:,2));
20     %*** compute fine-mesh solution
21     W_fine = buildW(coordinates_fine,elements_fine) ...
22              + buildHypsingStabilization(coordinates_fine,...
23                                          elements_fine);
24     b_fine = buildHypsingRHS(coordinates_fine,elements_fine,...
25                              phih_fine);
26     x_fine = W_fine\b_fine;
27     %*** compute (h-h/2)-error estimator tilde-mu
28     mu_tilde = computeEstHypMuTilde(elements_fine,elements,...
29                                     father2son,x_fine);
30     %*** mark elements for refinement
31     marked = markElements(theta,percentage,mu_tilde + osc);
32     %*** generate new mesh
33     [coordinates,elements] = refineBoundaryMesh(coordinates,...
34                                                 elements,marked);
35 end
```

5.4 Implementation of adaptive algorithm (Listing 2)

The Neumann data ϕ is provided by a `Matlab`-function `phi.m`. Recall that we discretize the given Neumann data in order to provide a black-box algorithm for the computation of right-hand side $(1/2 - K')\Phi_\ell$.

The MATLAB script of Listing 2 realizes the adaptive Algorithm1 by use of `HILBERT`.

- We use the adaptivity parameter $\theta = 1/4$ in (9) and mark at least the 25% of elements with the largest indicators (Line 8–9).
- Inside the adaptive loop, we first build a uniformly refined mesh for the computation of $\widehat{G}_\ell$ (Line 14–15).
- Then, we discretize the given boundary data (Line 17–18).
- To solve (37), we need to compute the stiffness matrix $\mathbf{W}$ as well as the stabilization matrix $\mathbf{S}$ (Line 21–23).
- The function `buildHypsingRHS` computes the right-hand side $(1/2 - K')\Phi_\ell$ in (37). Therefore, Lines 24–25 compute the coefficient vector corresponding to the fine mesh solution $\widehat{G}_\ell$.
- The reaturn values of the functions `computeEstHypMuTilde` and `discretizeNeumannData` are vectors of the quadratic error estimator contributions $\widetilde{\mu}_{N,\ell}(E)^2$ and $\mathrm{osc}_{N,\ell}(E)^2$, respectively. The marking criterion (9) is provided by means of the function `markElements` (Line 31).

5.5 Numerical experiment

We perform Example 1 as benchmark and compare the results obtained by the proposed adaptive algorithm with those from a uniform approach. The Neumann data $\phi = \partial_n u$ is given, and the missing information $g = u|_\Gamma$ is computed by solving the hypersingular integral equation. Figure 4 shows error, error estimators, and data oscillations with respect to the number of boundary elements, the computational time, and the memory consumption. The singularities of the Cauchy data lead to a reduced order of convergence $\mathcal{O}(\#\mathcal{E}_\ell^{-2/3})$ when using uniform meshes. Again, the adaptive algorithm reveals the optimal convergence behavior. Moreover, the adaptive approach in our experiment is clearly superior to a uniform strategy in the sense that available computational resources are used more efficiently.

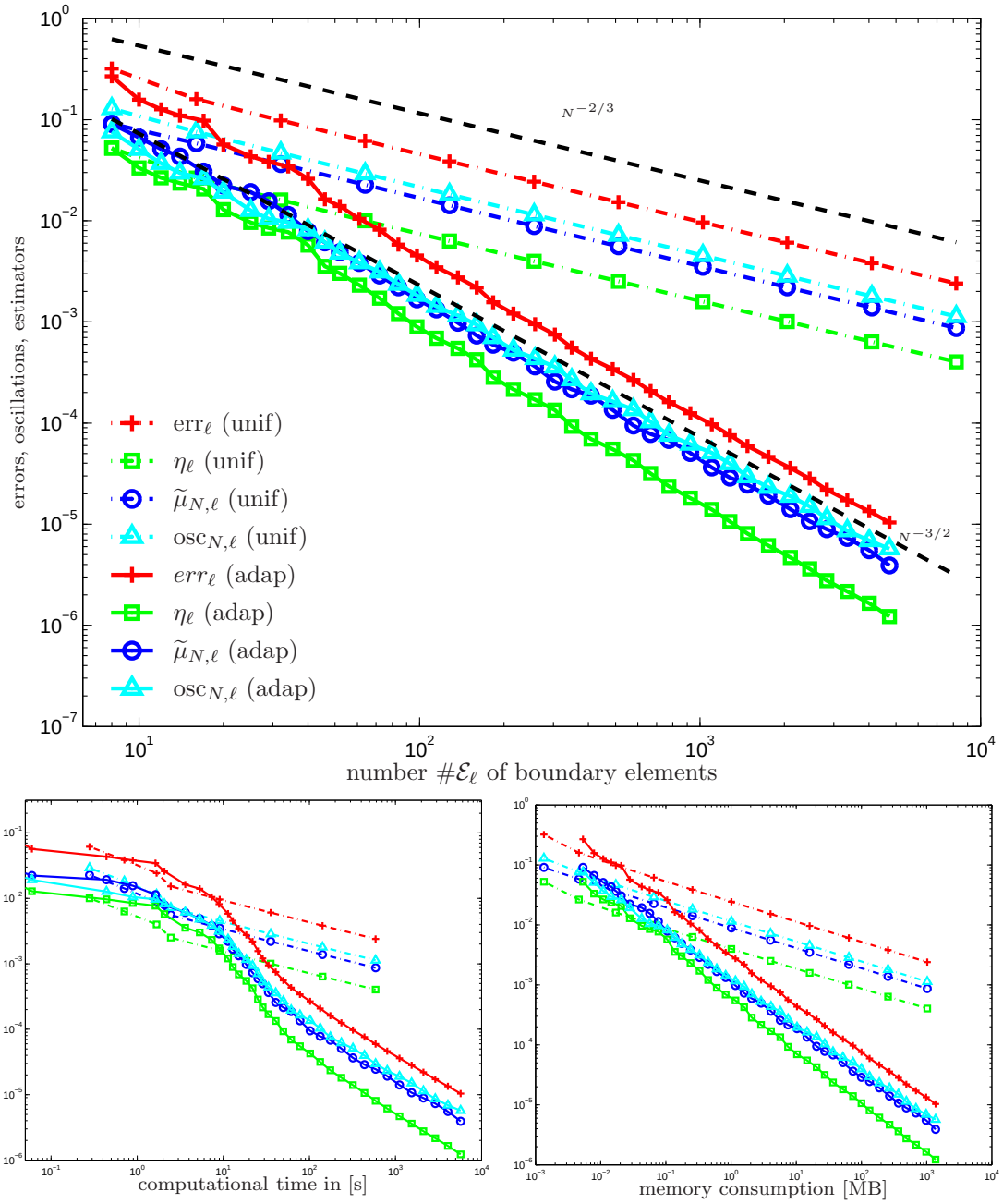


Figure 4: Example 1 computed with given Neumann and unknown Dirichlet data. For uniform mesh-refinement, the singularities of the Cauchy data lead to a reduced order of convergence $\mathcal{O}(\#\mathcal{E}_\ell^{-2/3})$, whereas the adaptive strategy recovers the optimal order of convergence $\mathcal{O}(\#\mathcal{E}_\ell^{-3/2})$. Moreover, the adaptive scheme is also superior with respect to computational time and memory consumption.

6 A mixed boundary value problem for the poisson equation

In this section, we use the techniques introduced in Section 4–5 to solve a mixed boundary value problem. Additionally, we consider non-vanishing volume forces:

$$\begin{aligned} -\Delta u &= f \quad \text{in } \Omega, \\ u &= u_D \quad \text{on } \Gamma_D, \\ u &= \phi_N \quad \text{on } \Gamma_N, \end{aligned} \tag{43}$$

where Γ_N is assumed to be connected. For the equivalent integral formulation of (43), we choose (and fix) arbitrary extensions $\overline{u}_D \in H^{1/2}(\Gamma)$ and $\overline{\phi}_N \in H^{-1/2}(\Gamma)$ of the given data from Γ_D and Γ_N, resp., to the entire boundary Γ. The missing boundary data, which have to be computed, are

$$u_N := u|_\Gamma - \overline{u}_D \quad \text{and} \quad \phi_D := \partial_n u - \overline{\phi}_N. \tag{44}$$

With $\Gamma_* \subseteq \Gamma$, we consider Sobolev spaces $H^{1/2}(\Gamma_*) = \{v|_{\Gamma_*} \,|\, v \in H^{1/2}(\Gamma)\}$ and its dual $\widetilde{H}^{-1/2}(\Gamma_*) := H^{1/2}(\Gamma_*)^*$. Moreover, let $\widetilde{H}^{1/2}(\Gamma_*) := \{v|_{\Gamma_*} \,|\, v \in H^{1/2}(\Gamma) \text{ and } v|_{\Gamma\setminus\Gamma_*} = 0\}$. With the so-called Calderón projector

$$A = \begin{pmatrix} -K & V \\ W & K' \end{pmatrix}, \tag{45}$$

the unknown data $u_N \in \widetilde{H}^{1/2}(\Gamma_N)$ and $\phi_D \in \widetilde{H}^{-1/2}(\Gamma_D)$ then satisfy the system of integral equations

$$A \begin{pmatrix} u_N \\ \phi_D \end{pmatrix} = (1/2 - A) \begin{pmatrix} \overline{u}_D \\ \overline{\phi}_N \end{pmatrix} - \begin{pmatrix} N_0 f \\ N_1 f \end{pmatrix} =: \mathcal{F} \quad \text{on } \Gamma_D \times \Gamma_N. \tag{46}$$

The definition

$$|\!|\!| (u_N, \phi_D) |\!|\!|_A^2 := \langle\!\langle (u_N, \phi_D), (u_N, \phi_D) \rangle\!\rangle_A = \langle W u_N \,,\, u_N \rangle_{\Gamma_N} + \langle V \phi_D \,,\, \phi_D \rangle_{\Gamma_D} \tag{47}$$

provides a norm on $\mathcal{H} = \widetilde{H}^{1/2}(\Gamma_N) \times \widetilde{H}^{-1/2}(\Gamma_D)$ which is equivalent to the usual product norm.

In order to provide black-box schemes to solve (46), we do not only discretize the given boundary data, but also the volume forces. To that end, let $\mathcal{T}_\ell$ denote a triangulation of the domain Ω with $\mathcal{T}_\ell|_\Gamma = \mathcal{E}_\ell$. We then replace $N_0 f$ by $N_0 F_\ell$ with $F_\ell = \pi_\ell f \in \mathcal{P}^0(\mathcal{T}_\ell)$ the L^2-orthogonal projection of f onto the space of piecewise constants over $\mathcal{T}_\ell$. To compute an approximation of $N_1 f$, we use the well-known identity

$$N_1 = (-1/2 + K')V^{-1} N_0,$$

see [17, 20, 19]. The discrete scheme now reads as follows: First, we solve Symm's integral equation

$$\langle V \Lambda_\ell \,,\, \Psi_\ell \rangle = \langle N_0 F_\ell \,,\, \Psi_\ell \rangle \quad \text{for all } \Psi_\ell \in \mathcal{P}^0(\mathcal{E}_\ell) \tag{48}$$

to obtain an approximation $\Lambda_\ell \approx V^{-1} N_0 f$. In the next step, we seek a solution $\mathbf{U}_\ell = (G_{N,\ell}, \Phi_{D,\ell}) \in \mathcal{S}^1(\mathcal{E}_\ell|_{\Gamma_N}) \times \mathcal{P}^0(\mathcal{E}_\ell|_{\Gamma_D})$ of the mixed problem

$$\langle\!\langle \mathbf{U}_\ell \,, \mathbf{V}_\ell \rangle\!\rangle_A = \langle \mathcal{F}_\ell \,,\, \mathbf{V}_\ell \rangle_{\mathcal{H}^\star \times \mathcal{H}} \text{ for all } \mathbf{V}_\ell \in \mathcal{S}^1(\mathcal{E}_\ell|_{\Gamma_N}) \times \mathcal{P}^0(\mathcal{E}_\ell|_{\Gamma_D}),$$

where

$$\mathcal{F}_\ell := (1/2 - A) \begin{pmatrix} I_\ell \overline{u}_D \\ \Pi_\ell \overline{\phi}_N \end{pmatrix} - \begin{pmatrix} N_0 F_\ell \\ (-1/2 + K')\Lambda_\ell. \end{pmatrix}$$

The algorithm for solving a mixed problem with volume force thus sketched as follows:

- Input: volume mesh $\mathcal{T}_\ell$, boundary mesh $\mathcal{E}_\ell = \mathcal{T}_\ell|_\Gamma$.
- Compute $\mathbf{g} = U_{D,\ell} = I_\ell \overline{u}_D$, $\mathbf{p} = \Phi_{N,\ell} = \Pi_\ell \overline{\phi}_N$ and $\mathbf{f} = F_\ell = \pi_\ell f$.
- Solve the linear system

$$\mathbf{V}\mathbf{y} = \mathbf{N}\mathbf{f}. \tag{49}$$

- Solve the linear system

$$\begin{pmatrix} \mathbf{W}|_{\Gamma_N \times \Gamma_N} & \mathbf{K}^T|_{\Gamma_N \times \Gamma_D} \\ -\mathbf{K}|_{\Gamma_D \times \Gamma_N} & \mathbf{V}|_{\Gamma_D \times \Gamma_D} \end{pmatrix} \mathbf{x} = \begin{pmatrix} \left(\frac{1}{2}\,\mathbf{p}^T\mathbf{M} - \mathbf{p}^T\mathbf{K} - \mathbf{g}^T\mathbf{W}\right)^T - (\frac{1}{2}\,\mathbf{M}^T - \mathbf{K}^T)\mathbf{y}|_{\Gamma_N} \\ \left(\frac{1}{2}\,\mathbf{M}\mathbf{g} + \mathbf{K}\mathbf{g} - \mathbf{V}\mathbf{p} - \mathbf{N}\mathbf{f}\right)|_{\Gamma_D} \end{pmatrix}. \tag{50}$$

For details, the reader is refered to [2]

6.1 A posteriori error estimate for mixed problem

Note that the energy norm $|||\cdot|||_A$ induced by the Calderón projector A can be written in terms of the energy norms $|||\cdot|||_{V(\Gamma_D)}$ and $|||\cdot|||_{W(\Gamma_N)}$ induced by the simple-layer potential $V \in L(\widetilde{H}^{-1/2}(\Gamma_D); H^{1/2}(\Gamma_D))$ and the hypersingular integral operator $W \in L(\widetilde{H}^{1/2}(\Gamma_N); H^{-1/2}(\Gamma_N))$. According to (47), there holds

$$|||(u_N, \phi_D)|||_A^2 = |||u_N|||_{W(\Gamma_N)}^2 + |||\phi_D|||_{V(\Gamma_D)}^2.$$

For a posteriori error estimation of the Galerkin error and the data oscillations on the boundary, we may therefore use the estimators introduced above for the Dirichlet and Neumann problem.

To include the discretization error introduced by solving (49), we use the local error estimator

$$\widetilde{\mu}_{V,\ell}^2 = \|h_\ell^{1/2}(1 - \Pi_\ell)\widehat{\Lambda}_\ell\|_{L^2(\Gamma)}^2,$$

where $\widehat{\Lambda}_\ell$ is the solution of (48) with respect to the fine mesh $\widehat{\mathcal{E}}_\ell$.

To steer an adaptive mesh-refining algorithm, it is therefore natural to use the combined error estimator

$$\rho_\ell^2 := \widetilde{\mu}_\ell^2 + \mathrm{osc}_\ell^2 = (\widetilde{\mu}_\ell^2 + \mathrm{osc}_\ell^2) + (\widetilde{\mu}_{V,\ell}^2 + \mathrm{osc}_{V,\ell}^2),$$

where $\mathrm{osc}_{V,\ell}^2 := \|h_\ell(F_\ell - f)\|_{L^2}^2$ denote the data oscillations of the volume forces. The estimator $\widetilde{\mu}_\ell(E)$ and $\mathrm{osc}_\ell(E)$ are either $\widetilde{\mu}_{D,\ell}(E), \mathrm{osc}_{D,\ell}(E)$ or $\widetilde{\mu}_{N,\ell}(E), \mathrm{osc}_{N,\ell}(E)$ in the case of $E \subseteq \Gamma_D$ or $E \subseteq \Gamma_N$, respectively. We refer to the `HILBERT` documentation [1] for details.

6.2 Numerical experiment

We compute the numerical solution of Example 2. The results of the numerical experiment are visualized in Figure 5. For the refinement of the volume mesh we use newest-vertex bisection where marked triangles are bisected, see e.g. [21, Chapter 5]. Further refinements are performed to ensure regularity of the volume mesh and the constraint $\mathcal{T}_\ell|_\Gamma = \mathcal{E}_\ell$. The last condition has turned out to increase stability of the integral operators as implemented in `HILBERT`.

In case of uniform mesh-refinements, we simply perform three bisections per triangle, i.e. all edges of $\mathcal{T}_\ell$ are halved.

Recall that the volume forces have a weak singularity at $(0.14, 0.14)$. However, the data oscillations of f seem to decay fast enough in the sense that the order of convergence with respect to the number of boundary elements is not limited by resolution of volume data in practice. However, strong generic singularities of ϕ and $\partial_n \widetilde{N} f$ at the reentrant corner limit the convergence behavior to approximately $\mathcal{O}(\#\mathcal{E}_\ell^{-4/7})$ in the case of uniform mesh-refinement.

Our adaptive algorithm now additionally measures the error due to the approximation of $\partial_n \widetilde{N} f$ and data oscillations of f over $\mathcal{T}_\ell$. It recovers the mathematically optimal order of convergence $\mathcal{O}(\#\mathcal{E}_\ell^{-3/2})$. Moreover, as in the other case studies, the accuracy of the computation is increased with respect to computational time. Since Example 2 includes the computation of the Newtonian potential $\widetilde{N} f|_\Gamma$, the figure showing the error and estimators with respect to memory consumption now also takes the storage requirements of $\mathbf{N}$ into account. It is remarkable that the adaptive approach is significantly superior to a uniform approach in this respect.

7 Conclusions and remarks

Features. The Matlab program package `HILBERT` provides stable implementations of the discrete integral operators corresponding to the 2D Laplacian as well as many other functions necessary for an easy accessible implementation of adaptive BEM. In this paper, only a short presentation of the library is possible. However, `HILBERT` is distributed with a full documentation [1] and many example benchmarks, some of which have gently been proposed at conferences by colleagues in order to stress-test our proposed adaptive approach under difficult conditions. Besides the features presented here, `HILBERT` comes along with many visualization tools, further error estimators, and different marking strategies.

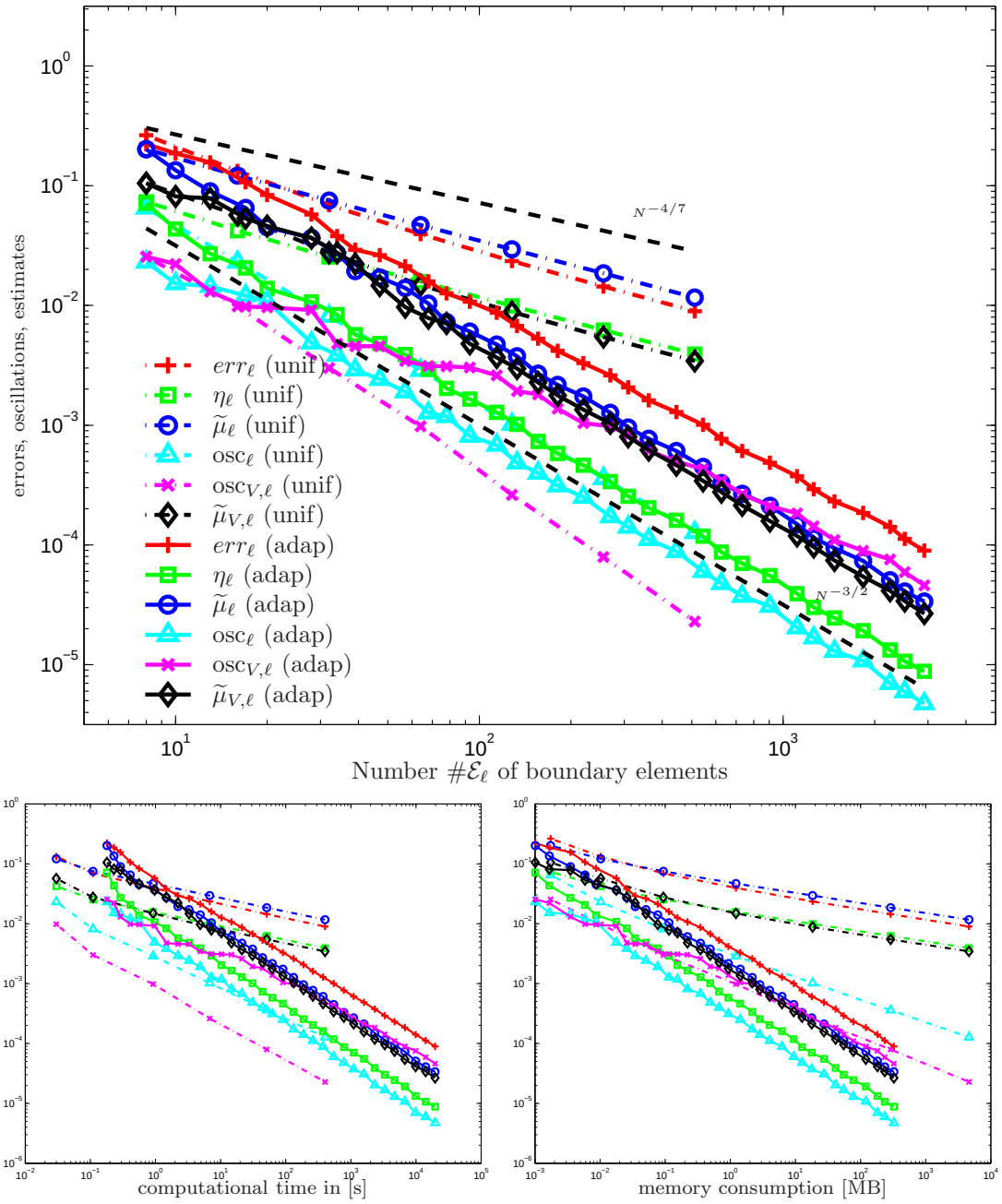


Figure 5: Example 2 computed with mixed boundary data. For uniform mesh-refinement, the singularities of ϕ and $\partial_n \widetilde{N} f$ lead to reduced order of convergence $\mathcal{O}(\#\mathcal{E}_\ell^{-4/7})$, whereas the adaptive strategy recovers the optimal order of convergence $\mathcal{O}(\#\mathcal{E}_\ell^{-3/2})$. Even though the volume data f has a weak singularity, a reduction of order of convergence due to bad resolution of f in the uniform case is not observed.

The Listings 1 and 2 included in Section 4 and 5 illustrate that adaptive BEM may be easily implemented by use of `HILBERT`. This makes the tool not only interesting for scientists, but also for lecturers planning classes on BEM or scientific computing. The package is under constant development, and updates are released continuously. In the near future, functionality for the numerical solution of transmission problems by use of adaptive FEM-BEM coupling as well as linear elasticity will be included (visit `http://www.asc.tuwien.ac.at/abem/hilbert` for detailed information on the ongoing development).

The recent progress in the analytical understanding of adaptivity in general and in the context of

BEM specifically, allows to implement mathematically justified adaptive algorithms, which automatically resolve the singularities of both, analytical solution and given data. We observe that our proposed algorithm — based on easy-to-implement $(h - h/2)$-type error estimators — empirically succeeds to recover the optimal order of convergence in all benchmarks performed so far. One of the aims of the authors is to emphasize the use of adaptivity and to make the concept more accessible to practitioners.

Restrictions. `HILBERT` is academic code in the sense that a `Matlab` implementation generically might be to slow for use in industrial applications. Moreover, `HILBERT` currently only provides implementations of the integral operators associated with the 2D Laplacian. It is restricted to lowest order elements and the canonical basis functions.

The most important restriction for practitioners might be that resolution of geometry is, so far, not included in the error estimation. In particular, all analytical results as well as the implementation demand Ω to be a polygonal domain, and boundary elements are chosen to be affine line segments.

Acknowledgements. Petra Goldenits acknowledges a grant of the TU graduate school *Partial Differential Equations in Technical Systems — Modeling, Simulation, and Control* funded by Vienna University of Technology. The research of the other authors is supported through the FWF project *Adaptive Boundary Element Method*, funded by the Austrian Science Fund (FWF) under grant P21732.

References

[1] M. Aurada, M. Ebner, M. Feischl, S. Ferraz-Leite, P. Goldenits, M. Karkulik, M. Mayr, D. Praetorius: *Hilbert (Release 2): A MATLAB implementation of adaptive BEM*, ASC Report **14/2010**, Institute for Analysis and Scientific Computing, Vienna University of Technology, 2010.

[2] M. Aurada, S. Ferraz-Leite, P. Goldenits, M. Karkulik, M. Mayr, D. Praetorius: *Convergence of adaptive BEM for some mixed boundary value problem*, Appl. Numer. Math., in print 2011.

[3] M. Aurada, S. Ferraz-Leite, D. Praetorius: *Estimator reduction and convergence of adaptive FEM and BEM*, ASC Report **27/2009**, Institute for Analysis and Scientific Computing, Vienna University of Technology, 2009.

[4] M. Aurada, P. Goldenits, D. Praetorius: *Convergence of data perturbed adaptive boundary element methods*, ASC Report **40/2009**, Institute for Analysis and Scientific Computing, Vienna University of Technology, 2009.

[5] J. Cascon, C. Kreuzer, R. Nochetto, K. Siebert: *Quasi-optimal convergence rate for an adaptive finite element method*, SIAM J. Numer. Anal. **46** (2008), 2524–2550.

[6] C. Carstensen, D. Praetorius: *Averaging techniques for the effective numerical solution of Symm's integral equation of the first kind*, SIAM J. Sci. Comput., **27** (2006), 1226–1260.

[7] C. Carstensen, D. Praetorius: *Averaging techniques for the a posteriori BEM error control for a hypersingular integral equation in two dimensions*, SIAM J. Sci. Comput., **29** (2007), 782–810.

[8] W. Doerfler: *A convergent adaptive algorithm for Possion's equation*, SIAM J. Numer. Anal. **33** (1996), 1106–1124.

[9] W. Doerfler, R. Nochetto: *Small data oscillation implies the saturation assumption*, Numer. Math. **91** (2002), 112.

[10] C. Erath, S. Ferraz-Leite, S. Funken, D. Praetorius: *Energy norm based a posteriori error estimation for boundary element methods in two dimensions*, Appl. Numer. Math., **59** (2009), 2713-2734.

[11] C. Erath, S. Funken, P. Goldenits, D. Praetorius: *Simple error estimators for the Galerkin BEM for some hypersingular integral equation in 2D*, ASC Report **20/2009**, Institute for Analysis and Scientific Computing, Vienna University of Technology, 2009.

[12] S. Ferraz-Leite, C. Ortner, D. Praetorius: *Convergence of simple adaptive Galerkin schemes based on $h - h/2$ error estimators*, Numer. Math., **116** (2010), 291-316.

[13] S. Ferraz-Leite, D. Praetorius: *Simple a posteriori error estimators for the h-version of the boundary element method*, Computing **83** (2008), 135–162.

[14] W. Hackbusch: *Hierarchische Matrizen. Algorithmen und Analysis.* Springer-Verlag, Berlin, 2009.

[15] M. Maischak: *The analytical computation of the Galerkin elements for the Laplace, Lamé and Helmholtz equation in 2D-BEM*, Preprint, Institute for Applied Mathematics, University of Hannover, 2001.

[16] M. Mayr: *Stabile Implementierung der Randelementmethode auf stark adaptierten Netzen*, Bachelor thesis (in German), Institute for Analysis and Scientific Computing, Vienna University of Technology, 2009.

[17] W. McLean: *Strongly elliptic systems and boundary integral equations*, Cambridge University Press, 2000.

[18] P. Mund, E. Stephan, J. Weisse: *Two-level methods for the single layer potential in $\mathbb{R}^3$*, Computing **60** (1998), 243–266.

[19] S. Sauter, C. Schwab: *Boundary element methods*, Springer Series in Computational Mathematics 39, Springer-Verlag, Berlin, 2011.

[20] O. Steinbach: *Numerical approximation methods for elliptic boundary value problems: finite and boundary elements*, Springer, New York, 2008.

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