TECHNISCHE
UNIVERSITÄT
WIEN
Vienna University of Technology

Bachelor Thesis

Different Strategies for the Adaptive FEM-BEM Coupling

written at the
Institute for Analysis and Scientific Computing
of the Vienna University of Technology

supervised by
Ao.Univ.Prof. Dipl.Math. Dr.techn. Dirk Praetorius

by
Michael Feischl
Schottenfeldgasse 92
1070 Wien

Contents

1 Model Problem

This section presents the interface problem which will be treated in the following sections. Let Ω^{int} be a bounded Lipschitz domain in $\mathbb{R}^2$ with boundary $\Gamma := \partial\Omega^{int}$ and an outward pointing normal vector n. The domain Ω^{ext} is the unbounded region of the problem, i.e $\Omega^{ext} := \mathbb{R}^2 \backslash \overline{\Omega^{int}}$. A schematically view of this setting is given in Figure 1. Consider the following problem

$$-\Delta u^i = f \qquad \text{in } \Omega^{int}, \tag{1a}$$
$$-\Delta u^e = 0 \qquad \text{in } \Omega^{ext}, \tag{1b}$$
$$\gamma u^i - \gamma^e u^e = u_0 \qquad \text{on } \Gamma, \tag{1c}$$
$$\partial_n u^i - \partial_n^e u^e = \phi_0 \qquad \text{on } \Gamma, \tag{1d}$$
$$u^e(x) = \mathcal{O}(|x|^{-1}) \qquad \text{as } |x| \to \infty, \tag{1e}$$

where γ, γ^e denote the interior and exterior trace operator and ∂_n, ∂_n^e the according normal derivatives. These operators are defined for $u \in H^1(\Omega^{int})$ and $u \in H^1_{\ell oc}(\Omega^{ext})$ in Section 2. The given data satisfy $f \in L^2(\Omega^{int})$, $u_0 \in H^{1/2}(\Gamma)$, and $\phi_0 \in H^{-1/2}(\Gamma)$. The sought solutions (u^i, u^e) solve (1) in the weak sense, as explained in Section 2, and satisfy $u^i \in H^1(\Omega^{int})$ as well as $u^e \in H^1_{\ell oc}(\Omega^{ext})$. To guarantee some properties of the integral operators in Section 3, we further assume $\operatorname{diam}(\Omega^{int}) < 1$.

As usual, the weak form of the interior problem is given by

$$a(u^i, v) - \langle \partial_n u^i \, , v \rangle_\Gamma = \langle f \, , v \rangle_{\Omega^{int}}, \quad \text{for } v \in H^1(\Omega^{int}), \tag{2}$$

where $a(u, v) := \int_{\Omega^{int}} \nabla u \cdot \nabla v \, \mathrm{dx}$ and $\langle \cdot, \cdot \rangle$ is the L^2-scalar product on the given space. The coupling of the interior and the exterior problem, i.e. the weak formulation of (1), is part of the next sections.

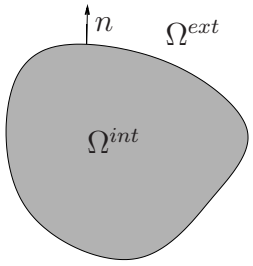


Figure 1: $\mathbb{R}^2$ split into Ω^{int}, Γ, and Ω^{ext}.

2 Sobolev Spaces

In this section, Ω is a bounded Lipschitz domain in $\mathbb{R}^2$ with boundary $\Gamma = \partial\Omega$. Note that these are the same assumptions as for Ω^{int} from Section 1. By Lipschitz domain,

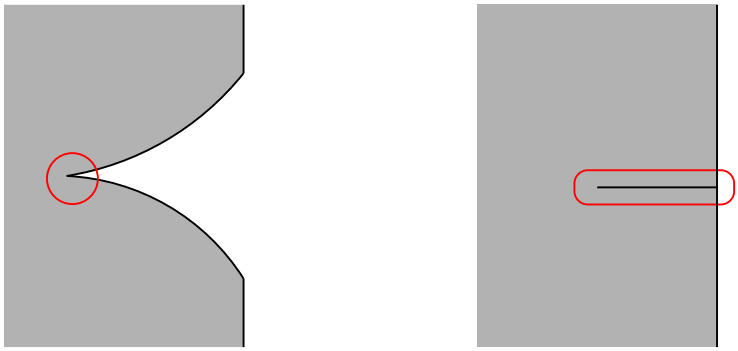


Figure 2: Non-Lipschitz domains in $\mathbb{R}^2$. The crucial points are marked red.

we denote a domain in euclidean space whose boundary can be locally represented by a Lipschitz continuous function. Exactly, Ω is called Lipschitz, if and only if, for every point $p \in \Gamma$ exists $r > 0$ and a function $h_p : B_r(p) \to B_1(0)$ with

- h_p is a bijection,
- both, h_p and h_p^{-1} are Lipschitz continuous,
- $h_p(B_r(p) \cap \Gamma) = \{(x, y) \in B_1(0) \,:\, y = 0\}$,
- $h_p(B_r(p) \cap \Omega) = \{(x, y) \in B_1(0) \,:\, y > 0\}$.

Here, $B_r(p)$ denotes the open ball with radius $r > 0$ and center $p \in \mathbb{R}^2$.

2.1 Sobolev Space $H^1(\Omega)$

We choose the classical definition of $H^1(\Omega)$ as space of all weakly differentiable functions $u \in L^2(\Omega)$ whose gradients belongs to $L^2(\Omega)^2$ as well. In more mathematically terms, we define

$$H^1(\Omega) := \{u \in L^2(\Omega) \,:\, \nabla u \text{ exists in the weak sense and } \nabla u \in L^2(\Omega)^2\}.$$

Associated with the norm

$$\|u\|_{H^1(\Omega)} = \left(\|u\|^2_{L^2(\Omega)} + \|\nabla u\|^2_{L^2(\Omega)}\right)^{1/2},$$

$H^1(\Omega)$ is a separable Hilbert space. Due to a well-known result of Meyers-Serrin, $C^\infty(\overline{\Omega})$ is a dense subspace of $H^1(\Omega)$.
Additionally, we define

$$H^1_{\ell oc}(\widetilde{\Omega}) := \{u : \widetilde{\Omega} \to \mathbb{R} : \text{ for any compact } K \subseteq \overline{\widetilde{\Omega}} \text{ holds } u|_K \in H^1(K)\},$$

where $\widetilde{\Omega}$ is an unbounded domain in $\mathbb{R}^2$. We associate $H^1_{\ell oc}(\widetilde{\Omega})$ with the initial topology τ induced by all seminorms of the form

$$p_K : H^1_{\ell oc}(\widetilde{\Omega}) \to \mathbb{R},\ u \mapsto \| u|_K \|_{H^1(K)},$$

where $K \subseteq \overline{\widetilde{\Omega}}$ is compact. It can be shown that $(H^1_{\ell oc}(\widetilde{\Omega}), \tau)$ is complete.

2.2 Sobolev Space $\widetilde{H}^{-1}(\Omega)$

For any $f \in L^2(\Omega)$

$$v \mapsto \langle f\,, v\rangle_\Omega$$

defines a functional in $H^1(\Omega)^*$. The following lemma shows that essentially all linear and continuous functionals on $H^1(\Omega)$ are of this type.

Lemma 2.1. *Let X and Y be real Hilbert spaces with continuous inclusion $\iota : X \to Y$. Then, the Riesz mapping $J_Y : Y \to Y^*$, $J_Y y := \langle y\,, .\rangle_Y$ is well-defined as operator $J_Y \in L(Y, X^*)$ and $J_Y(Y)$ is a dense subspace of X^*.*

We apply the lemma for $X = H^1(\Omega)$ and $Y = L^2(\Omega)$. We then obtain that $L^2(\Omega)$ is a dense subspace of $H^1(\Omega)^*$, and the duality brackets (f, v) coincide with the L^2-scalar product $\langle f\,, v\rangle_\Omega$ provided that $f \in L^2(\Omega)$. We denote by $\widetilde{H}^{-1}(\Omega)$ the dual space of $H^1(\Omega)$ with respect to the extended L^2-scalar product.

2.3 Sobolev Spaces $H^{1/2}(\Gamma)$ and $H^{-1/2}(\Gamma)$

The Sobolev Space $H^{1/2}(\Gamma)$ is defined by

$$H^{1/2}(\Gamma) := \{v \in L^2(\Gamma) : \|v\|_{H^{1/2}(\Gamma)} < \infty\},$$

and associated with the norm $\|v\|_{H^{1/2}(\Gamma)} = \left(\|v\|^2_{L^2(\Gamma)} + |v|^2_{H^{1/2}(\Gamma)}\right)^{1/2}$, where

$$|v|^2_{H^{1/2}(\Gamma)} = \int_\Gamma \int_\Gamma \frac{|v(x) - v(y)|^2}{|x - y|^2}\, \mathrm{d}\Gamma(y)\, \mathrm{d}\Gamma(x)$$

denotes the *Sobolev-Slobodeckij* seminorm. It can be shown that $H^{1/2}(\Gamma)$ is a Hilbert space. There holds $H^{1/2}(\Gamma) \subseteq L^2(\Gamma)$ with continuous inclusion. Therefore, Lemma 2.1 applies with $X = H^{1/2}(\Gamma)$ and $Y = L^2(\Gamma)$. We see that $L^2(\Gamma)$ is a dense subspace of $H^{1/2}(\Gamma)^*$. As above, we denote by $H^{-1/2}(\Gamma)$ the dual space of $H^{1/2}(\Gamma)$ with respect to the extended L^2-scalar product.

The following Theorem is given without proof, compare e.g. [10, Theorem 3.37].

Theorem 2.2. *There is a unique trace operator $\gamma \in L(H^1(\Omega), H^{1/2}(\Gamma))$ such that $\gamma u = u|_\Gamma$ for any $u \in C^\infty(\overline{\Omega})$.*

By use of the trace operator, we are able to characterize the functions $u \in H^1(\Omega)$ which vanish on the boundary

$$H_0^1(\Omega) := \{u \in H^1(\Omega) \,:\, \gamma u \equiv 0 \in H^{1/2}(\Gamma)\}.$$

To define the trace of functions $u \in H^1_{\ell oc}(\overline{\Omega}^c)$, we need the exterior trace operator γ^e.

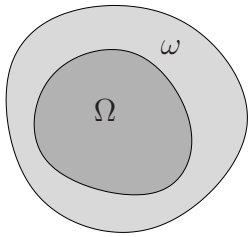


Figure 3: Schematically view of Ω and ω from Theorem 2.3

Theorem 2.3. *There is a continuous trace operator $\gamma^e : H^1_{\ell oc}(\overline{\Omega}^c) \to H^{1/2}(\Gamma)$ such that for any bounded, open Lipschitz domain $\widetilde{\Omega} \supseteq \overline{\Omega}$ with $\Gamma \cap \partial\widetilde{\Omega} = \emptyset$ holds*

$$\gamma^e u = (\gamma\, u|_{\widetilde{\Omega}\setminus\overline{\Omega}})\Big|_\Gamma \quad \text{for all } u \in H^1_{\ell oc}(\overline{\Omega}^c),$$

where γ is the trace operator from Theorem 2.2 with respect to $\widetilde{\Omega} \setminus \overline{\Omega}$.

Proof. Step 1: Let $\widetilde{\Omega} \subseteq \mathbb{R}^2$ be a bounded, open Lipschitz domain with $\widetilde{\Omega} \supseteq \Omega$ and $\Gamma \cap \partial\widetilde{\Omega} = \emptyset$. We define $\omega := \widetilde{\Omega} \setminus \overline{\Omega}$. Let $\eta \in C^\infty(\mathbb{R}^2)$ be a function which separates $\overline{\Omega}$ and $\widetilde{\Omega}^c$, i.e.

$$\eta(x) = 1 \text{ for } x \in \overline{\Omega},$$
$$\operatorname{supp}(\eta) \subseteq \widetilde{\Omega}.$$

$C^\infty(\overline{\omega})$ is a dense subspace in $H^1(\omega)$. Thus, for any $u \in H^1(\omega)$, we find a sequence $(u_n)_{n\in\mathbb{N}} \subseteq C^\infty(\overline{\omega})$ with $u_n \to u$ in $H^1(\omega)$. Note that ω is a bounded, open Lipschitz domain and if we consider γ with respect to ω, there holds

$$(\gamma u)|_\Gamma = \lim_{n\to\infty} (\gamma u_n)|_\Gamma = \lim_{n\to\infty} u_n|_\Gamma = \lim_{n\to\infty} (\eta u_n)|_\Gamma = (\gamma(\eta u))|_\Gamma\,.$$

We define the exterior trace operator γ^e as

$$\gamma^e : H^1_{\ell oc}(\overline{\Omega}^c) \to H^{1/2}(\Gamma),\ u \mapsto (\gamma(\,u|_\omega))|_\Gamma\,,$$

where γ is the trace operator with respect to ω. This operator is clearly linear. At first, we want to prove that this definition is independent from the choice of $\widetilde{\Omega}$. Assume that there are bounded, open Lipschitz domains $\widetilde{\Omega}_1$, $\widetilde{\Omega}_2$ with $\widetilde{\Omega}_i \supseteq \Omega$ and $\Gamma \cap \partial\widetilde{\Omega}_i = \emptyset$ for $i = 1, 2$. Let $\omega_i := \widetilde{\Omega}_i \setminus \overline{\Omega},\ i = 1, 2$ induce the interior trace operators γ_1, γ_2 from Theorem 2.2 and the exterior trace operators γ_1^e, γ_2^e, as stated above. Analogously, we choose η_i to

separate $\overline{\Omega}$ and $(\widetilde{\Omega}_i)^c$ $i = 1, 2$. For any $u \in H^1_{\ell oc}(\overline{\Omega}^c)$ and for any $\varepsilon > 0$, we find a function $u_\varepsilon \in C^\infty(\overline{\omega_1 \cup \omega_2})$ with $\|u - u_\varepsilon\|_{H^1(\omega_1 \cup \omega_2)} < \varepsilon$ and particularly $\|u - u_\varepsilon\|_{H^1(\omega_i)} < \varepsilon,\ i = 1, 2$. We obtain

$$\begin{aligned}
\|\gamma_1^e u - \gamma_2^e u\|_{H^{1/2}(\Gamma)} &\leq \|\gamma_1^e u - u_\varepsilon\|_{H^{1/2}(\Gamma)} + \|\gamma_2^e u - u_\varepsilon\|_{H^{1/2}(\Gamma)} \\
&\leq \|\gamma_1 \,(u - u_\varepsilon)|_{\omega_1}\,\|_{H^{1/2}(\Gamma)} + \|\gamma_2 \,(u - u_\varepsilon)|_{\omega_2}\,\|_{H^{1/2}(\Gamma)} \\
&= \|\gamma_1 \,(\eta_1(u - u_\varepsilon))|_{\omega_1}\,\|_{H^{1/2}(\partial\omega_1)} + \|\gamma_2 \,(\eta_2(u - u_\varepsilon))\,|_{\omega_2}\|_{H^{1/2}(\partial\omega_2)} \\
&\leq (C_1 + C_2)\max(\|\eta_1(u - u_\varepsilon)\|_{H^1(\omega_1)}\,,\, \|\eta_2(u - u_\varepsilon)\|_{H^1(\omega_2)}).
\end{aligned}$$

The constants C_i, $i = 1, 2$ depend on the continuity of γ_i. If we define $C_\eta := \max(\|\eta_1\|_\infty, \|\eta_2\|_\infty, \|\nabla\eta_1\|_\infty, \|\nabla\eta_2\|_\infty)$, we conclude

$$\begin{aligned}
\|\eta_i(u - u_\varepsilon)\|^2_{H^1(\omega_i)} &\leq C_\eta^2 \|u - u_\varepsilon\|^2_{L^2(\omega_i)} + \int_{\omega_i} (\nabla\eta_i(u - u_\varepsilon) + \eta_i\nabla(u - u_\varepsilon))^2 \,\mathrm{dx} \\
&\leq C_\eta^2 \|u - u_\varepsilon\|^2_{L^2(\omega_i)} + 2\int_{\omega_i} (\nabla\eta_i(u - u_\varepsilon))^2 + (\eta_i\nabla(u - u_\varepsilon))^2 \,\mathrm{dx} \\
&\leq 3C_\eta^2 \|u - u_\varepsilon\|^2_{H^1(\omega_i)}.
\end{aligned}$$

Finally,

$$\|\gamma_1^e u - \gamma_2^e u\|_{H^{1/2}(\Gamma)} \leq 3(C_1 + C_2)C_\eta\varepsilon.$$

Step 2: The topology of $H^1_{\ell oc}(\overline{\Omega}^c)$ is induced by the family of seminorms $p_K : H^1_{\ell oc}(\overline{\Omega}^c) \to \mathbb{R}$, $u \mapsto \|\, u|_K\,\|_{H^1(K)}$ for all compact $K \subseteq \Omega^c$. Choose $\widetilde{K} \supseteq \Omega$ compact with $\partial\widetilde{K} \cap \partial\Omega = \emptyset$ and $\partial\widetilde{K}$ Lipschitz. Set $K = \widetilde{K} \setminus \Omega$. Again, we define the function $\eta \in C^\infty(\mathbb{R}^2)$ to separate $\overline{\Omega}$ and $\widetilde{K}^c$, i.e.

$$\eta(x) = 1 \text{ for } x \in \overline{\Omega} \quad \text{and} \quad \eta(x) = 0 \quad \text{for } x \in \widetilde{K}^c.$$

By $\widetilde{\Gamma}$ we denote ∂K. For all $u \in H^1_{\ell oc}(\overline{\Omega}^c)$ holds

$$\|\gamma^e u\|_{H^{1/2}(\Gamma)} = \|\,(\gamma\, u|_K)|_\Gamma\,\|_{H^{1/2}(\Gamma)} = \|\gamma u\eta\|_{H^{1/2}(\widetilde{\Gamma})} \leq C\|u\eta\|_{H^1(K)}.$$

If the constant C_η denotes $\max(\|\eta\|_\infty, \||\nabla\eta|\|_\infty)$, we get

$$\|\gamma^e u\|_{H^{1/2}(\Gamma)} \leq 3C \cdot C_\eta \|u\|_{H^1(K)} = \tilde{C} p_K(u) \quad \text{for all } u \in H^1_{\ell oc}(\overline{\Omega}^c).$$

Let $\delta > 0$ and $O := p_K^{-1}([0, \delta/\tilde{C})) \subseteq H^1_{\ell oc}(\overline{\Omega}^c)$, then O is an open neighborhood of zero in $H^1_{\ell oc}(\overline{\Omega}^c)$ and for all $u, v \in H^1_{\ell oc}(\overline{\Omega}^c)$ with $u - v \in O$, we have

$$\|\gamma^e u - \gamma^e v\|_{H^{1/2}(\Gamma)} \leq \tilde{C} p_K(u - v) \leq \delta.$$

Thus, γ^e is continuous as a mapping from $H^1_{\ell oc}(\overline{\Omega}^c)$ to $H^{1/2}(\Gamma)$. □

2.4 Weak Form of the Laplace Equation

Assume $f \in \widetilde{H}^{-1}(\Omega)$. We say that $u \in H^1(\Omega)$ solves the equation

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

in the weak sense, provided there is a functional $g \in H^{-1/2}(\Gamma)$ such that

$$\langle f\,, v\rangle_\Omega = \langle \nabla u\,, \nabla v\rangle_\Omega - \langle g\,, \gamma v\rangle_\Gamma \quad \text{for all } v \in H^1(\Omega). \tag{3}$$

Due to the fact that for $u \in C^2(\Omega)$ and $f := -\Delta u$ holds $g = \partial_n u$, we denote by $\partial_n u$ the functional g from (3) for all $u \in H^1(\Omega)$ with $-\Delta u = f \in \widetilde{H}^{-1}(\Omega)$.
A function $u \in H^1_{\ell oc}(\overline{\Omega}^c)$ solves the equation

$$-\Delta u = 0 \quad \text{in } \mathbb{R}^2 \setminus \overline{\Omega}$$

in the weak sense, if there exists $h \in H^{-1/2}(\Gamma)$ such that

$$\langle \nabla u\,, \nabla v\rangle_{\mathbb{R}^2\setminus\overline{\Omega}} - \langle h\,, \gamma^e v\rangle_\Gamma = 0 \quad \text{for all } v \in \mathcal{D}(\mathbb{R}^2 \setminus \overline{\Omega}), \tag{4}$$

where $\mathcal{D}(\mathbb{R}^2 \setminus \overline{\Omega})$ denotes the space of all functions $u : \mathbb{R}^2 \setminus \overline{\Omega} \to \mathbb{R}$ with compact support and $u \in C^\infty(\mathrm{supp}(u))$. By $\partial_n^e u$ we denote the functional $-h$. One can think of ∂_n and ∂_n^e as

$$\partial_n u(x_0) = \lim_{x\in\Omega,\, x\to x_0} \nabla u(x)\cdot n(x_0),$$

and

$$\partial_n^e u(x_0) = \lim_{x\in\mathbb{R}^2\setminus\overline{\Omega},\, x\to x_0} \nabla u(x)\cdot n(x_0),$$

for sufficiently smooth functions u.

2.5 Equivalence of Norms

One important theorem which we will need throughout this text, is the following.

Theorem 2.4. *Let $r : H^1(\Omega) \to \mathbb{R}$ be a bounded seminorm. Suppose that $r(u) = 0$ implies $u = 0$ for any constant $u \in H^1(\Omega)$, then*

$$\|u\|_r := \Big(r(u)^2 + \|\nabla u\|^2_{L^2(\Omega)}\Big)^{1/2}$$

defines an equivalent norm on $H^1(\Omega)$.

For a proof, we refer to [15, Theorem 2.2].

3 Integral Operators

We define the *Newton kernel* in two dimensions

$$G(z) := -\frac{1}{2\pi} \log |z| \quad z \in \mathbb{R}^2 \setminus \{0\}.$$

In the distributional sense, this is the fundamental solution of the Laplace equation, i.e there holds

$$\int_\Omega \nabla G \cdot \nabla v \, \mathrm{dx} = v(0) \quad \text{for all } v \in \mathcal{D}(\Omega).$$

The *Newton kernel* is the kernel of the following three integral operators $\widetilde{N}$, $\widetilde{V}$, and $\widetilde{K}$, namely

- the *Newton potential* of $f : \Omega \to \mathbb{R}$

$$\widetilde{N} f(x) := \int_\Omega G(x-y) f(y) \, \mathrm{dy} \quad \text{for } x \in \Omega,$$

- the *simple-layer potential* of $\phi : \Gamma \to \mathbb{R}$

$$\widetilde{V} \phi(x) := \int_\Gamma G(x-y) \phi(y) \, \mathrm{d}\Gamma(y) \quad \text{for } x \in \Omega,$$

- the *double-layer potential* of $\phi : \Gamma \to \mathbb{R}$

$$\widetilde{K} \phi(x) := \int_\Gamma \partial_{n(y)} G(x-y) \phi(y) \, \mathrm{d}\Gamma(y) \quad \text{for } x \in \Omega.$$

Obviously, these operators are all linear. Below, we give some crucial properties.

Theorem 3.1.
(i) *$\widetilde{N}$ allows a unique extension $\widetilde{N} \in L(\widetilde{H}^{-1}(\Omega), H^1(\Omega))$ from $\mathcal{D}(\Omega)$ to $\widetilde{H}^{-1}(\Omega)$.*
(ii) *$-\Delta \widetilde{N} f = f$ for all $f \in \widetilde{H}^{-1}(\Omega)$.*
(iii) *$\gamma \widetilde{N} \in L(\widetilde{H}^{-1}(\Omega), H^{1/2}(\Gamma))$.*
(iv) *$\partial_n \widetilde{N} \in L(\widetilde{H}^{-1}(\Omega), H^{-1/2}(\Gamma))$.*

Theorem 3.2.
(i) *$\widetilde{V}$ allows a unique extension $\widetilde{V} \in L(H^{-1/2}(\Gamma), H^1(\Omega))$ from $L^2(\Gamma)$ to $H^{-1/2}(\Gamma)$.*
(ii) *$-\Delta \widetilde{V} \phi = 0 \in \widetilde{H}^{-1}(\Omega)$ for all $\phi \in H^{-1/2}(\Gamma)$.*
(iii) *$\gamma \widetilde{V} \in L(H^{-1/2}(\Gamma), H^{1/2}(\Gamma))$.*
(iv) *$\partial_n \widetilde{V} \in L(H^{-1/2}(\Gamma), H^{-1/2}(\Gamma))$.*

Theorem 3.3.
(i) *There holds $\widetilde{K} \in L(H^{1/2}(\Gamma), H^1(\Omega))$.*
(ii) *$-\Delta \widetilde{K} \phi = 0 \in \widetilde{H}^{-1}(\Omega)$ for all $\phi \in H^{-1/2}(\Gamma)$.*
(iii) *$\gamma \widetilde{K} \in L(H^{1/2}(\Gamma), H^{1/2}(\Gamma))$.*
(iv) *$\partial_n \widetilde{K} \in L(H^{1/2}(\Gamma), H^{-1/2}(\Gamma))$.*

We can analogously define $\widetilde{V}$ and $\widetilde{K}$ on $\mathbb{R}^2 \setminus \overline{\Omega}$

$$\widetilde{V}^e\phi(x) := \int_\Gamma G(x-y)\phi(y)\,\mathrm{d}\Gamma(y) \quad \text{for } x \in \mathbb{R}^2 \setminus \overline{\Omega},$$

$$\widetilde{K}^e\phi(x) := \int_\Gamma \partial_{n(y)} G(x-y)\phi(y)\,\mathrm{d}\Gamma(y) \quad \text{for } x \in \mathbb{R}^2 \setminus \overline{\Omega}.$$

According to elementary calculus, there holds $\widetilde{V}^e\phi$, $\widetilde{K}^e\phi \in C^\infty(\mathbb{R}^2 \setminus \overline{\Omega})$ and $-\Delta\widetilde{V}\phi = -\Delta\widetilde{K}\phi = 0$ in $\mathbb{R}^2 \setminus \Gamma$ for all $\phi \in L^1(\Gamma) \supseteq L^2(\Gamma)$. Obviously, there holds $H^{1/2}(\Gamma) \subseteq L^2(\Gamma)$. Thus, we obtain $-\Delta\widetilde{K}\psi = 0$ in $\mathbb{R}^2 \setminus \Gamma$ for all $\psi \in H^{1/2}(\Gamma)$.

Lemma 3.4. *$\widetilde{V}^e$ allows an unique extension to $\widetilde{V}^e : H^{-1/2}(\Gamma) \to H^1_{\ell oc}(\overline{\Omega}^c)$ which satisfies*

$$\|\widetilde{V}^e\phi\|_{H^1(K)} \le C_K\|\phi\|_{H^{-1/2}(\Gamma)} \quad \text{for all } \phi \in H^{-1/2}(\Gamma),\ K \subseteq \Omega^c \text{ compact.}$$

Proof. Assume $K \subseteq \Omega^c$ compact and $\phi \in H^{-1/2}(\Gamma)$. Choose $r > 0$, such that $K \subseteq B_r(0)$ and define $\widetilde{\Omega} := B_r(0) \setminus \overline{\Omega}$. The domain $\widetilde{\Omega}$ is Lipschitz, and therefore we can define $\widetilde{V}_{\widetilde{\Omega}} : H^{-1/2}(\partial\widetilde{\Omega}) \to H^1(\widetilde{\Omega})$. For any $\psi \in H^{1/2}(\partial\widetilde{\Omega})$ holds

$$\|\psi\|_{H^{1/2}(\Gamma)} \le \|\psi\|_{H^{1/2}(\partial\widetilde{\Omega})}.$$

Thus, we can extend $\phi \in H^{-1/2}(\Gamma)$ to a functional $\tilde{\phi}$ on $H^{1/2}(\partial\widetilde{\Omega})$, by

$$\tilde{\phi}(\psi) := \langle \phi, \psi|_\Gamma \rangle_\Gamma \quad \text{for all } \psi \in H^{1/2}(\partial\widetilde{\Omega}).$$

Note that $\|\tilde{\phi}\|_{H^{-1/2}(\partial\widetilde{\Omega})} \le \|\phi\|_{H^{-1/2}(\Gamma)}$. For any $\phi \in L^2(\Gamma)$, the extension $\tilde{\phi}$ reads

$$\tilde{\phi} \in L^2(\partial\widetilde{\Omega}), \quad \tilde{\phi}(x) = \phi(x) \quad \text{for } x \in \Gamma, \quad \tilde{\phi}(x) = 0 \quad \text{for } x \in \partial\widetilde{\Omega} \setminus \Gamma.$$

We define

$$(\widetilde{V}^e\phi)\Big|_K = (\widetilde{V}_{\widetilde{\Omega}}(\tilde{\phi}))\Big|_K.$$

Let $(\phi_\ell)_{\ell\in\mathbb{N}} \subseteq L^2(\Gamma)$ be a sequence with $\phi_\ell \to \phi$ in $H^{-1/2}(\Gamma)$. Then, $\|\tilde{\phi} - \tilde{\phi}_\ell\|_{H^{-1/2}(\partial\widetilde{\Omega})} \le \|\phi - \phi_\ell\|_{H^{-1/2}(\Gamma)} \to 0$ by definition, i.e $\tilde{\phi}_\ell \to \tilde{\phi}$ in $H^{-1/2}(\partial\widetilde{\Omega})$. It follows

$$(\widetilde{V}_{\widetilde{\Omega}}(\tilde{\phi}))\Big|_K = \lim_{\ell\to\infty} (\widetilde{V}_{\widetilde{\Omega}}(\tilde{\phi}_\ell))\Big|_K = \lim_{\ell\to\infty} (\widetilde{V}(\phi_\ell))\Big|_K.$$

This shows, that the definition of $\widetilde{V}^e$ is independent of the choice of $\widetilde{\Omega}$. Now, we conclude

$$\|\widetilde{V}^e\phi\|_{H^1(K)} \le \|\widetilde{V}^e\phi\|_{H^1(\widetilde{\Omega})} = \|\widetilde{V}_{\widetilde{\Omega}}\tilde{\phi}\|_{H^1(\widetilde{\Omega})} \le C\|\tilde{\phi}\|_{H^{-1/2}(\partial\widetilde{\Omega})} \le C\|\phi\|_{H^{-1/2}(\Gamma)}.$$

Let $\widehat{V}$ denote another extension of $\widetilde{V}^e$. Then, the last estimate provides

$$\begin{aligned}\|\widetilde{V}^e\phi - \widehat{V}\phi\|_{H^1(K)} &\le \|\widetilde{V}^e\phi - \widetilde{V}^e\phi_\ell\|_{H^1(K)} + \|\widetilde{V}^e\phi_\ell - \widehat{V}\phi_\ell\|_{H^1(K)} + \|\widehat{V}\phi_\ell - \widehat{V}\phi\|_{H^1(K)} \\ &\le C\|\phi - \phi_\ell\|_{H^{-1/2}(\Gamma)},\end{aligned}$$

for any $\phi_\ell \in L^2(\Gamma)$, $\phi \in H^{-1/2}(\Gamma)$. This, and the fact that $L^2(\Gamma)$ is dense in $H^{-1/2}(\Gamma)$, yield the uniqueness. ∎

By definition of $\widetilde{V}^e$, we see

$$(-\Delta\widetilde{V}^e\phi)\Big|_K = -\Delta(\widetilde{V}^e\phi)\Big|_K = -\Delta\widetilde{V}_{\widetilde{\Omega}}\tilde{\phi} = 0 \in \widetilde{H}^{-1}(K),$$

for any compact $K \subseteq \Omega^c$ and $\phi \in H^{-1/2}(\Gamma)$.

Lemma 3.5. *There holds $\widetilde{K}^e\psi \in H^1_{\ell oc}(\overline{\Omega}^c)$ with $\|\widetilde{K}^e\psi\|_{H^1(K)} \leq C_K\|\psi\|_{H^{1/2}(\Gamma)}$ for any $\psi \in H^{1/2}(\Gamma)$ and for any compact $K \subseteq \Omega^c$.*

Proof. Assume $K \subseteq \Omega^c$ compact and $\psi \in H^{1/2}(\Gamma)$. Choose $r > 0$, such that $K \subseteq B_r(0)$ and define $\widetilde{\Omega} := B_r(0) \setminus \overline{\Omega}$. The domain $\widetilde{\Omega}$ is Lipschitz, and therefore we can define $\widetilde{K}_{\widetilde{\Omega}} : H^{1/2}(\partial\widetilde{\Omega}) \to H^1(\widetilde{\Omega})$. The distance δ between Γ and $\partial B_r(0)$ is positive, therefore $|x-y| > \delta > 0$ for all $(x,y) \in (\Gamma \times \partial B_r(0)) \cup (\partial B_r(0) \times \Gamma)$. Extend any $\psi \in H^{1/2}(\Gamma)$ by zero to $\tilde{\psi} : \partial\widetilde{\Omega} \to \mathbb{R}$. There holds

$$\begin{aligned}
|\tilde{\psi}|^2_{H^{1/2}(\partial\widetilde{\Omega})} &= \int_\Gamma\int_\Gamma \frac{|\psi(x)-\psi(y)|^2}{|x-y|^2}\,\mathrm{d}\Gamma(y)\,\mathrm{d}\Gamma(x) + \int_\Gamma\int_{\partial\widetilde{\Omega}\setminus\Gamma} \frac{|\psi(x)|^2}{|x-y|^2}\,\mathrm{d}\Gamma(y)\,\mathrm{d}\Gamma(x) \\
&\quad + \int_{\partial\widetilde{\Omega}\setminus\Gamma}\int_\Gamma \frac{|\psi(y)|^2}{|x-y|^2}\,\mathrm{d}\Gamma(y)\,\mathrm{d}\Gamma(x) \\
&\leq |\psi|^2_{H^{1/2}(\Gamma)} + C\|\psi\|^2_{L^2(\Gamma)} \leq C\|\psi\|^2_{H^{1/2}(\Gamma)},
\end{aligned}$$

where $C > 0$ depends on $\widetilde{\Omega}$ and δ. It follows $\tilde{\psi} \in H^{1/2}(\partial\widetilde{\Omega})$. Thus, we get

$$\|\widetilde{K}^e\psi\|_{H^1(K)} \leq \|\widetilde{K}^e\psi\|_{H^1(\widetilde{\Omega})} \leq \|\widetilde{K}_{\widetilde{\Omega}}\tilde{\psi}\|_{H^1(\widetilde{\Omega})} \leq C\|\tilde{\psi}\|_{H^{1/2}(\partial\widetilde{\Omega})} \leq C\|\psi\|_{H^{1/2}(\Gamma)},$$

which proves the statement. □

From now on, we do not make any notational difference between $\widetilde{V}$ and $\widetilde{V}^e$ as well as $\widetilde{K}$ and $\widetilde{K}^e$. The usual notation for the three integral operators, their traces, and normal derivatives is the following:

- $N_0 := \gamma\widetilde{N} \in L(\widetilde{H}^{-1}(\Omega), H^{1/2}(\Gamma))$
- $N_1 := \partial_n\widetilde{N} \in L(\widetilde{H}^{-1}(\Omega), H^{-1/2}(\Gamma))$
- $V := \gamma\widetilde{V} \in L(H^{-1/2}(\Gamma), H^{1/2}(\Gamma))$, *simple-layer potential*
- $K' := \partial_n\widetilde{V} - \frac{1}{2} \in L(H^{-1/2}(\Gamma), H^{-1/2}(\Gamma))$, *adjoint double-layer potential*
- $K := \gamma\widetilde{K} + \frac{1}{2} \in L(H^{1/2}(\Gamma), H^{1/2}(\Gamma))$, *double-layer potential*
- $W := -\partial_n\widetilde{K} \in L(H^{1/2}(\Gamma), H^{-1/2}(\Gamma))$, *hypersingular integral operator*

There are some fundamental relations between the operators.

Theorem 3.6 (jump conditions)**.** *There holds*

- $\gamma\widetilde{V} = \gamma^e\widetilde{V}$
- $\partial_n\widetilde{V} = \frac{1}{2} + K'$
- $\partial_n^e\widetilde{V} = -\frac{1}{2} + K'$
- $\gamma\widetilde{K} = -\frac{1}{2} + K$,
- $\gamma^e\widetilde{K} = \frac{1}{2} + K$,
- $\partial_n\widetilde{K} = \partial_n^e\widetilde{K} = -W$.

The proof of these statements can be found in [15, Lemma 6.4 - 6.9]. There are some further nontrivial properties about V, K, and W, which are now stated (cf. [12]).

Theorem 3.7. (i). *The operator V is symmetric, i.e.*

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

(ii). *Provided that* $\operatorname{diam}(\Omega) < 1$*, the operator V is elliptic, i.e.*

$$\langle V\phi, \phi\rangle_\Gamma \geq C_V \|\phi\|^2_{H^{-1/2}(\Gamma)} \quad \text{for all } \phi \in H^{-1/2}(\Gamma),$$

for some $C_V > 0$ depending only on Γ.

(iii). *The operator W is symmetric, i.e.*

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

(iv). *The operator W is positive semidefinite, i.e.*

$$\langle W\phi, \phi\rangle_\Gamma \geq 0 \quad \text{for all } \phi \in H^{1/2}(\Gamma).$$

(v). *The operator W is $H^{1/2}_*(\Gamma)$-elliptic, i.e*

$$\langle W\phi, \phi\rangle_\Gamma \geq C_W \|\phi\|^2_{H^{1/2}(\Gamma)} \quad \text{for all } \phi \in H^{1/2}_*(\Gamma) := \{w \in H^{1/2}(\Gamma) \,:\, \int_\Gamma w \,\mathrm{d}\Gamma = 0\},$$

where $C_W > 0$ depends on Γ. One can show that $\ker W = \mathbb{R}$*, i.e. the kernel of W are the constant functions.*

(vi). *The operator K' is the adjoint of K in the functional analytic sense, i.e.*

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

Additionally, there holds $Kc = -\frac{1}{2}c$ for any constant function c.

Lemma 3.8. *For $u \in H^{1/2}(\Gamma)$ with $\langle Wu\,, u\rangle_\Gamma = 0$ follows $u = c \ \in \mathbb{R}$.*

Proof. Let $u = u_0 + c$ be a decomposition of u with $u_0 \in H^{1/2}_*(\Gamma)$ and $c \in \mathbb{R}$. By use of (iii) and (v) from Theorem 3.7, we conclude

$$\begin{aligned} 0 &= \langle Wu\,, u\rangle_\Gamma = \langle Wu_0\,, u_0\rangle_\Gamma + \langle Wu_0\,, c\rangle_\Gamma + \langle Wc\,, u_0 + c\rangle_\Gamma \\ &= \langle Wu_0\,, u_0\rangle_\Gamma + \langle u_0\,, Wc\rangle_\Gamma = \langle Wu_0\,, u_0\rangle \geq C_W \|u_0\|^2_{H^{1/2}(\Gamma)} \geq 0. \end{aligned}$$

Thus, $u_0 = 0$ and $u = c$. □

We want to study the behavior of $\widetilde{K}\phi$ at infinity.

Theorem 3.9. *For any $\phi \in H^{1/2}(\Gamma)$, the function $\widetilde{K}\phi$ fulfills the condition*

$$\widetilde{K}\phi(x) = \mathcal{O}(1/|x|) \quad \text{as } |x| \to \infty.$$

Proof. There holds $\nabla_y \log|x-y| = -|x-y|^{-2}(x_1 - y_1, x_2 - y_2)$. Hence, we get

$$|\nabla_y \log|x-y|| = \frac{1}{|x-y|}.$$

Because of the boundedness of Γ, there exists a constant $C > 0$ with $|y| \leq C$ for all $y \in \Gamma$. Consequently, we obtain for $|x| > 2C$.

$$\frac{1}{|x-y|} \leq \frac{1}{\frac{|x-y|}{|x|}}\frac{1}{|x|} \leq \frac{1}{1 - \frac{C}{|x|}}\frac{1}{|x|} \leq 2\frac{1}{|x|} \quad y \in \Gamma.$$

Now, we obtain

$$|\widetilde{K}\phi(x)| = \frac{1}{2\pi}\left|\int_\Gamma \partial_{n(y)} \log|x-y|\phi(y)\,\mathrm{d}\Gamma(y)\right| \leq \frac{1}{2\pi}\int_\Gamma |\nabla_y \log|x-y||\,|\phi(y)|\,\mathrm{d}\Gamma(y)$$

$$\leq 2\left(\int_\Gamma 1\,\mathrm{d}\Gamma(y)\right)^{1/2} \|\phi\|_{L^2(\Gamma)}\frac{1}{|x|} \quad \text{for all } |x| > 2C,$$

which is the stated condition. □

There holds a similar result for $\widetilde{V}\phi$.

Theorem 3.10. *For any $\phi \in H^{-1/2}(\Gamma)$, the function $\widetilde{V}\phi$ fulfills the condition*

$$\widetilde{V}\phi(x) = \frac{-1}{2\pi}\langle \phi\,, 1\rangle_\Gamma \log|x| + \mathcal{O}(1/|x|) \quad \text{as } x \to \infty,$$

almost everywhere in Ω^{ext}.

Proof. The space $L^2(\Gamma)$ is dense in $H^{-1/2}(\Gamma)$. Thus, for any $\varepsilon > 0$, we find a function ϕ_ε with $\|\phi - \phi_\varepsilon\|_{H^{-1/2}(\Gamma)} < \varepsilon$. Consider $|x| > 2C$, with $|y| < C$ for all $y \in \Gamma$.

$$\widetilde{V}\phi_\varepsilon(x) = \frac{-1}{2\pi}\left(\log|x| \int_\Gamma \phi_\varepsilon(y)\,\mathrm{d}\Gamma(y) + \int_\Gamma \log\frac{|x-y|}{|x|}\phi_\varepsilon(y)\,\mathrm{d}\Gamma(y)\right). \tag{5}$$

It is easy to see that

$$\log\frac{|x-y|}{|x|} \leq \frac{|x-y|}{|x|} - 1 \leq \frac{|y|}{|x|}, \tag{6}$$

and analogously to the proof of Theorem 3.9 we conclude

$$|\nabla_y \log\tfrac{|x-y|}{|x|}| = |\nabla_y \log|x-y|| = \frac{1}{|x-y|} \leq 2\frac{1}{|x|} \quad |x| > 2C,\ y \in \Gamma. \tag{7}$$

We use the continuity of the interior trace operator γ to derive an upper bound for $\|\log\frac{|x-\cdot|}{|x|}\|_{H^{1/2}(\Gamma)}$. There holds

$$\|\gamma(\log\tfrac{|x-\cdot|}{|x|})\|_{H^{1/2}(\Gamma)} \leq C\|\log\tfrac{|x-\cdot|}{|x|}\|_{H^1(\Omega^{int})} = C\left(\|\log\tfrac{|x-\cdot|}{|x|}\|_{L^2(\Omega^{int})} + \|\nabla_y(\log\tfrac{|x-\cdot|}{|x|})\|_{L^2(\Omega^{int})}\right).$$

With the estimates (6) and (7), we obtain

$$\|\log\tfrac{|x-\cdot|}{|x|}\|_{H^{1/2}(\Gamma)} \leq C\frac{1}{|x|}\left(\left(\int_{\Omega^{int}}|y|^2\right)^{1/2} + 2\left(\int_{\Omega^{int}}\mathrm{dx}\right)^{1/2}\right).$$

Note that because of the boundedness of Γ, there holds $1 \in H^{1/2}(\Gamma)$. Thus, we may treat the integrals in (5) as duality brackets as stated in Lemma 2.1 and write

$$\widetilde{V}\phi_\varepsilon(x) = \frac{-1}{2\pi}\log|x|\langle\phi_\varepsilon, 1\rangle_\Gamma + \langle\phi_\varepsilon, \log\tfrac{|x-\cdot|}{|x|}\rangle_G,$$

where due to the definition of duality holds

$$|\langle\phi_\varepsilon, \log\tfrac{|x-\cdot|}{|x|}\rangle_G| \leq \|\phi_\varepsilon\|_{H^{-1/2}(\Gamma)}\|\log\tfrac{|x-\cdot|}{|x|}\|_{H^{1/2}(\Gamma)} \leq \widetilde{C}\frac{1}{|x|}\|\phi_\varepsilon\|_{H^{-1/2}(\Gamma)} \quad |x| > 2C.$$

Assume, there exists a set M with positive Lebesgue measure such that

$$|V\phi(x)| > \frac{-1}{2\pi}\langle\phi, 1\rangle_\Gamma \log|x| + \widetilde{C}\frac{1}{|x|}\|\phi\|_{H^{-1/2}(\Gamma)} \quad \text{a. e. in } M. \tag{8}$$

Measure theory states the existence of a compact set K with positive measure and $K \subseteq M$. By computing the L^2-norm of (8) over K, we obtain

$$\|V\phi\|_{L^2(K)} > \frac{1}{2\pi}\langle\phi, 1\rangle_\Gamma\|\log|\cdot|\|_{L^2(K)} + \widetilde{C}\|\phi\|_{H^{-1/2}(\Gamma)}\|\tfrac{1}{|\cdot|}\|_{L^2(K)},$$

whence we find a constant $\delta > 0$ with

$$\|V\phi\|_{L^2(K)} > \delta + \frac{1}{2\pi}\langle \phi\,, 1\rangle_\Gamma \|\log|\cdot|\|_{L^2(K)} + \widetilde{C}\|\phi\|_{H^{-1/2}(\Gamma)}\|\tfrac{1}{|\cdot|}\|_{L^2(K)}.$$

Lemma 3.4 states the existence of ϕ_ε with $\|V\phi - V\phi_\varepsilon\|_{L^2(K)} \leq \|V\phi - V\phi_\varepsilon\|_{H^1(K)} \leq \delta/2$. Thus, we conclude

$$\begin{aligned}
\|V\phi - V\phi_\varepsilon\|_{L^2(K)} &\geq \|V\phi\|_{L^2(K)} - \|V\phi_\varepsilon\|_{L^2(K)} \\
&\geq \delta + \frac{1}{2\pi}\langle \phi - \phi_\varepsilon\,, 1\rangle_\Gamma \|\log|\cdot|\|_{L^2(K)} \\
&\quad + \widetilde{C}\left(\|\phi\|_{H^{-1/2}(\Gamma)} - \|\phi_\varepsilon\|_{H^{-1/2}(\Gamma)}\right)\|\tfrac{1}{|\cdot|}\|_{L^2(K)} \\
&\geq \delta - \frac{1}{2\pi}\|\log|\cdot|\|_{L^2(K)}\varepsilon + \widetilde{C}\|\tfrac{1}{|\cdot|}\|_{L^2(K)}\varepsilon.
\end{aligned}$$

With ε sufficiently small, we derive a contradiction. □

3.1 Notation

To avoid unhandy notation, we wont always distinguish between a function $u \in H^1(\Omega)$ and its trace γu or $\gamma^e u$.

- Use in the scalar product, e.g.

$$\langle u\,, v\rangle_\Gamma \text{ instead of } \langle \gamma u\,, \gamma^e v\rangle_\Gamma \quad u \in H^1(\Omega),\, v \in H^1_{\ell oc}(\mathbb{R}^2 \setminus \overline{\Omega}).$$

- Use in combination with an operator, e.g.

$$\left(\tfrac{1}{2} - K\right) u \text{ instead of } \left(\tfrac{1}{2} - K\right)\gamma u \quad u \in H^1(\Omega).$$

4 Representation Formulas

It is a well-known result of the theory of boundary integral equations (e.g [12, Theorem 15]) that for a solution $u \in H^1(\Omega^{int})$ of $-\Delta u = f \in H^{-1}(\Omega^{int})$ holds the following representation formula

$$u = \widetilde{N} f + \widetilde{V}\partial_n u - \widetilde{K}\gamma u \quad \text{a. e. in } \Omega^{int}. \tag{9}$$

Additionally, the Cauchy data $(\gamma u, \partial_n u) \in H^{1/2}(\Gamma) \times H^{-1/2}(\Gamma)$ satisfy the Calderón System

$$\begin{pmatrix} \gamma u \\ \partial_n u \end{pmatrix} = \begin{pmatrix} \frac{1}{2} - K & V \\ W & \frac{1}{2} + K' \end{pmatrix} \begin{pmatrix} \gamma u \\ \partial_n u \end{pmatrix} + \begin{pmatrix} N_0 f \\ N_1 f \end{pmatrix}. \tag{10}$$

If we want to write the exterior part of (1) as an integral equation on the boundary Γ, we need a representation formula which is valid on Ω^{ext}, and therefore the radiation condition

$u(x) = \mathcal{O}(|x|^{-1})$ as $|x| \to \infty$ in (1e) plays an important role. Compare [10, Lemma 7.11] and [10, Theorem 7.12] to see that the solution $u \in H^1_{\ell oc}(\Omega^{ext})$ of

$$-\Delta u = 0 \quad \text{in } \Omega^{ext} \tag{11}$$

satisfies the exterior representation formula

$$u = \mathcal{M}u - \widetilde{V}\partial_n^e u + \widetilde{K}u \quad \text{in } \Omega^{ext},$$

with a certain function $\mathcal{M}u \in C^\infty(\mathbb{R}^2)$. In some special cases, the function $\mathcal{M}u$ is easy to determine. The following theorem is taken from [10, Theorem 8.9].

Theorem 4.1. *Let $u \in H^1_{\ell oc}(\Omega^{ext})$ be the solution of (11).*

(i) The function u satisfies $\mathcal{M}u \equiv 0$ on $\mathbb{R}^2$ if and only if there is a constant $a \in \mathbb{R}$ such that

$$u(x) = a \log |x| + \mathcal{O}(|x|^{-1}) \quad \text{as } |x| \to \infty. \tag{12}$$

(ii) The function u satisfies $\mathcal{M}u \equiv b$ on $\mathbb{R}^2$ if and only if there is a constant $b \in \mathbb{R}$ such that

$$u(x) = b + \mathcal{O}(|x|^{-1}) \quad \text{as } |x| \to \infty. \tag{13}$$

Thus, we can choose between two different radiation conditions and still get a simple representation of $\mathcal{M}u$. We prefer the condition (13), because it is closer to the behavior of u^e in three dimensions. Note that for any solution (u^i, u^e) of the model problem (1) and any $c \in \mathbb{R}$, the pair $(u^i + c, u^e + c)$ is a solution of the model problem, too. Consequently, we are free to choose the constant $b = 0$ and get the radiation condition stated in (1).

Theorem 4.2 (exterior Representation formula)**.** *A solution $u \in H^1_{\ell oc}(\Omega^{ext})$ of*

$$\begin{aligned} -\Delta u &= 0 && \text{in } \Omega^{ext}, \\ u(x) &= \mathcal{O}(1/|x|) && \text{as } |x| \to \infty, \end{aligned}$$

satisfies the exterior representation formula

$$u = -\widetilde{V}\partial_n^e u + \widetilde{K}u, \quad \text{in } \Omega^{ext}. \tag{14}$$

With this result, we can give some crucial conditions for the data f and ϕ_0 from (1).

Theorem 4.3 (solvability of the model problem)**.** *A necessary condition for the solvability of the model problem* (1) *is*

$$\langle \phi_0 \, , 1 \rangle_\Gamma + \langle f \, , 1 \rangle_{\Omega^{int}} = 0. \tag{15}$$

Proof. The function u^e in (1) satisfies the conditions of Theorem 4.2 and therefore, we know

$$u^e = -\widetilde{V}\partial_n^e u^e + \widetilde{K}\gamma^e u^e, \quad \text{in } \Omega^{ext}.$$

With Theorem 3.10 and Theorem 3.9, we see that u^e necessarily has to fulfill $\langle \partial_n^e u^e, 1\rangle_\Gamma = 0$ to satisfy the radiation condition (1e). With the jump condition (1d), we can rewrite the weak formulation (2) of the interior problem as follows

$$a(u^i, v) - \langle \partial_n^e u^e, v\rangle_\Gamma = \langle \phi_0, v\rangle_\Gamma + \langle f, v\rangle_{\Omega^{int}} \quad \text{for } v \in H^1(\Omega^{int}).$$

Testing with $v = 1$ yields

$$0 = -\langle \partial_n^e u^e, 1\rangle_\Gamma = \langle \phi_0, 1\rangle_\Gamma + \langle f, 1\rangle_{\Omega^{int}}.$$

□

Note that with the solvability results in Theorem 5.5, Theorem 6.3 and Theorem 7.6, we see that the conditions from Theorem 4.3 are also sufficient.

5 Method 1: Simple-Layer Ansatz (Bielak-MacCamy Coupling)

5.1 Weak Form and Stability Analysis

This method first appeared in [3] and is also known as *Bielak-MacCamy-Coupling*. We choose a representation for the exterior solution u^e only using the simple-layer potential and an unknown density $\phi \in H^{-1/2}(\Gamma)$

$$u^e(x) = \left(\widetilde{V}\phi\right)(x) \quad \text{in } \Omega^{ext}. \tag{16}$$

Theorem 3.10 states that there has to hold $\langle \phi, 1\rangle_\Gamma = 0$ if we want u^e to satisfy the radiation condition (1e). Note that the representation in (16) can easily be found by defining $\phi = V^{-1}\gamma^e u^e$. Section 3 states $-\Delta\widetilde{V}\phi = 0$ in Ω^{ext} and by definition, we have $\gamma^e\widetilde{V}\phi = \gamma^e\widetilde{V}V^{-1}\gamma^e u^e = u^e$. Thus, $\widetilde{V}\phi$ and u^e are solutions of

$$\begin{aligned} -\Delta u &= 0 && \text{in } \Omega^{ext}, \\ \gamma^e u &= \gamma^e u^e && \text{on } \Gamma. \end{aligned}$$

Therefore, we obtain $u^e = \widetilde{V}\phi$, cf. [10, Theorem 8.10]. Now, we can reformulate the model problem.

Definition 5.1. *Find* $(u, \phi) \in H^1(\Omega^{int}) \times H^{-1/2}(\Gamma)$ *such that*

$$\boxed{\begin{aligned} a(u,v) + \langle \left(\tfrac{1}{2} - K'\right)\phi, v\rangle_\Gamma &= \langle f, v\rangle_{\Omega^{int}} + \langle \phi_0, v\rangle_\Gamma && \text{for all } v \in H^1(\Omega^{int}) \\ \langle u, \psi\rangle_\Gamma - \langle V\phi, \psi\rangle_\Gamma &= \langle u_0, \psi\rangle_\Gamma && \text{for all } \psi \in H^{-1/2}(\Gamma) \end{aligned}} \tag{17}$$

Theorem 5.2. *Assume that* $\langle \phi_0 \, , 1 \rangle_\Gamma + \langle f \, , 1 \rangle_{\Omega^{int}} = 0$. *Then, the formulations* (1) *and* (17) *are equivalent in the following sense:*

(i) *If* $(u^i, u^e) \in H^1(\Omega^{int}) \times H^1_{\ell oc}(\Omega^{ext})$ *solves* (1), *the pair* $(u, \phi) \in H^1(\Omega^{int}) \times H^{-1/2}(\Gamma)$ *with* $u := u^i$ *and* $\phi := V^{-1}\gamma^e u^e$ *solves* (17).

(ii) *If* $(u, \phi) \in H^1(\Omega^{int}) \times H^{-1/2}(\Gamma)$ *solves* (17), *the definition* $u^i := u$ *and* $u^e := \widetilde{V}\phi$ *provides a solution of* (1).

Proof. Step 1: Suppose that (u^i, u^e) solves the model problem (1), then for $u = u^i$ holds the first Green's formula

$$a(u, v) - \langle \partial_n u \, , v \rangle_\Gamma = \langle f \, , v \rangle_{\Omega^{int}} \quad \text{for all } v \in H^1(\Omega^{int}).$$

By use of the jump condition $\partial_n u - \partial_n^e u^e = \phi_0$ of (1d), this equation reads

$$a(u, v) - \langle \partial_n^e u^e \, , v \rangle_\Gamma = \langle f \, , v \rangle_{\Omega^{int}} + \langle \phi_0 \, , v \rangle_\Gamma. \tag{18}$$

Finally, we replace $\partial_n^e u^e = \partial_n^e \widetilde{V}\phi = (K' - 1/2)\phi$ as stated in Theorem 3.6 and obtain

$$a(u, v) + \langle (\tfrac{1}{2} - K')\phi \, , v \rangle_\Gamma = \langle f \, , v \rangle_{\Omega^{int}} + \langle \phi_0 \, , v \rangle_\Gamma \quad \text{for all } v \in H^1(\Omega^{int}), \tag{19}$$

which is the first equation of (17). There holds $\langle u^e \, , \psi \rangle_\Gamma - \langle V\phi \, , \psi \rangle_\Gamma = 0$ for all $\psi \in H^{-1/2}(\Gamma)$. This follows from the representation of u^e and the fact that $\gamma\widetilde{V} = V$. If we apply the jump condition $\gamma u - \gamma^e u^e = u_0$, we receive the second equation of (17)

$$\langle u \, , \psi \rangle_\Gamma - \langle V\phi \, , \psi \rangle_\Gamma = \langle u_0 \, , \psi \rangle_\Gamma \quad \text{for all } \psi \in H^{-1/2}(\Gamma).$$

Step 2: Suppose that (u, ϕ) solves (17). We start with the first equation of (17). As shown in *Step 1* , it is equivalent to

$$a(u^i, v) - \langle \partial_n^e \widetilde{V}\phi \, , v \rangle_\Gamma = \langle f \, , v \rangle_{\Omega^{int}} + \langle \phi_0 \, , v \rangle_\Gamma \quad \text{for all } v \in H^1(\Omega^{int}). \tag{20}$$

This is the weak formulation of

$$-\Delta u^i = f \quad \text{in } \Omega^{int},$$
$$\partial_n u^i - \partial_n^e \widetilde{V}\phi = \phi_0.$$

which are the equations (1a) and (1d). Replace $V\phi$ by u^e in the second equation of (17) to obtain $\langle u \, , \psi \rangle_\Gamma - \langle u^e \, , \psi \rangle_\Gamma = \langle u_0 \, , \psi \rangle_\Gamma$ for all $\psi \in H^{-1/2}(\Gamma)$. This is the trace jump in (1c). Theorem 3.2 shows that for all $\phi \in H^{-1/2}(\Gamma)$ holds $\Delta\widetilde{V}\phi = 0$ in $\mathbb{R}^2 \backslash \Gamma$. It remains to prove the radiation condition. At the beginning of this section, we saw that $\langle \phi \, , 1 \rangle_\Gamma = 0$ is sufficient for $u^e := \widetilde{V}\phi$ to satisfy $u^e(x) = \mathcal{O}(1/|x|)$ as $|x| \to \infty$. Testing the first equation of (17) with $v = 1$, we see

$$\langle \left(\tfrac{1}{2} - K'\right) \phi \, , 1 \rangle_\Gamma = \langle f \, , 1 \rangle_{\Omega^{int}} + \langle \phi_0 \, , 1 \rangle_\Gamma.$$

With $K1 = -1/2$, this is equivalent to

$$\langle \phi \, , 1 \rangle = \langle \phi \, , \left(\tfrac{1}{2} - K\right) 1 \rangle_\Gamma = \langle f \, , 1 \rangle_{\Omega^{int}} + \langle \phi_0 \, , 1 \rangle_\Gamma = 0.$$

The last equation holds due to the properties of f and ϕ_0 stated in Theorem 4.3. □

Now, we want to prove the solvability of (17). For $(d_1, d_2) \in \widetilde{H}^{-1}(\Omega^{int}) \times H^{1/2}(\Gamma)$, we consider the problem:
Find $(u, \phi) \in H^1(\Omega^{int}) \times H^{-1/2}(\Gamma)$ such that

$$\begin{aligned} a(u,v) + \langle \left(\tfrac{1}{2} - K'\right)\phi \,, v\rangle_\Gamma &= d_1(v) \qquad \text{for all } v \in H^1(\Omega^{int}), \\ \langle u \,, \psi\rangle_\Gamma - \langle V\phi \,, \psi\rangle_\Gamma &= d_2(\psi) \qquad \text{for all } \psi \in H^{-1/2}(\Gamma). \end{aligned} \tag{21}$$

For the special choices

$$d_1 = \langle f \,, \cdot\rangle_{\Omega^{int}} + \langle \phi_0 \,, \cdot\rangle_\Gamma \quad \text{and} \quad d_2 = \langle u_0 \,, \cdot\rangle_\Gamma, \tag{22}$$

this is an equivalent formulation of (17). Note that the second equation of (21) is equivalent to $V\phi = \gamma u - d_2$. Since V is elliptic, we may eliminate ϕ in the first equation of (21). This leads to

$$a(u,v) + \langle (\tfrac{1}{2} - K')V^{-1}\left(u - d_2\right), v\rangle_\Gamma = d_1(v) \quad \text{for all } v \in H^1(\Omega^{int}),$$

and finally to

$$a(u,v) + \langle \tfrac{1}{2}V^{-1}u \,, v\rangle_\Gamma - \langle K'V^{-1}u \,, v\rangle_\Gamma = \langle d_1 \,, v\rangle_{\Omega^{int}} + \langle (\tfrac{1}{2} - K')V^{-1}d_2 \,, v\rangle_\Gamma =: L(v), \tag{23}$$

for all $v \in H^1(\Omega^{int})$. If we choose d_1, d_2 as in (22), we see that there holds $L(v) = \langle f \,, v\rangle_{\Omega^{int}} + \langle \phi_0 \,, v\rangle_\Gamma + \langle \left(\tfrac{1}{2} - K'\right)V^{-1}u_0 \,, v\rangle_\Gamma$. We define the bilinear forms

$$b_1(u,v) := a(u,v) + \langle \tfrac{1}{2}V^{-1}u \,, v\rangle_\Gamma \text{ and } b_2(u,v) := -\langle K'V^{-1}u \,, v\rangle_\Gamma.$$

for all pairs $(u, v) \in H^1(\Omega^{int}) \times H^1(\Omega^{int})$. Now, (23) reads as follows

$$b_1(u,v) + b_2(u,v) = L(v) \quad \forall v \in H^1(\Omega^{int}). \tag{24}$$

Proposition 5.3. *The operator $B_1 : H^1(\Omega^{int}) \to \widetilde{H}^{-1}(\Omega^{int})$ defined by $\langle B_1 u \,, v\rangle_{\Omega^{int}} = b_1(u,v)$ for all $v \in H^1(\Omega^{int})$ is an isomorphism.*

Proof. The operator V is elliptic and bounded. For every $u \in H^{1/2}(\Gamma)$, we can find a $w \in H^{-1/2}(\Gamma)$ with $Vw = u$. Thus, with the continuity of V, there holds

$$\langle V^{-1}u \,, u\rangle_\Gamma = \langle w \,, Vw\rangle_\Gamma \geq C_V \|w\|^2_{H^{-1/2}(\Gamma)} \geq C\|u\|^2_{H^{1/2}(\Gamma)}.$$

In other words, V^{-1} is elliptic. We see

$$b_1(u,u) = a(u,u) + \langle \tfrac{1}{2}V^{-1}u \,, u\rangle_\Gamma \geq \|\nabla u\|^2_{L^2(\Omega^{int})} + C\|u\|^2_{H^{1/2}(\Gamma)}.$$

As stated in Theorem 2.4, $\|u\|^2 := \|u\|^2_{H^{1/2}(\Gamma)} + \|\nabla u\|^2_{L^2(\Omega^{int})}$ defines an equivalent norm on $H^1(\Omega^{int})$. Thus, there exists a positive constant C_1 with $b_1(u,u) \geq C_1\|u\|^2_{H^1(\Omega^{int})}$. Additionally, we can prove that b_1 is bounded

$$b_1(u,v) \leq \|\nabla u\|_{L^2(\Omega^{int})}\|\nabla v\|_{L^2(\Omega^{int})} + C\|u\|_{H^{1/2}(\Gamma)}\|v\|_{H^{1/2}(\Gamma)} \leq C_2\|u\|_{H^1(\Omega^{int})}\|v\|_{H^1(\Omega^{int})}.$$

Where the constant C_2 depends on the continuity of V^{-1} and the trace operator γ. Given $\ell \in \widetilde{H}^{-1}(\Omega^{int})$, the Lax-Milgram lemma now states the unique solvability of

$$\langle B_1 u\,, v\rangle_{\Omega^{int}} = b_1(u,v) = \langle \ell\,, v\rangle_{\Omega^{int}} \quad \text{for all } v \in H^1(\Omega^{int}).$$

In particular, this proves the unique solvability of $B_1 u = \ell$ for all $\ell \in \widetilde{H}^{-1}(\Omega^{int})$. Hence, the operator B_1 is bijective and bounded. Now, the *open mapping theorem* proves the statement. □

The proof of the following proposition can be found in [9, Lemma 1].

Proposition 5.4. *Suppose $\Gamma \subset \mathbb{R}^2$ to be a C^2-manifold. Then, the operator $B_2 : H^1(\Omega^{int}) \mapsto \widetilde{H}^{-1}(\Omega^{int})$ defined by $\langle B_2 u\,, v\rangle_1 = b_2(u,v)$ for all $v \in H^1(\Omega^{int})$ is compact.*

By use of the operators B_1 and B_2 defined in Theorem 5.3 and 5.4, we can rewrite (24) in the following form:
Find $u \in H^1(\Omega^{int})$ such that

$$(B_1 + B_2)\, u = L$$

If we apply B_1^{-1}, the equation becomes

$$\left(I + B_1^{-1} B_2\right) u = B_1^{-1} L.$$

The operator $B_1^{-1}B_2$ is compact, and therefore we can apply *Fredholm's Alternative* which states that $I + B_1^{-1}B_2$ being injective implies that $I + B_1^{-1}B_2$ is already bijective, cf. e.g. [16, Satz VI.2.4].

Theorem 5.5. *Suppose $\Gamma \subset \mathbb{R}^2$ to be a C^2-manifold. Then, there exists a unique solution $(u,\phi) \in H^1(\Omega^{int}) \times H^{-1/2}(\Gamma)$ of (21) for any data $(d_1, d_2) \in \widetilde{H}^{-1}(\Omega^{int}) \times H^{1/2}(\Gamma)$. There holds*

$$\|u\|_{H^1(\Omega^{int})} + \|\phi\|_{H^{-1/2}(\Gamma)} \leq C \left(\|d_1\|_{\widetilde{H}^{-1}(\Omega^{int})} + \|d_2\|_{H^{1/2}(\Gamma)} \right). \tag{25}$$

Where C only depends on Ω^{int}.

Proof. Step 1: Due to the considerations above, it is sufficient to prove that $\ker\left(I + B_1^{-1}B_2\right) = \{0\}$. Assume that $\left(I + B_1^{-1}B_2\right) u = 0$ for some $u \in H^1(\Omega^{int})$. There holds

$$b_1(u,v) + b_2(u,v) = 0 \quad \text{for all } v \in H^1(\Omega^{int}),$$

which is equivalent to

$$a(u,v) + \langle \tfrac{1}{2} V^{-1} u\,, v\rangle_\Gamma - \langle K' V^{-1} u\,, v\rangle_\Gamma = 0 \quad \text{for all } v \in H^1(\Omega^{int}),$$

which is (23) with $L = 0$. With $\phi := V^{-1}u$, we can rewrite this equation as

$$a(u,v) + \langle (\tfrac{1}{2} - K')\phi\,, v\rangle_\Gamma = 0 \quad \text{for all } v \in H^1(\Omega^{int}). \tag{26}$$

Additionally, we get

$$\langle u\,,\psi\rangle_\Gamma - \langle V\phi\,,\psi\rangle_\Gamma = 0 \quad \text{for all } \psi \in H^{-1/2}(\Gamma). \tag{27}$$

The equations (26) and (27) are equivalent to (17) with $f \equiv 0$, $\phi_0 \equiv 0$, and $u_0 \equiv 0$. Theorem 5.2 shows that the pair (u^i, u^e) with $u^i = u$ and $u^e = V\phi$ solves the model problem (1) with homogeneous data, i.e

$$\begin{aligned} -\Delta u^i &= 0 \quad \text{in } \Omega^{int}, \\ -\Delta u^e &= 0 \quad \text{in } \Omega^{ext}, \\ \gamma u^i &= \gamma^e u^e \quad \text{on } \Gamma, \\ \partial_n u^i &= \partial_n^e u^e \quad \text{on } \Gamma, \\ u^e(x) &= \mathcal{O}(1/|x|) \quad \text{as } |x| \to \infty. \end{aligned}$$

Theorem 4.2 and (9) state that we can write u^i and u^e as

$$u^i = \widetilde{V}\partial_n u^i - \widetilde{K}\gamma u^i \quad \text{in } \Omega^{int}, \tag{28}$$

and

$$u^e = -\widetilde{V}\partial_n^e u^e + \widetilde{K}\gamma^e u^e \quad \text{in } \Omega^{ext}. \tag{29}$$

If we apply the interior and exterior trace operators, we obtain

$$\gamma u^i = V\partial_n u^i + (\tfrac{1}{2} - K)\gamma u^i \;\text{ and }\; \gamma^e u^e = -V\partial_n^e u^e + (\tfrac{1}{2} + K)\gamma^e u^e.$$

Together with $\gamma u^i = \gamma^e u^e$ and $\partial_n u^i = \partial_n^e u^e$, this yields

$$2\gamma u^i = \gamma u^i + \gamma^e u^e = \tfrac{1}{2}\gamma u^i + \tfrac{1}{2}\gamma^e u^e = \gamma u^i.$$

Thus, it follows $\gamma u^i \equiv 0$. On the other hand, we conclude

$$0 = \gamma u^i - \gamma^e u^e = 2(V\partial_n u^i - K\gamma u^i) = -2V\partial_n u^i,$$

and therefore $\partial_n u^i = 0$. With the formulas (28) and (29), this yields $u^i \equiv u^e \equiv 0$. Consequently, there is a unique solution (u, ϕ) of (21).

Step 2: We now know that $Au := (B_1 + B_2)\,u = L$ is uniquely solvable for any $L \in \widetilde{H}^{-1}(\Omega^{int})$. The boundedness of b_1 implies the continuity of B_1. The operator B_2 is compact and particularly continuous. Thus, $A \in L\left(H^1(\Omega^{int}), \widetilde{H}^{-1}(\Omega^{int})\right)$ and A is bijective. Now, the *open mapping theorem* proves the continuity of A^{-1}. The solution u thus satisfies

$$\begin{aligned} \|u\|_{H^1(\Omega^{int})} = \|A^{-1}L\|_{H^1(\Omega^{int})} &\leq C_A\|\langle d_1\,,\cdot\rangle_{\Omega^{int}} + \langle(\tfrac{1}{2} - K')V^{-1}d_2\,,\cdot\rangle_\Gamma\|_{\widetilde{H}^{-1}(\Omega^{int})} \\ &\leq C\left(\|d_1\|_{\widetilde{H}^{-1}(\Omega^{int})} + \|d_2\|_{H^{1/2}(\Gamma)}\right). \end{aligned}$$

The same holds for ϕ

$$\|\phi\|_{H^{-1/2}(\Gamma)} = \|V^{-1}(u - d_2)\|_{H^{-1/2}(\Gamma)} \leq \widetilde{C}\left(\|d_1\|_{\widetilde{H}^{-1}(\Omega^{int})} + \|d_2\|_{H^{1/2}(\Gamma)}\right).$$

Adding these two estimates, we conclude (25). Note that the constants only depend on V^{-1}, K', and the trace operator γ. □

Corollary 5.6. *For the special choices* $d_1 = \langle f, \cdot \rangle_{\Omega^{int}} + \langle \phi_0, \cdot \rangle_\Gamma$ *and* $d_2 = \langle u_0, \cdot \rangle_\Gamma$ *with* $(f, u_0, \phi_0) \in L^2(\Omega^{int}) \times H^{1/2}(\Gamma) \times H^{-1/2}(\Gamma)$*, the solution* (u, ϕ) *of* (21) *provides a solution of* (17)*, too. There holds*

$$\|u\|_{H^1(\Omega^{int})} + \|\phi\|_{H^{-1/2}(\Gamma)} \leq C \left(\|f\|_{L^2(\Omega^{int})} + \|u_0\|_{H^{1/2}(\Gamma)} + \|\phi_0\|_{H^{-1/2}(\Gamma)} \right) \tag{30}$$

Proof. Obviously, the norms of d_1 and d_2 are bounded by

$$\begin{aligned} \|d_1\|_{\widetilde{H}^{-1}(\Omega^{int})} &\leq \|f\|_{L^2(\Omega^{int})} + \|\phi_0\|_{H^{-1/2}(\Gamma)}, \\ \|d_2\|_{H^{1/2}(\Gamma)} &\leq \|u_0\|_{H^{1/2}(\Gamma)}. \end{aligned}$$

Now, the statement follows by use of (25). □

5.2 Discretization

To solve the problem (17) numerically, we choose finite dimensional subspaces $X_\ell \subset H^1(\Omega^{int})$ and $Y_\ell \subset H^{-1/2}(\Gamma)$. The Galerkin formulation of (17) then reads as follows: Find $(u_\ell, \phi_\ell) \in X_\ell \times Y_\ell$ such that

$$\boxed{\begin{aligned} a(u_\ell, v_\ell) + \langle \left(\tfrac{1}{2} - K'\right) \phi_\ell, v_\ell \rangle_\Gamma &= \langle f, v_\ell \rangle_{\Omega^{int}} + \langle \phi_0, v_\ell \rangle_\Gamma, \\ \langle u_\ell, \psi_\ell \rangle_\Gamma - \langle V \phi_\ell, \psi_\ell \rangle_\Gamma &= \langle u_0, \psi_\ell \rangle_\Gamma, \end{aligned}} \tag{31}$$

for any $(v_\ell, \psi_\ell) \in X_\ell \times Y_\ell$. Note that in practical applications Ω^{int} is a polygon and therefore Γ is no differentiable manifold (compare the conditions in Theorem 5.5). Thus, the stability and solvability results from above do not apply.

6 Method 2: Direct Method (Johson-Nédélec Coupling)

6.1 Weak Form and Stability Analysis

The coupling method presented in this section first appeared in [9] and is known as *Johnson-Nédélec-Coupling*. In contrast to Method 1 from Section 5, now we use both, simple-layer and double-layer potential to represent u in the exterior domain as

$$u^e = -\widetilde{V} \partial_n^e u^e + \widetilde{K} u^e \quad \text{in } \Omega^{ext}. \tag{32}$$

Theorem 3.10 and Theorem 3.9 state that $\partial_n^e u^e$ has to satisfy $\langle \partial_n^e u^e, 1 \rangle_\Gamma = 0$ to get the right behavior at infinity (1e).

Definition 6.1. *Find* $(u, \phi) \in H^1(\Omega^{int}) \times H^{-1/2}(\Gamma)$ *such that*

$$\boxed{\begin{aligned} a(u, v) - \langle \phi, v \rangle_\Gamma &= \langle f, v \rangle_{\Omega^{int}} + \langle \phi_0, v \rangle_\Gamma && \text{for all } v \in H^1(\Omega^{int}) \\ \langle \left(\tfrac{1}{2} - K\right) u, \psi \rangle_\Gamma + \langle V \phi, \psi \rangle_\Gamma &= \langle \left(\tfrac{1}{2} - K\right) u_0, \psi \rangle_\Gamma && \text{for all } \psi \in H^{-1/2}(\Gamma) \end{aligned}} \tag{33}$$

Theorem 6.2. *Assume that* $\langle \phi_0 \, , 1 \rangle_\Gamma + \langle f \, , 1 \rangle_{\Omega^{int}} = 0$. *Then, the formulations* (1) *and* (33) *are equivalent in the following sense:*

(i) *If* $(u^i, u^e) \in H^1(\Omega^{int}) \times H^1_{\ell oc}(\Omega^{ext})$ *solves* (1), *the pair* $(u, \phi) \in H^1(\Omega^{int}) \times H^{-1/2}(\Gamma)$ *with* $u := u^i$ *and* $\phi := \partial_n^e u^e$ *solves* (33).

(ii) *If* $(u, \phi) \in H^1(\Omega^{int}) \times H^{-1/2}(\Gamma)$ *solves* (33), *the definition* $u^i := u$ *and* $u^e := -\widetilde{V}\phi + \widetilde{K}(u - u_0)$ *provides a solution of* (1).

Proof. Step 1: Suppose (u^i, u^e) solves (1), then (u, ϕ) satisfies

$$a(u, v) - \langle \phi + \phi_0 \, , v \rangle_\Gamma = \langle f \, , v \rangle_{\Omega^{int}} \quad \text{for all } v \in H^1(\Omega^{int}),$$

as shown in Section 1. This is equivalently written as

$$a(u, v) - \langle \phi \, , v \rangle_\Gamma = \langle f \, , v \rangle_{\Omega^{int}} + \langle \phi_0 \, , v \rangle_\Gamma \quad \text{for all } v \in H^1(\Omega^{int}).$$

To prove the second equation, we start with the representation formula for the exterior domain (14)

$$u^e = -\widetilde{V}\partial_n^e u^e + \widetilde{K}\gamma^e u^e \quad \text{in } \Omega^{ext}.$$

If we apply the trace operator γ^e to this equation, we get

$$\left(\tfrac{1}{2} - K\right)\gamma^e u^e + V\partial_n^e u^e = 0 \quad \text{in } H^{1/2}(\Gamma).$$

By use of the jump condition (1c), the equation becomes

$$\left(\tfrac{1}{2} - K\right)\gamma u^i + V\partial_n^e u^e = \left(\tfrac{1}{2} - K\right) u_0 \quad \text{in } H^{1/2}(\Gamma).$$

Replacing u^i with u, $\partial_n^e u^e$ with ϕ, and testing the whole equation with $\psi \in H^{-1/2}(\Gamma)$, we obtain the second equation of (33).

Step 2: Suppose that (u, ϕ) solves (33), then the first equation states that u satisfies

$$\begin{array}{rcll} -\Delta u & = & f & \text{in } \Omega^{int}, \\ \partial_n u & = & \phi + \phi_0 & \text{on } \Gamma, \end{array} \tag{34}$$

in the weak sense. The second equation of (33) is equivalent to

$$\langle u \, , \psi \rangle_\Gamma = \langle (\tfrac{1}{2} + K)u \, , \psi \rangle_\Gamma - \langle V\phi \, , \psi \rangle_\Gamma + \langle (\tfrac{1}{2} - K)u_0 \, , \psi \rangle_\Gamma.$$

This leads to

$$\begin{array}{rcl} \langle u - u_0 \, , \psi \rangle_\Gamma & = & \langle (\tfrac{1}{2} + K)(u - u_0) \, , \psi \rangle_\Gamma - \langle V\phi \, , \psi \rangle_\Gamma \\ & = & \langle \gamma^e \left(\widetilde{K}(u - u_0) - \widetilde{V}\phi \right) , \psi \rangle_\Gamma \\ & = & \langle u^e \, , \psi \rangle_\Gamma \quad \text{for all } \psi \in H^{-1/2}(\Gamma), \end{array}$$

which is just the jump condition for the traces (1c). Now, we want to prove that $u^e := -\widetilde{V}\phi + \widetilde{K}(u - u_0)$ solves

$$\begin{array}{rcll} -\Delta u^e & = & 0 & \text{in } \Omega^{ext}, \\ \gamma^e u^e & = & u - u_0 & \text{on } \Gamma, \\ u^e(x) & = & \mathcal{O}(|x|^{-1}) & \text{as } |x| \to \infty. \end{array} \tag{35}$$

The first equation follows from the fact that $-\Delta\left(\widetilde{V}\lambda + \widetilde{K}w\right) = 0 \quad$ in $\mathbb{R}^2\backslash\Gamma$ for arbitrary $(\lambda, w) \in H^{-1/2}(\Gamma) \times H^1(\Omega^{int})$ as stated in Theorem 3.2 and Theorem 3.3. Testing the first equation of (33) with $v = 1$ yields $-\langle \phi\,, 1\rangle_\Gamma = \langle f\,, 1\rangle_{\Omega^{int}} + \langle \phi_0\,, 1\rangle_\Gamma = 0$, and therefore the radiation condition is satisfied. Note that we now already know, that the equations (1a), (1b), (1c), and (1e) of the model problem are satisfied. Theorem 4.2 states that for the solution of (35) holds

$$u^e = -\widetilde{V}\partial_n^e u^e + \widetilde{K}u^e \quad \text{in } \Omega^{ext}. \tag{36}$$

If we subtract $u^e = -\widetilde{V}\phi + \widetilde{K}(u - u_0)$ from (36), we get

$$0 = \widetilde{V}(\phi - \partial_n^e u^e),$$

whence

$$0 = V(\phi - \partial_n^e u^e).$$

Due to the ellipticity of V, this implies

$$\phi = \partial_n^e u^e.$$

Plugging this result into (34), we obtain the jump condition for the normal derivatives stated in (1d). □

It remains to consider the stability of this method. The result also holds for the discrete case, hence we start with the Galerkin discretization.

6.2 Discretization

As usual, we choose finite-dimensional subspaces $X_\ell \subset H^1(\Omega^{int})$ and $Y_\ell \subset H^{-1/2}(\Gamma)$. The Galerkin formulation of (33) reads as follows: Find $(u_\ell, \phi_\ell) \in X_\ell \times Y_\ell$ such that

$$\boxed{\begin{array}{rcl} a(u_\ell, v_\ell) - \langle \phi_\ell\,, v_\ell\rangle_\Gamma & = & \langle f\,, v_\ell\rangle_{\Omega^{int}} + \langle \phi_0\,, v_\ell\rangle, \\ \langle \left(\frac{1}{2} - K\right) u_\ell\,, \psi_\ell\rangle_\Gamma + \langle V\phi_\ell\,, \psi_\ell\rangle_\Gamma & = & \langle \left(\frac{1}{2} - K\right) u_0\,, \psi_\ell\rangle_\Gamma, \end{array}} \tag{37}$$

6.3 Stability Estimate for the Discrete and the Continuous Case

The following theorem provides the solvability and stability for both, the continuous and the discrete case. Thus, for subspaces $X_\ell \subseteq H^1(\Omega^{int})$, $Y_\ell \subseteq H^{-1/2}(\Gamma)$, and $(d_1, d_2) \in X_\ell^* \times Y_\ell^*$, we consider the problem:
Find $(u_\ell, \phi_\ell) \in X_\ell \times Y_\ell$ such that

$$\begin{aligned} a(u_\ell, v_\ell) - \langle \phi_\ell, v_\ell \rangle_\Gamma &= d_1(v_\ell) \qquad \text{for all } v_\ell \in X_\ell, \\ \langle \left(\tfrac{1}{2} - K\right) u_\ell, \psi_\ell \rangle_\Gamma + \langle V\phi_\ell, \psi_\ell \rangle_\Gamma &= d_2(\psi_\ell) \qquad \text{for all } \psi_\ell \in Y_\ell. \end{aligned} \tag{38}$$

Theorem 6.3 (F.-J. Sayas)**.** *For any choice of closed spaces $X_\ell \subset H^1(\Omega^{int})$ and $Y_\ell \subset H^{-1/2}(\Gamma)$ (finite or infinite dimensional) with $1 \in Y_\ell$, there exists a unique solution $(u_\ell, \phi_\ell) \in X_\ell \times Y_\ell$ of (38). There holds*

$$\|u_\ell\|_{H^1(\Omega^{int})} + \|\phi_\ell\|_{H^{-1/2}(\Gamma)} \leq C \left(\|d_1\|_{\widetilde{H}^{-1}(\Omega^{int})} + \|d_2\|_{H^{1/2}(\Gamma)} \right),$$

where the constant $C > 0$ is independent of the choice of spaces.

For a proof, consider [14]. We give a simple consequence of this result.

Corollary 6.4. *For any data $(f, u_0, \phi_0) \in L^2(\Omega^{int}) \times H^{1/2}(\Gamma) \times H^{-1/2}(\Gamma)$, there exist unique solutions $(u_\ell, \phi_\ell) \in X_\ell \times Y_\ell$ of (37) as well as $(u, \phi) \in H^1(\Omega^{int}) \times H^{-1/2}(\Gamma)$ of (33). There holds*

$$\begin{aligned} \|u_\ell\|_{H^1(\Omega^{int})} + \|\phi_\ell\|_{H^{-1/2}(\Gamma)} &\leq C \left(\|f\|_{L^2(\Omega^{int})} + \|u_0\|_{H^{1/2}(\Gamma)} + \|\phi_0\|_{H^{-1/2}(\Gamma)} \right), \\ \|u\|_{H^1(\Omega^{int})} + \|\phi\|_{H^{-1/2}(\Gamma)} &\leq C \left(\|f\|_{L^2(\Omega^{int})} + \|u_0\|_{H^{1/2}(\Gamma)} + \|\phi_0\|_{H^{-1/2}(\Gamma)} \right), \end{aligned}$$

where the constant $C > 0$ doesn't depend on the choice of spaces.

Proof. Choose $d_1 = \langle f, \cdot \rangle_{\Omega^{int}} + \langle \phi_0, \cdot \rangle \in \widetilde{H}^{-1}(\Omega^{int})$ and $d_2 = \langle \left(\frac{1}{2} - K\right) u_0, \cdot \rangle_\Gamma \in H^{1/2}(\Gamma)$. With $X_\ell \times Y_\ell = H^1(\Omega^{int}) \times H^{-1/2}(\Gamma)$, the formulations (33) and (38) are equivalent. To see the equivalence of (38) and (37), consider X_ℓ and Y_ℓ being finite-dimensional and restrict d_1 and d_2 to X_ℓ resp. Y_ℓ. In both cases, the norms of d_1 and d_2 are bounded by

$$\begin{aligned} \|d_1\|_{\widetilde{H}^{-1}(\Omega^{int})} &\leq \|f\|_{L^2(\Omega^{int})} + \|\phi_0\|_{\widetilde{H}^{-1}(\Gamma)}, \\ \|d_2\|_{H^{1/2}(\Gamma)} &\leq C \|u_0\|_{H^{1/2}(\Gamma)}, \end{aligned}$$

where $C > 0$ only depends on Γ and the continuity of K. The statement follows by use of Theorem 6.3. ∎

7 Method 3: Symmetric Coupling

This section presents the best known coupling method. Proposed by Costabel, it first appeared in [8]. The representation of the solution in the exterior domain is the same as in Section 6, namely

$$u^e = -\widetilde{V} \partial_n^e u^e + \widetilde{K} u^e \quad \text{in } \Omega^{ext}. \tag{39}$$

This two methods only differ in how the interior and the exterior solutions are linked at the boundary.

Definition 7.1. *Find* $(u,\phi) \in H^1(\Omega^{int}) \times H^{-1/2}(\Gamma)$ *such that for all* $(v,\psi) \in H^1(\Omega^{int}) \times H^{-1/2}(\Gamma)$

$$\boxed{\begin{array}{rcl} a(u,v) + \langle \left(K' - \frac{1}{2}\right)\phi\,, v\rangle_\Gamma + \langle Wu\,, v\rangle_\Gamma &=& \langle f\,, v\rangle_{\Omega^{int}} + \langle \phi_0 + Wu_0\,, v\rangle_\Gamma, \\ \langle \left(\frac{1}{2} - K\right)u\,, \psi\rangle_\Gamma + \langle V\phi\,, \psi\rangle_\Gamma &=& \langle \left(\frac{1}{2} - K\right)u_0\,, \psi\rangle_\Gamma. \end{array}} \tag{40}$$

Note that if we multiply the second equation of (40) by -1, the bilinear form induced by (40) becomes symmetric, which explains the name of the method.

Theorem 7.2. *Assume that* $\langle \phi_0\,, 1\rangle_\Gamma + \langle f\,, 1\rangle_{\Omega^{int}} = 0$. *Then, the formulations* (1) *and* (40) *are equivalent in the following sense:*

(i) *If* $(u^i, u^e) \in H^1(\Omega^{int}) \times H^1_{\ell oc}(\Omega^{ext})$ *solves* (1)*, the pair* $(u,\phi) \in H^1(\Omega^{int}) \times H^{-1/2}(\Gamma)$ *with* $u := u^i$ *and* $\phi := \partial_n^e u^e$ *solves* (40).

(ii) *If* $(u,\phi) \in H^1(\Omega^{int}) \times H^{-1/2}(\Gamma)$ *solves* (40)*, the definition* $u^i := u$ *and* $u^e := -\widetilde{V}\phi + \widetilde{K}(u - u_0)$ *provides a solution of* (1).

Proof. Step 1: As always, we start with the solution (u^i, u^e) of the model problem (1). Then, (u,ϕ) satisfies

$$a(u,v) - \langle \phi + \phi_0\,, v\rangle_\Gamma = \langle f\,, v\rangle_{\Omega^{int}} \quad \text{for all } v \in H^1(\Omega^{int}) \tag{41}$$

Applying the exterior normal derivative to the representation of u^e, we obtain

$$\phi = \partial_n^e u^e = -\left(-\tfrac{1}{2} + K'\right)\partial_n^e u^e - Wu^e \quad \text{in } H^{-1/2}(\Gamma).$$

Plugging this result into (41), we get

$$a(u,v) + \langle (-\tfrac{1}{2} + K')\partial_n^e u^e\,, v\rangle_\Gamma + \langle Wu^e\,, v\rangle_\Gamma = \langle f\,, v\rangle_{\Omega^{int}} + \langle \phi_0\,, v\rangle_\Gamma \quad v \in H^1(\Omega^{int}).$$

Now, we use the jump conditions for the trace (1c) and replace $\partial_n^e u^e$ by ϕ. This yields the first equation of (40). The second equation of (40) is the same as used in Method 2, cf. (33). Hence, we omit the proof.

Step 2: Suppose the pair (u,ϕ) solves (40). The first equation of (40) is equivalently written

$$a(u,v) + \langle (K' - \tfrac{1}{2})\phi + W(u - u_0)\,, v\rangle_\Gamma = \langle f\,, v\rangle_{\Omega^{int}} + \langle \phi_0\,, v\rangle_\Gamma. \tag{42}$$

Similar to *Step 1*, we see that $\partial_n^e u^e = \partial_n^e \left(-\widetilde{V}\phi + \widetilde{K}(u - u_0)\right) = -\left(-\frac{1}{2} + K'\right)\phi - W(u - u_0)$. Thus, (42) reads

$$a(u,v) - \langle \partial_n^e u^e\,, v\rangle_\Gamma = \langle f\,, v\rangle_{\Omega^{int}} + \langle \phi_0\,, v\rangle_\Gamma \quad \text{for all } v \in H^1(\Omega^{int}),$$

which is just the weak formulation of

$$-\Delta u = f \quad \text{in } \Omega^{int},$$

$$\partial_n u = \partial_n^e u^e + \phi_0 \quad \text{on } \Gamma.$$

The second equation of (40) is equivalent to

$$\langle u\,,\psi\rangle_\Gamma = \langle(\tfrac{1}{2}+K)u\,,\psi\rangle_\Gamma - \langle V\phi\,,\psi\rangle_\Gamma + \langle(\tfrac{1}{2}-K)u_0\,,\psi\rangle_\Gamma.$$

As in *Step 2* from the proof of Theorem 6.2, we obtain the jump condition (1c)

$$\langle u-u_0\,,\psi\rangle_\Gamma = \langle(\tfrac{1}{2}+K)(u-u_0)\,,\psi\rangle_\Gamma - \langle V\phi\,,\psi\rangle_\Gamma = \langle u^e\,,\psi\rangle_\Gamma \quad \text{for all } \psi\in H^{-1/2}(\Gamma).$$

With the fact that $-\Delta\left(\widetilde{V}\lambda+\widetilde{K}\omega\right)=0$ in $\mathbb{R}^2\backslash\Gamma$ for arbitrary $(\lambda,\omega)\in H^{-1/2}(\Gamma)\times H^1(\Omega^{int})$, we prove that $u^e=-\widetilde{V}\phi+\widetilde{K}(u-u_0)$ solves

$$\begin{array}{rcll} -\Delta u^e &=& 0 & \text{in } \Omega^{ext},\\ \gamma^e u^e &=& u-u_0 & \text{on } \Gamma,\\ \partial_n^e u^e &=& \phi-\phi_0 & \text{on } \Gamma. \end{array} \tag{43}$$

It remains to prove the radiation condition (1c). Choose $v=1$ in the first equation of (40) to see

$$\begin{aligned} -\langle\phi\,,1\rangle_\Gamma = \langle\phi\,,(K-\tfrac{1}{2})1\rangle_\Gamma + \langle u\,,W1\rangle_\Gamma &= \langle f\,,1\rangle_{\Omega^{int}} + \langle\phi_0\,,1\rangle_\Gamma + \langle u_0\,,W1\rangle_\Gamma\\ &= \langle f\,,1\rangle_{\Omega^{int}} + \langle\phi_0\,,1\rangle_\Gamma = 0. \end{aligned}$$

Theorem 3.10 and Theorem 3.9 state that u^e satisfies (1e). □

7.1 Discretization

As usual we choose finite-dimensional subspaces $X_\ell\subset H^1(\Omega^{int})$ and $Y_\ell\subset H^{-1/2}(\Gamma)$. The Galerkin formulation of (33) reads as follows: Find $(u_\ell,\phi_\ell)\in X_\ell\times Y_\ell$ such that for all $(v_\ell,\psi_\ell)\in X_\ell\times Y_\ell$

$$\boxed{\begin{array}{rcl} a(u_\ell,v_\ell)+\langle\left(K'-\frac{1}{2}\right)\phi_\ell\,,v_\ell\rangle_\Gamma+\langle Wu_\ell\,,v_\ell\rangle_\Gamma &=& \langle f\,,v_\ell\rangle_{\Omega^{int}}+\langle\phi_0+Wu_0\,,v_\ell\rangle_\Gamma\\ \langle\left(\frac{1}{2}-K\right)u_\ell\,,\psi_\ell\rangle_\Gamma+\langle V\phi_\ell\,,\psi_\ell\rangle_\Gamma &=& \langle\left(\frac{1}{2}-K\right)u_0\,,\psi_\ell\rangle_\Gamma \end{array}} \tag{44}$$

7.2 Stability Estimate for the Discrete and the Continuous Case

We choose a similar approach as in [6]. By bootstrapping the results in [6], we may simultaneously prove the discrete and the continuous case. We assume $X_\ell\subseteq H^1(\Omega^{int})$ and $Y_\ell\subseteq H^{-1/2}(\Gamma)$ to be closed subspaces. Note that the choices $X_\ell=H^1(\Omega^{int})$ and $Y_\ell=H^{-1/2}(\Gamma)$ are valid, too. Let $i_\ell: X_\ell\to H^1(\Omega^{int})$ and $j_\ell: Y_\ell\to H^{-1/2}(\Gamma)$ denote the canonical embeddings with their duals $i_\ell': \widetilde{H}^{-1}(\Omega^{int})\to X_\ell^*$ and $j_\ell': H^{1/2}(\Gamma)\to Y_\ell^*$. We define the discrete integral operators

$$V_\ell := j_\ell' V j_\ell,\quad K_\ell := j_\ell' K\gamma i_\ell,\quad W_\ell := i_\ell'\gamma' W\gamma i_\ell,\quad K_\ell' := i_\ell'\gamma' K' j_\ell. \tag{45}$$

Obviously, the discrete operators are all linear and bounded. Note that for $X_\ell=H^1(\Omega^{int})$ and $Y_\ell=H^{-1/2}(\Gamma)$, the operators coincide with the previously defined operators V, K, K', and W. There holds

$$\langle W_\ell v_\ell\,,v_\ell\rangle_\Gamma = \langle W\gamma i_\ell v_\ell\,,\gamma i_\ell v_\ell\rangle_\Gamma \geq 0 \quad \text{for all } v_\ell\in X_\ell.$$

Lemma 7.3. *The operators V_ℓ and V_ℓ^{-1} are uniformly elliptic, i.e.*

$$\langle V_\ell \phi_\ell \, , \phi_\ell \rangle_\Gamma \geq C_V \|\phi_\ell\|^2_{H^{-1/2}(\Gamma)} \quad \text{and} \quad \langle V_\ell^{-1} \psi_\ell \, , \psi_\ell \rangle_\Gamma \geq C \|\psi_\ell\|^2_{H^{1/2}(\Gamma)},$$

for all $(\phi_\ell, \psi_\ell) \in Y_\ell \times Y_\ell^*$, where $C > 0$ and $C_V > 0$ are independent of the chosen spaces X_ℓ and Y_ℓ.

Proof. The ellipticity of V_ℓ is a direct consequence of the ellipticity of V, i.e.

$$\langle V_\ell \phi_\ell \, , \phi_\ell \rangle_\Gamma = \langle V j_\ell \phi_\ell \, , j_\ell \phi_\ell \rangle_\Gamma \geq C_V \|\phi_\ell\|^2_{H^{-1/2}(\Gamma)} \quad \text{for all } \phi_\ell \in Y_\ell.$$

Now, the lemma of Lax-Milgram applies and we obtain that

$$\langle V_\ell \phi_\ell \, , \psi_\ell \rangle_\Gamma = \ell(\psi_\ell) \quad \text{for all } \psi_\ell \in Y_\ell^*,$$

is uniquely solvable for all $\ell \in Y_\ell$ and that V_ℓ is an isomorphism. Thus, for any $\phi_\ell \in Y_\ell$, one can find $\psi_\ell \in Y_\ell^*$ with $V_\ell \psi_\ell = \phi_\ell$. Hence, we get

$$\langle V_\ell^{-1} \phi_\ell \, , \phi_\ell \rangle_\Gamma = \langle \psi_\ell \, , V_\ell \psi_\ell \rangle_\Gamma \geq C_V \|\psi_\ell\|^2_{H^{-1/2}(\Gamma)}.$$

The operator norms of j_ℓ and its dual j_ℓ' are $1 = \|j_\ell\| = \|j_\ell'\|$, therefore

$$\begin{aligned} \langle V_\ell^{-1} \phi_\ell \, , \phi_\ell \rangle_\Gamma \geq C_V \|\psi_\ell\|^2_{H^{-1/2}(\Gamma)} &\geq \|V_\ell\|^{-2} C_V \|V_\ell \psi_\ell\|^2_{H^{-1/2}(\Gamma)} \\ &\geq (\|j_\ell\| \|V\| \|j_\ell'\|)^{-2} C_V \|\phi_\ell\|^2_{H^{-1/2}(\Gamma)} \\ &= C \|\phi_\ell\|_{H^{-1/2}(\Gamma)}. \end{aligned}$$

This proves the statement. □

Now, we may define the *Steclow-Poincare* operator.

Definition 7.4. *By $S_\ell : X_\ell \to X_\ell$, we denote*

$$S_\ell := W_\ell + (\tfrac{1}{2} I_\ell' - K_\ell') V_\ell^{-1} (\tfrac{1}{2} I_\ell - K_\ell),$$

where $I_\ell := j_\ell' \gamma i_\ell$. For convenience, we also define

$$S = W + (\tfrac{1}{2} - K') V^{-1} (\tfrac{1}{2} - K),$$

which corresponds to the definition of S_ℓ with $X_\ell = H^1(\Omega^{int})$ and $Y_\ell = H^{-1/2}(\Gamma)$.

Theorem 7.5. *There is a constant $c_0 > 0$, such that for any closed subspace $X_\ell \times Y_\ell \subseteq H^1(\Omega^{int}) \times H^{-1/2}(\Gamma)$ with $1 \in Y_\ell$ holds*

$$\langle S_\ell u_\ell \, , u_\ell \rangle_\Gamma \geq c_0 \|u_\ell\|^2_{H^{1/2}(\Gamma)} \quad \text{for all } u_\ell \in X_\ell.$$

Proof. We want to prove the following result

$$\exists c_0 > 0 : \forall (X_\ell, Y_\ell) \subseteq H^1(\Omega^{int}) \times H^{-1/2}(\Gamma) \text{ with } 1 \in Y_\ell : \forall u_\ell \in X_\ell : \langle S_\ell u_\ell \, , u_\ell \rangle_\Gamma \geq c_0 \|u_\ell\|^2_{H^{1/2}(\Gamma)}.$$

Assume that the conclusion is false, i.e.

$$\forall c > 0 : \exists (X_\ell, Y_\ell) \subseteq H^1(\Omega^{int}) \times H^{-1/2}(\Gamma) \text{ with } 1 \in Y_\ell : \exists u_\ell \in X_\ell : \langle S_\ell u_\ell \, , u_\ell \rangle_\Gamma < c \|u_\ell\|^2_{H^{1/2}(\Gamma)}.$$

Then, one can construct a sequence of functions $(u_{\ell_n})_{n \in \mathbb{N}}$ in $H^1(\Omega^{int})$ with

$$u_{\ell_n} \in X_{\ell_n}, \quad \langle S_{\ell_n} u_{\ell_n} \, , u_{\ell_n} \rangle_\Gamma \leq \tfrac{1}{n} \|u_{\ell_n}\|^2_{H^{1/2}(\Gamma)} \quad \text{for all } n \in \mathbb{N}.$$

Without loss of generality, we set $\|u_{\ell_n}\|_{H^{1/2}(\Gamma)} = 1$ and obtain

$$\langle S_{\ell_n} u_{\ell_n} \, , u_{\ell_n} \rangle_\Gamma \leq \tfrac{1}{n} \quad \text{for all } n \in \mathbb{N}.$$

With $\|\gamma u_{\ell_n}\|_{H^{1/2}(\Gamma)} \leq 1$, the *Banach-Alaoglu-Theorem* states the weak convergence of a subsequence of $(u_{\ell_n})_{n \in \mathbb{N}}$ towards some $w \in H^{1/2}(\Gamma)$. To avoid unhandy notation, we assume that we chose such a subsequence at the beginning, i.e.

$$\langle u_{\ell_n} \, , v \rangle_\Gamma \to \langle w \, , v \rangle_\Gamma \quad \text{for all } v \in H^{1/2}(\Gamma).$$

Now, by definition of S_ℓ we firstly conclude

$$\langle S_{\ell_n} u_{\ell_n} \, , u_{\ell_n} \rangle_\Gamma = \langle W_{\ell_n} u_{\ell_n} \, , u_{\ell_n} \rangle_\Gamma + \langle V_{\ell_n}^{-1} (\tfrac{1}{2} I_{\ell_n} - K_{\ell_n}) u_{\ell_n} \, , (\tfrac{1}{2} I_{\ell_n} - K_{\ell_n}) u_{\ell_n} \rangle_\Gamma.$$

The ellipticity of V_ℓ^{-1} and the semidefiniteness of W yield

$$0 \leftarrow \langle S_{\ell_n} u_{\ell_n} \, , u_{\ell_n} \rangle_\Gamma \geq \langle W_{\ell_n} u_{\ell_n} \, , u_{\ell_n} \rangle_\Gamma = \langle W u_{\ell_n} \, , u_{\ell_n} \rangle \geq 0.$$

Note that the functional

$$H^{1/2}(\Gamma) \to \mathbb{R}, u \mapsto \langle W u \, , u \rangle_\Gamma$$

is continuous and convex, whence weakly lower semicontinuous. This implies

$$0 \leq \langle W w \, , w \rangle \leq \liminf_{n \to \infty} \langle W u_{\ell_n} \, , u_{\ell_n} \rangle = 0.$$

i.e. w is constant as stated in Lemma 3.8. Let $u_{\ell_n} = v_n + w_n$ be a decomposition of u_{ℓ_n} with $v_n \in H^1_*(\Omega^{int})$ and $w_n \in \mathbb{R}$. Due to the $H^1_*(\Omega^{int})$-ellipticity of W, we obtain

$$\begin{aligned} 0 \leftarrow \langle W u_{\ell_n} \, , u_{\ell_n} \rangle_\Gamma &= \langle W v_n \, , v_n \rangle_\Gamma + \langle v_n \, , W w_n \rangle_\Gamma + \langle W w_n \, , v_n + w_n \rangle_\Gamma \\ &= \langle W v_n \, , v_n \rangle \geq C_W \|v_n\|^2_{H^{1/2}(\Gamma)}. \end{aligned}$$

Thus, $v_n \to 0$ strongly in $H^{1/2}(\Gamma)$. It follows

$$\langle v_{\ell_n} \, , 1 \rangle_\Gamma + w_n \langle 1 \, , 1 \rangle_\Gamma = \langle u_{\ell_n} \, , 1 \rangle_\Gamma \to w \langle 1 \, , 1 \rangle_\Gamma,$$

and with $v_n \to 0$ we obtain $w_n \to w$ in $\mathbb{R}$. Altogether, we have strong convergence of u_{ℓ_n} towards the constant w in $H^{1/2}(\Gamma)$. On the other hand, there holds

$$\begin{aligned} 0 \leftarrow \langle S_{\ell_n} u_{\ell_n}, u_{\ell_n} \rangle_\Gamma &\geq \langle V_{\ell_n}^{-1}(\tfrac{1}{2} I_{\ell_n} - K_{\ell_n}) u_{\ell_n}, (\tfrac{1}{2} I_{\ell_n} - K_{\ell_n}) u_{\ell_n} \rangle_\Gamma \\ &\geq C \| (\tfrac{1}{2} I_{\ell_n} - K_{\ell_n}) u_{\ell_n} \|_{H^{-1/2}(\Gamma)} \quad \text{for all } n \in \mathbb{N}, \end{aligned}$$

due to the semidefiniteness of W and the ellipticity of $V_{\ell_n}^{-1}$. Hence,

$$0 = \lim_{n\to\infty} \langle 1, (\tfrac{1}{2} I_{\ell_n} - K_{\ell_n}) u_{\ell_n} \rangle_\Gamma = \lim_{n\to\infty} \langle 1, \tfrac{1}{2} \gamma u_{\ell_n} - K \gamma u_{\ell_n} \rangle_\Gamma = \langle 1, \tfrac{1}{2} \gamma w - K \gamma w \rangle_\Gamma = w \langle 1, 1 \rangle,$$

i.e. $w = 0$. This contradicts $\|w\|_{H^{1/2}(\Gamma)} = \lim_{n\to\infty} \|u_{\ell_n}\|_{H^{1/2}(\Gamma)} = 1$.

□

Now, we are able to give a stability estimate for both, the discrete and the continuous case. With $(d_1, d_2) \in X_\ell^* \times Y_\ell^*$, we consider the following problem:
Find $(u_\ell, \phi_\ell) \in X_\ell \times Y_\ell$ such that

$$\begin{aligned} a(u_\ell, v_\ell) + \langle \big(K' - \tfrac{1}{2}\big) \phi_\ell, v_\ell \rangle_\Gamma + \langle W u_\ell, v_\ell \rangle_\Gamma &= d_1(v_\ell), \\ \langle \big(\tfrac{1}{2} - K\big) u_\ell, \psi_\ell \rangle_\Gamma + \langle V \phi_\ell, \psi_\ell \rangle_\Gamma &= d_2(\psi_\ell), \end{aligned} \tag{46}$$

for all $(v_\ell, \psi_\ell) \in X_\ell \times Y_\ell$.

Theorem 7.6. *For any choice of spaces $X_\ell \subseteq H^1(\Omega^{int})$ and $Y_\ell \subseteq H^{-1/2}(\Gamma)$ (finite of infinite dimensional) with $1 \in Y_\ell$, there exists a solution $(u_\ell, \phi_\ell) \in X_\ell \times Y_\ell$ of (46). There holds*

$$\|u_\ell\|_{H^1(\Omega^{int})} + \|\phi_\ell\|_{H^{-1/2}(\Gamma)} \leq C \left(\|d_1\|_{\widetilde{H}^{-1}(\Omega^{int})} + \|d_2\|_{H^{1/2}(\Gamma)} \right).$$

Proof. The second equation of (46) can be rearranged to

$$\phi_\ell = -V_\ell^{-1}(\tfrac{1}{2} I_\ell - K_\ell)(u_\ell - d_2) \quad \text{in } Y_\ell. \tag{47}$$

If we eliminate ϕ_ℓ in the first equation of (46), the problem reads

$$a(u_\ell, v_\ell) + \langle S_\ell u_\ell, v_\ell \rangle_\Gamma = L(v_\ell) \quad \text{for all } v_\ell \in X_\ell, \tag{48}$$

where $L(v_\ell) := d_1(v_\ell) + \langle (S_\ell - W_\ell) d_2, v_\ell \rangle_\Gamma \in X_\ell^* = X_\ell$. With Theorem 7.5, we have

$$a(u_\ell, u_\ell) + \langle S_\ell u_\ell, u_\ell \rangle_\Gamma \geq \|\nabla u_\ell\|_{L^2(\Omega^{int})}^2 + C \|u_\ell\|_{H^{1/2}(\Omega^{int})}^2 \quad \text{for all } u_\ell \in X_\ell,$$

where C doesn't depend on the choice of (X_ℓ, Y_ℓ). Due to Theorem 2.4, we see that the right hand side of the equation denotes an equivalent norm on $H^1(\Omega^{int})$, and thus is bounded below, i.e.

$$a(u_\ell, u_\ell) + \langle S_\ell u_\ell, u_\ell \rangle_\Gamma \geq C \|u_\ell\|_{H^1(\Omega^{int})}^2 \quad \text{for all } u_\ell \in X_\ell.$$

Now, the lemma of Lax-Milgram applies and states the unique solvability of (46). The solution u_ℓ satisfies

$$\|u_\ell\|_{H^1(\Omega^{int})} \leq C\|L\|_{\widetilde{H}^{-1}(\Omega^{int})} \leq C\left(\|d_1\|_{\widetilde{H}^{-1}(\Omega^{int})} + \|d_2\|_{H^{1/2}(\Gamma)}\right),$$

where $C > 0$ doesn't depend on the choice of (X_ℓ, Y_ℓ). The operator norms of i_ℓ, j_ℓ are $1 = \|i_\ell\| = \|j_\ell\|$, and so are their duals $1 = \|i'_\ell\| = \|j'_\ell\|$. Therefore, all operators in (47) are uniformly bounded. Finally, we obtain

$$\|\phi_\ell\|_{H^{-1/2}(\Omega^{int})} \leq C\left(\|d_1\|_{\widetilde{H}^{-1}(\Omega^{int})} + \|d_2\|_{H^{1/2}(\Gamma)}\right).$$

The combination of both estimates concludes the prove. □

Corollary 7.7. *For any data $(f, u_0, \phi_0) \in L^2(\Omega^{int}) \times H^{1/2}(\Gamma) \times H^{-1/2}(\Gamma)$, there exist solutions $(u_\ell, \phi_\ell) \in X_\ell \times Y_\ell$ of (44) and $(u, \phi) \in H^1(\Omega^{int}) \times H^{-1/2}(\Gamma)$ of (40). There holds*

$$\begin{aligned}\|u\|_{H^1(\Omega^{int})} + \|\phi\|_{H^{-1/2}(\Gamma)} &\leq C\left(\|f\|_{L^2(\Omega^{int})} + \|u_0\|_{H^{1/2}(\Gamma)} + \|\phi_0\|_{H^{-1/2}(\Gamma)}\right),\\ \|u_\ell\|_{H^1(\Omega^{int})} + \|\phi_\ell\|_{H^{-1/2}(\Gamma)} &\leq C\left(\|f\|_{L^2(\Omega^{int})} + \|u_0\|_{H^{1/2}(\Gamma)} + \|\phi_0\|_{H^{-1/2}(\Gamma)}\right),\end{aligned}$$

where $C > 0$ is independent of the choice of X_ℓ and Y_ℓ.

Proof. Choose $d_1 = \langle f, \cdot\rangle_{\Omega^{int}} + \langle \phi_0 + Wu_0, \cdot\rangle \in \widetilde{H}^{-1}(\Omega^{int})$ and $d_2 = \langle \left(\frac{1}{2} - K\right) u_0, \cdot\rangle_\Gamma \in H^{1/2}(\Gamma)$. With $X_\ell \times Y_\ell = H^1(\Omega^{int}) \times H^{-1/2}(\Gamma)$, the formulations (40) and (46) are equivalent. To see the equivalence of (46) and (44), consider X_ℓ and Y_ℓ being finite-dimensional and restrict d_1 and d_2 to X_ℓ resp. Y_ℓ. In both cases, the norms of d_1 and d_2 are bounded by

$$\begin{aligned}\|d_1\|_{\widetilde{H}^{-1}(\Omega^{int})} &\leq \|f\|_{L^2(\Omega^{int})} + \|\phi_0\|_{\widetilde{H}^{-1}(\Gamma)} + C_1\|u_0\|_{H^{1/2}(\Gamma)},\\ \|d_2\|_{H^{1/2}(\Gamma)} &\leq C_2\|u_0\|_{H^{1/2}(\Gamma)},\end{aligned}$$

where $C_2 > 0$ depends only on Γ and the continuity of K as well as $C_1 > 0$ depends only on the continuity of W. The statement follows by use of Theorem 7.6. □

7.3 Equivalence of Stability and Céa-Lemma

In view of numerical experiments, it is good to know some facts about convergence of the presented coupling methods. As a first step, we want to prove that the methods from Section 6 and Section 7 provide quasi best approximations of the exact solution in the sense of the *Céa-Lemma*.

Definition 7.8. *Let Z be a Hilbert space and let $a(\cdot,\cdot)$ denote a continuous bilinear form $a : Z \times Z \to \mathbb{R}$. For any finite-dimensional subspace $Z_\ell \subset Z$, we consider the induced bounded operators*

$$\begin{aligned}A : Z \to Z^*, &\qquad \langle Ax, y\rangle = a(x,y) \quad \text{for all } x,\, y \in Z,\\ A_\ell : Z_\ell \to Z_\ell^*, &\qquad \langle A_\ell x_\ell, y_\ell\rangle = a(x_\ell, y_\ell) \quad \text{for all } x_\ell,\, y_\ell \in Z_\ell.\end{aligned}$$

We seek for solutions z, z_ℓ of the variational formulations

$$\begin{aligned} a(z,y) &= f(y) \quad \text{for all } y \in Z, \\ a(z_\ell, y_\ell) &= f(y_\ell) \quad \text{for all } y_\ell \in Z_\ell, \end{aligned}$$

where $f \in Z^*$. We call z_ℓ the Galerkin approximation of z.

Lemma 7.9. *With the notation from Definition 7.8, the following two statements are equivalent*

- *Stability: A and A_ℓ are bijective and there holds*

$$\|A_\ell^{-1}\| \leq C,$$

  where $C > 0$ is independent of the choice of Z_ℓ.

- *Céa-Lemma: A is bijective and if there exists the Galerkin approximation z_ℓ for a closed subspace $Z_\ell \subset Z$, there holds*

$$\|z - z_\ell\|_Z \leq \widetilde{C} \inf_{x_\ell \in Z_\ell} \|z - x_\ell\|_Z,$$

  where $\widetilde{C} > 0$ is independent of the choice of Z_ℓ.

Proof. Step 1, Stability implies Céa-Lemma: The bijectivity of A and A_ℓ yield the existence of z and the respective Galerkin approximation z_ℓ. For arbitrary $x_\ell \in Z_\ell$ holds

$$\|z_\ell - x_\ell\|_Z = \|A_\ell^{-1} A_\ell (z_\ell - x_\ell)\|_Z \leq C \|A_\ell(z_\ell - x_\ell)\|_{Z^*} = C \sup_{y_\ell \in Z_\ell,\, \|y_\ell\|_Z = 1} a(z_\ell - x_\ell, y_\ell).$$

Fix $y_\ell \in Z_\ell$ with $\|y_\ell\|_Z = 1$, to see

$$a(z_\ell - x_\ell, y_\ell) = f(y_\ell) - a(x_\ell, y_\ell) = a(z - x_\ell, y_\ell) \leq \|a\| \|z - x_\ell\|_Z.$$

Consequently, we conclude

$$\|z_\ell - x_\ell\|_Z \leq C\|a\| \|z - x_\ell\|_Z.$$

The triangle inequality yields

$$\|z - z_\ell\|_Z \leq \|z - x_\ell\|_Z + \|z_\ell - x_\ell\|_Z \leq (1 + C\|a\|) \|z - x_\ell\|_Z.$$

Taking the infimum over all $x_\ell \in Z_\ell$ proves the *Céa-Lemma.*
Step 2, Céa-Lemma implies Stability: At first, we want to prove the bijectivity of A_ℓ. Let z_ℓ, $\widetilde{z}_\ell \in Z_\ell$ with $A_\ell z_\ell = A_\ell \widetilde{z}_\ell$. Then, there holds by definition

$$a(z_\ell, y_\ell) = \langle A_\ell z_\ell \,, y_\ell \rangle = \langle A_\ell \widetilde{z}_\ell \,, y_\ell \rangle = a(\widetilde{z}_\ell, y_\ell) = \langle A \widetilde{z}_\ell \,, y_\ell \rangle \quad \text{for all } y_\ell \in Z_\ell.$$

Thus, $z_\ell \in Z_\ell$ is a Galerkin approximation of $\widetilde{z}_\ell$. Therefore, the *Céa-Lemma* implies $\|z_\ell - \widetilde{z}_\ell\|_Z \leq \widetilde{C} \inf_{x_\ell \in Z_\ell} \|\widetilde{z}_\ell - x_\ell\|_Z = 0$. It is an elementary result from Linear Algebra that the injectivity of the finite-dimensional operator A_ℓ implies its bijectivity. Let $A_\ell z_\ell =: f_\ell \in Z_\ell^*$ and $f \in Z^*$ be an extension of f_ℓ with $\|f_\ell\|_{Z_\ell^*} = \|f\|_{Z^*}$, then

$$a(z_\ell, y_\ell) = f_\ell(y_\ell) = f(y_\ell) = a(A^{-1}f, y_\ell) \quad \text{for all } y_\ell \in Z_\ell.$$

Thus, z_ℓ is a Galerkin approximation of $z := A^{-1}f$ and therefore the *Céa-Lemma* states

$$\|z_\ell\|_Z \leq \|z_\ell - z\|_Z + \|z\|_Z \leq \widetilde{C}\|z - y_\ell\|_Z + \|z\|_Z \quad \text{for all } y_\ell \in Z_\ell.$$

Choose $y_\ell = 0$ to see

$$\begin{aligned}\|A_\ell^{-1} f_\ell\|_Z = \|z_\ell\|_Z &\leq (\widetilde{C}+1)\|z\|_Z = (\widetilde{C}+1)\|A^{-1}f\|_Z \leq (\widetilde{C}+1)\|A^{-1}\|\|f\|_{Z^*} \\ &= (\widetilde{C}+1)\|A^{-1}\|\|f_\ell\|_{Z_\ell^*},\end{aligned}$$

where the *open mapping theorem* implies the boundedness of A^{-1}. □

Let $X_\ell \subseteq H^1(\Omega^{int})$ and $Y_\ell \subseteq H^{-1/2}(\Gamma)$ denote finite-dimensional subspaces and let a_1 and a_2 denote the bilinear forms induced by (38) and (46), i.e.

$$a_1((u,v),(\phi,\psi)) = a(u,v) - \langle \phi, v\rangle_\Gamma + \langle \big(\tfrac{1}{2} - K\big)\, u, \psi\rangle_\Gamma + \langle V\phi, \psi\rangle_\Gamma, \tag{49}$$

$$a_2((u,v),(\phi,\psi)) = a(u,v) + \langle \big(K' - \tfrac{1}{2}\big)\, \phi, v\rangle_\Gamma + \langle Wu, v\rangle_\Gamma + \langle \big(\tfrac{1}{2} - K\big)\, u, \psi\rangle_\Gamma + \langle V\phi, \psi\rangle_\Gamma, \tag{50}$$

for all $(u,v),(\phi,\psi) \in H^1(\Omega^{int}) \times H^{-1/2}(\Gamma)$. For $(d_1,d_2) \in \widetilde{H}^{-1}(\Omega^{int}) \times H^{1/2}(\Gamma)$, we define the solutions $(u^1,\phi^1) \in H^1(\Omega^{int}) \times H^{-1/2}(\Gamma)$ and $(u_\ell^1,\phi_\ell^1) \in X_\ell \times Y_\ell$ as well as $(u^2,\phi^2) \in H^1(\Omega^{int}) \times H^{-1/2}(\Gamma)$ and $(u_\ell^2,\phi_\ell^2) \in X_\ell \times Y_\ell$ by

$$\begin{aligned}
a_1((u^1,\phi^1),(v,\psi)) &= d_1(v) + d_2(\psi) && \text{for all } (v,\psi) \in H^1(\Omega^{int}) \times H^{-1/2}(\Gamma) \\
a_1((u_\ell^1,\phi_\ell^1),(v_\ell,\psi_\ell)) &= d_1(v_\ell) + d_2(\psi_\ell) && \text{for all } (v_\ell,\psi_\ell) \in X_\ell \times Y_\ell \\
a_2((u^2,\phi^2),(v,\psi)) &= d_1(v) + d_2(\psi) && \text{for all } (v,\psi) \in H^1(\Omega^{int}) \times H^{-1/2}(\Gamma) \\
a_2((u_\ell^2,\phi_\ell^2),(v_\ell,\psi_\ell)) &= d_1(v_\ell) + d_2(\psi_\ell) && \text{for all } (v_\ell,\psi_\ell) \in X_\ell \times Y_\ell
\end{aligned}$$

Note that Theorem 6.3 and Theorem 7.6 state the existence of these solutions.

Theorem 7.10. *With the notation from above holds*

$$\|u^i - u_\ell^i\|_{H^1(\Omega^{int})} + \|\phi^i - \phi_\ell^i\|_{H^{-1/2}(\Gamma)} \leq C \inf_{(v_\ell,\psi_\ell)\in X_\ell \times Y_\ell} \left(\|u^i - v_\ell\|_{H^1(\Omega^{int})} + \|\phi^i - \psi_\ell\|_{H^{-1/2}(\Gamma)}\right),$$

for $i = 1, 2$. The constant $C > 0$ is independent of the choice of X_ℓ and Y_ℓ.

Proof. We consider the Hilbert space $Z := H^1(\Omega^{int}) \times H^{-1/2}(\Gamma)$ and the subspaces $Z_\ell := X_\ell \times Y_\ell$. The bilinear forms in (49) induce the operators A and A_ℓ as in Definition 7.8. Recall that Theorem 6.3 and Theorem 7.6 state that *stability* holds for both methods from Section 6 and Section 7. Now, Lemma 7.9 proves the statement. □

8 A Posteriori Error Estimates

8.1 Residual-Based Error Estimates

In this section, we want to present a residual-based *a posteriori error estimator*, proposed by C. Carstensen and E. Stephan in [6]. Before we can proceed, we need some notation. We define the open convex hull of points $x_1, \dots, x_n \in \mathbb{R}^2$ by

$$\operatorname{conv}(x_1, \dots, x_n) = \{\sum_{j=1}^{n} a_j x_j \,:\, 0 < a_j < 1\,, \sum_{j=1}^{n} a_j = 1\}.$$

Definition 8.1. *Let Ω^{int} be as in (1) but additionally with polygonal boundary Γ. We consider the family $\mathcal{T} := (\mathcal{T}_\ell)_{\ell \in I}$ of triangulations of Ω^{int}, where*

$$\mathcal{T}_\ell = \{T_1, \dots, T_{N_\ell}\},$$

and T_i is an open triangle, i.e there are nodes $V_i^1, V_i^2, V_i^3 \in \mathbb{R}^2$ such that

$$T_i = \operatorname{conv}(V_i^1, V_i^2, V_i^3).$$

We define the edges E_i^j, $j = 1, 2, 3$ of T_i by

$$E_i^1 = \operatorname{conv}(V_i^1, V_i^2) \quad E_i^2 = \operatorname{conv}(V_i^2, V_i^3) \quad E_i^3 = \operatorname{conv}(V_i^3, V_i^1).$$

Additionally, we claim

$$T_i \cap T_j = \emptyset,\ i \neq j, \quad \bigcup_{j=1}^{N_\ell} \overline{T_j} = \overline{\Omega^{int}}.$$

For two triangles T_i, $T_j \in \mathcal{T}_\ell$ with $T_i \neq T_j$ holds exactly one of the following conditions:

- *T_i and T_j are not neighboured, i.e. $\overline{T_i} \cap \overline{T_j} = \emptyset$.*
- *T_i and T_j share a common edge, i.e $\partial T_i \cap \partial T_j = E_i^k \cap E_j^{k'}$, $k, k' \in \{1, 2, 3\}$.*
- *T_i and T_j share a common node, i.e $\partial T_i \cap \partial T_j = V_i^k \cap V_j^{k'}$, $k, k' \in \{1, 2, 3\}$.*

We define

$$\mathcal{S}_\ell := \{E_i^j \,:\, i = 1, \dots, N_\ell,\ j = 1, 2, 3\}.$$

Let

$$\mathcal{E}_\ell := \{E \in \mathcal{S}_\ell \,:\, E \subseteq \Gamma\}$$

be the set of boundary sides and let

$$\mathcal{S}_\ell^0 := \mathcal{S}_\ell \setminus \mathcal{E}_\ell$$

denote the set of interior sides. We define the shape regularity constant by

$$\sigma(\mathcal{T}_\ell) := \max_{T_i \in \mathcal{T}_\ell} \frac{\operatorname{diam}(T_i)^2}{|T_i|}.$$

For any $\ell \in I$, the triangulation $\mathcal{T}_\ell$ induces spaces X_ℓ and Y_ℓ

$$X_\ell := \{\eta_\ell \in C(\Omega^{int}) \,:\, \eta_\ell|_{T_i} \in \mathcal{P}_1 \text{ for any } T_i \in \mathcal{T}_\ell\} =: \mathcal{S}^1(\mathcal{T}_\ell),$$

$$Y_\ell := \{\xi_\ell \in L^\infty(\Omega^{int}) \,:\, \xi_\ell|_E \in \mathcal{P}_0 \text{ for any } E \in \mathcal{E}_\ell\} =: \mathcal{P}^0(\mathcal{E}_\ell).$$

By ∂_n and ∂_n^e, we denote in the following the interior and exterior normal derivatives on any element boundary. More precisely, let $E \in \mathcal{S}_\ell$ denote an edge of $T_i \in \mathcal{T}_\ell$. Then, there holds

$$\partial_n u_\ell(x_0) = \lim_{x \in T_i,\, x \to x_0} \nabla u_\ell(x) \cdot n, \quad \partial_n^e u_\ell(x_0) = \lim_{x \in \overline{T_i}^c,\, x \to x_0} \nabla u_\ell(x) \cdot n, \quad x_0 \in E,$$

where n is the outward pointing normal vector on E w.r.t. T_i. By $[\partial_n \cdot] = \partial_n - \partial_n^e$, we denote the jump over the side of a triangle. We will also need the local mesh size function $h_\ell : \overline{\Omega^{int}} \to \mathbb{R}$, where $h(x) = \operatorname{diam}(T_i)$ for $x \in T_i \in \mathcal{T}_\ell$ and $h(x) = \operatorname{diam}(E)$ for $x \in E \in \mathcal{S}_\ell$. Define the quantities

$$\begin{aligned}
R_1^2 &:= \sum_{\Delta \in \mathcal{T}_\ell} \operatorname{diam}(\Delta)^2 \int_\Delta |f|^2 \,\mathrm{dx} = \|h_\ell f\|^2_{L^2(\Omega^{int})},\\
R_2^2 &:= \sum_{E \in \mathcal{S}_\ell^0} \operatorname{diam}(E) \int_E [\partial_n u_\ell]^2 \,\mathrm{d}\Gamma = \|h_\ell^{1/2} [\partial_n u_\ell]\|^2_{L^2(\bigcup \mathcal{S}_\ell^0)}\\
R_3 &:= \|h_\ell^{1/2} \left(\phi_0 - \partial_n u_\ell + W(u_0 - u_\ell) - (K' - \tfrac{1}{2})\phi_\ell\right) \|_{L^2(\Gamma)},\\
R_4 &:= \|h_\ell^{1/2} \left((\tfrac{1}{2} - K)(u_0 - u_\ell) - V\phi_\ell\right)' \|_{L^2(\Gamma)}.
\end{aligned}$$

There holds the following *a posteriori* estimate for the solutions $(u, \phi) \in H^1(\Omega^{int}) \times H^{-1/2}(\Gamma)$ of (40) and $(u_\ell, \phi_\ell) \in X_\ell \times Y_\ell$ of (44).

Theorem 8.2. *There exists a constant $C > 0$, such that*

$$\|u - u_\ell\|_{H^1(\Omega^{int})} + \|\phi - \phi_\ell\|_{H^{-1/2}(\Gamma)} \le C \cdot (R_1 + R_2 + R_3 + R_4),$$

where $C > 0$ depends on Ω^{int} and the shape regularity constant $\sigma(\mathcal{T}_\ell)$.

The proof of Theorem 8.2 is split into several lemmas. The original paper [6] gives the proof for a more general model problem. If there are no simplifications due to the different conditions, we will refer to that paper. Throughout this section, we use

$$e := u - u_\ell, \quad \varepsilon := \phi - \phi_\ell, \quad \delta := \tfrac{1}{2}\left(\varepsilon + V^{-1}(\tfrac{1}{2} - K)e\right) \in H^{-1/2}(\Gamma),$$

as well as

$$\|(u_\ell, \phi_\ell)\|_\times := \|u_\ell\|_{H^1(\Omega^{int})} + \|\phi_\ell\|_{H^{-1/2}(\Gamma)} \quad \text{for all } (u_\ell, \phi_\ell) \in H^1(\Omega^{int}) \times H^{-1/2}(\Gamma).$$

Lemma 8.3. *For some constant $\beta > 0$ holds*

$$\beta \|(e,\varepsilon)\|_\times \|(e,\delta)\|_\times \leq T_1 + T_2 + T_3 + T_4,$$

where, for any $(v_\ell, \psi_\ell) \in X_\ell \times Y_\ell$,

$$\begin{aligned} T_1 &:= \sum_{\Delta \in \mathcal{T}_\ell} \int_\Delta f(e - v_\ell)\,\mathrm{dx}, \\ T_2 &:= -\sum_{E \in \mathcal{S}_\ell^0} \int [\partial_n u_\ell (e - v_\ell)]\,\mathrm{d}\Gamma, \\ T_3 &:= \langle \phi_0 - \partial_n u_\ell + W(u_0 - u_\ell) - (K' - \tfrac{1}{2})\phi_\ell \,, e - v_\ell \rangle_\Gamma, \\ T_4 &:= \langle \delta - \psi_\ell \,, (\tfrac{1}{2} - K)(u_0 - u_\ell) - V\phi_\ell \rangle_\Gamma. \end{aligned}$$

Proof. Step 1: Recall the bilinear form a_2 in (49), to rewrite the symmetric coupling (40) as

$$\begin{aligned} a_2((u,\phi)(v,\psi)) &= a(u,v) + \langle (K' - \tfrac{1}{2})\phi \,, v\rangle_\Gamma + \langle Wu \,, v\rangle_\Gamma + \langle (\tfrac{1}{2} - K)u \,, \psi\rangle_\Gamma + \langle V\phi \,, \psi\rangle_\Gamma \\ &= L(v,\psi) \end{aligned} \tag{51}$$

with the respective right-hand side

$$L(v,\psi) := \langle f \,, v\rangle_{\Omega^{int}} + \langle \phi_0 + Wu_0 \,, v\rangle_\Gamma + \langle (\tfrac{1}{2} - K)u_0 \,, \psi\rangle_\Gamma.$$

Now, we want to prove a lower bound for the following term

$$a_2((e,\varepsilon)(e,\delta)) := a(e,e) + \langle (K' - \tfrac{1}{2})\varepsilon \,, e\rangle_\Gamma + \langle We \,, e\rangle_\Gamma + \langle (\tfrac{1}{2} - K)e \,, \delta\rangle_\Gamma + \langle V\varepsilon \,, \delta\rangle_\Gamma. \tag{52}$$

By definition of δ, this is equivalently written as

$$\begin{aligned} a_2((e,\varepsilon)(e,\delta)) = a(e,e) + \langle (K' - \tfrac{1}{2})\varepsilon \,, e\rangle_\Gamma + \langle We \,, e\rangle_\Gamma + \tfrac{1}{2}\langle e \,, (\tfrac{1}{2} - K')\varepsilon\rangle_\Gamma \\ + \tfrac{1}{2}\langle e \,, (\tfrac{1}{2} - K')V^{-1}(\tfrac{1}{2} - K)e\rangle_\Gamma + \tfrac{1}{2}\langle V\varepsilon \,, \varepsilon + V^{-1}(\tfrac{1}{2} - K)e\rangle_\Gamma \end{aligned}$$

By use of the *Steclow-Poincare* operator S from Lemma 7.4, this reads

$$a_2((e,\varepsilon)(e,\delta)) = a(e,e) + \tfrac{1}{2}\langle We \,, e\rangle_\Gamma + \tfrac{1}{2}\langle Se \,, e\rangle + \tfrac{1}{2}\langle V\varepsilon \,, \varepsilon\rangle_\Gamma.$$

Due to Theorem 2.4, $a(e,e) + \frac{1}{2}\langle Se \,, e\rangle_\Gamma$ is bounded below by $C\|e\|^2_{H^1(\Omega^{int})}$ for some $C > 0$. With the ellipticity of V^{-1} and the semidefiniteness of W, we obtain that $a_2((e,\varepsilon)(e,\delta))$ is bounded below

$$a_2((e,\varepsilon)(e,\delta)) \geq C(\|e\|^2_{H^1(\Omega^{int})} + \|\varepsilon\|^2_{H^{-1/2}(\Gamma)}) \geq \tfrac{C}{2}\|(e,\varepsilon)\|^2_\times.$$

With $\|\delta\|_{H^{-1/2}(\Gamma)} \leq \widetilde{C}(\|\varepsilon\|_{H^{-1/2}(\Gamma)} + \|e\|_{H^1(\Omega^{int})}) = \widetilde{C}\|(e,\varepsilon)\|_\times$, where $\widetilde{C}$ depends on the continuity of V^{-1} and K, there holds $\|(e,\delta)\|_\times \leq \beta^{-1}\|(e,\varepsilon)\|_\times$ for some $\beta > 0$. Consequently,

$$a_2((e,\varepsilon)(e,\delta)) \geq \tfrac{C}{2}\|(e,\varepsilon)\|^2_\times \geq \tfrac{C}{2}\beta\|(e,\varepsilon)\|_\times \|(e,\delta)\|_\times.$$

Step 2: There holds the *Galerkin-Orthogonality*

$$a_2((e,\varepsilon)(v_\ell,\psi_\ell)) = 0 \quad \text{for all } (v_\ell,\psi_\ell) \in X_\ell \times Y_\ell,$$

thus (51) can be rearranged to

$$a_2((e,\varepsilon)(e,\delta)) = L(e-v_\ell,\delta-\psi_\ell) - a_2((u_\ell,\phi_\ell)(e-v_\ell,\delta-\psi_\ell)).$$

By definition of a_2 and L, this expression is equal to

$$\begin{array}{rl} \displaystyle\int_{\Omega^{int}} & f(e-v_\ell) - \nabla u_\ell \cdot \nabla(e-v_\ell)\,\mathrm{dx} \\ + & \langle \phi_0 + W(u_0-u_\ell) - (K'-\frac{1}{2})\phi_\ell\,, e-v_\ell\rangle_\Gamma \\ + & \langle (\frac{1}{2}-K)(u_0-u_\ell) - V\phi_\ell\,, \delta-\psi_\ell\rangle_\Gamma. \end{array} \tag{53}$$

Using *Green's formula* on each element $T_i \in \mathcal{T}_\ell$, we obtain

$$\begin{array}{rl} -\displaystyle\int_{\Omega^{int}} \nabla u_\ell\,\nabla(e-v_\ell)\,\mathrm{dx} = & -\displaystyle\sum_{E\in\mathcal{S}_\ell^0}\int_E [\partial_n u_\ell](e-v_\ell)\,\mathrm{d\Gamma} \\ & -\langle \partial_n u_\ell\,, e-v_\ell\rangle_\Gamma. \end{array} \tag{54}$$

The combination of (53) and (54) leads us to

$$L(e-v_\ell,\delta-\psi_\ell) - a_2((u_\ell,\phi_\ell),(e-v_\ell,\delta-\psi_\ell)) = T_1+T_2+T_3+T_4,$$

which concludes the proof. □

Now, we define the *Clément-Operator* I_ℓ. For further information on I_ℓ see e.g. [7] or [11]. The following lemma is taken from [7] and states the property of I_ℓ which makes the analysis work.

Lemma 8.4. *Let $I_\ell : H^1(\Omega^{int}) \to X_\ell$ denote the Clément-Operator for all $\ell \in I$. With $C>0$, the following holds. For any $T_i \in \mathcal{T}_\ell$ and integers k, q with $0 \le k \le q \le 2$,*

$$|I_\ell u - u|^2_{H^k(T_i)} \le C \cdot \mathrm{diam}(T_i)^{2(q-k)} |u|^2_{H^q(N)},$$

where $N := \bigcup\{T_j \in \mathcal{T}_\ell : \overline{T_j} \cap \overline{T_i} \neq \emptyset\}$ is the set of all neighbor elements.

For the proof of the following Lemma, we refer to [6, Lemma 8, 10, 11].

Lemma 8.5. *By choosing $v_\ell = I_\ell e$, we obtain*

$$\begin{array}{rcl} T_1 & \le & C\|\nabla e\|_{L^2(\Omega^{int})} R_1, \\ T_2 & \le & C\|\nabla e\|_{L^2(\Omega^{int})} R_2, \\ T_3 & \le & C\|\nabla e\|_{L^2(\Omega^{int})} R_3, \end{array}$$

where C only depends on $\mathcal{T}$.

Lemma 8.6. *For* $r := (\frac{1}{2} - K)(u_0 - u_\ell) - V\phi_\ell$ *we have*

$$\|r\|_{H^{1/2}(\Gamma)} \leq C\|h_\ell^{1/2} r'\|_{L^2(\Gamma)} = CR_4. \tag{55}$$

This yields

$$T_4 \leq \|\delta - \psi_\ell\|_{H^{-1/2}(\Gamma)} \|r\|_{H^{1/2}(\Gamma)} \leq C\|\delta - \psi_\ell\|_{H^{-1/2}(\Gamma)} R_4. \tag{56}$$

Proof. The estimate in (55) is proved in [4]. The conclusion (56) is an obvious consequence of the definition of T_4. □

Proof of Theorem 8.2. By use of Lemma 8.3, Lemma 8.5, and Lemma 8.6, we obtain

$$\begin{aligned} \beta\|(u - u_\ell, \phi - \phi_\ell)\|_\times \|(e, \delta)\|_\times &\leq T_1 + T_2 + T_3 + T_4 \\ &\leq C\|\nabla e\|_{L^2(\Omega^{int})}(R_1 + R_2 + R_3) + C\|\delta - \psi_\ell\|_{H^{-1/2}(\Gamma)} R_4 \\ &\leq C\|e\|_{H^1(\Omega^{int})}(R_1 + R_2 + R_3) + C\|\delta - \psi_\ell\|_{H^{-1/2}(\Gamma)} R_4. \end{aligned}$$

If we set $\psi_\ell = 0$, there holds

$$\|(u - u_\ell, \phi - \phi_\ell)\|_\times \|(e, \delta)\|_\times \leq \tilde{C}\|(e, \delta)\|_\times (R_1 + R_2 + R_3 + R_4).$$

Division by $\|(e, \delta)\|_\times$ concludes the proof. □

8.2 Error Estimates Based on the $H - H/2$ Strategy

A rather classical approach to derive an a posteriori estimate is presented in this section. As in the previous section, we define the Triangulation $\mathcal{T}_\ell$, the set of edges $\mathcal{S}_\ell$, the triangulation of the boundary $\mathcal{E}_\ell$ as well as the induced spaces X_ℓ and Y_ℓ, cf. Definition 8.1. Additionally, we consider the triangulation $\widehat{\mathcal{T}}_\ell$ which is generated from $\mathcal{T}_\ell$ by uniform refinement (bisec_3), see Section 11.1 for details. Analogously, $\widehat{\mathcal{T}}_\ell$ induces the spaces $\widehat{X}_\ell$ and $\widehat{Y}_\ell$. There holds

$$X_\ell \subset \widehat{X}_\ell \quad \text{and} \quad Y_\ell \subset \widehat{Y}_\ell.$$

Let $(u_\ell, \phi_\ell) \in X_\ell \times Y_\ell$ and $(\widehat{u}_\ell, \widehat{\phi}_\ell) \in \widehat{X}_\ell \times \widehat{Y}_\ell$ denote solutions of (44) with respect to the same data $(f, u_{0,\ell}, \phi_0) \in H^1(\Omega^{int}) \times H^{1/2}(\Gamma) \times H^{-1/2}(\Gamma)$ but with different discrete spaces. Here, $I_\ell^{\mathcal{T}} : C(\overline{\Omega^{int}}) \to \mathcal{S}^1(\mathcal{T}_\ell)$ denotes the nodal interpolation operator w.r.t. the "coarse" mesh $\mathcal{T}_\ell$ and $u_{0,\ell} = I_\ell u_0$. The approximation of u_ℓ is mandatory, because the involved right-hand side terms $\langle W u_0, v_\ell\rangle_\Gamma$ and $\langle K u_0, \psi\rangle_\Gamma$ can hardly be computed analytically. Now, we define the error estimator μ_ℓ as follows

$$\mu_\ell := \Big(\|(1 - I_\ell^{\mathcal{T}})\widehat{u}_\ell\|^2_{H^1(\Omega^{int})} + \|h_\ell^{1/2}(1 - \Pi_\ell)\widehat{\phi}_\ell\|^2_{L^2(\Gamma)}\Big)^{1/2}. \tag{57}$$

By $\Pi_\ell : L^2(\Gamma) \to \mathcal{P}^0(\mathcal{E}_\ell)$, we denote the L^2-orthogonal projection. Further, we define the data oscillations osc_ℓ as

$$\text{osc}_\ell := \|h_\ell^{1/2}(u_0 - u_{0,\ell})'\|_{L^2(\Gamma)},$$

where $(\cdot)'$ denotes the arclength derivative along Γ. The data oscillation term allows us to control the error, which arises from approximating the dirichlet data u_0 by its nodal interpolant $u_{0,\ell}$. Let $(u_\ell^*, \phi_\ell^*) \in X_\ell \times Y_\ell$ be the non perturbed solution of (44) with respect to the data (f, u_0, ϕ_0). Then, we obtain the following result.

Lemma 8.7. *There holds the approximation error estimate*

$$C_1 \left(\|u_\ell^* - u_\ell\|_{H^1(\Omega^{int})} + \|\phi_\ell^* - \phi_\ell\|_{H^{-1/2}(\Gamma)} \right) \leq \|u - u_\ell\|_{H^1(\Omega^{int})} + \|\phi - \phi_\ell\|_{H^{-1/2}(\Gamma)} \leq C_2 \mathrm{osc}_\ell,$$

where $(u, \phi) \in H^1(\Omega^{int}) \times H^{-1/2}(\Gamma)$ *denotes the exact solution of* (40)*.*

For a proof, we refer to [2, Proposition 1]. Parts of the analysis need the following assumption to work:
Saturation Assumption: There exists a constant $0 < q < 1$ which may only depend on $\mathcal{T}_0$ such that

$$\|u - \widehat{u}_\ell\|_{H^1(\Omega^{int})} + \|\phi - \widehat{\phi}_\ell\|_{H^{-1/2}(\Gamma)} \leq q \left(\|u - u_\ell\|_{H^1(\Omega^{int})} + \|\phi - \phi_\ell\|_{H^{-1/2}(\Gamma)} \right). \tag{58}$$

With some minor restrictions, the *saturation assumption* can be proved for the *finite element method*, however, there are no results available for the *boundary element method* or for *FEM-BEM coupling*. Many numerical experiments underline the *saturation assumption*, cf. Section 12.

Theorem 8.8. *The error estimator* μ_ℓ *is efficient, i.e. there holds*

$$\mu_\ell \leq C(\|u - u_\ell\|_{H^1(\Omega^{int})} + \|\phi - \phi_\ell\|_{H^{-1/2}(\Gamma)} + \mathrm{osc}_\ell),$$

where $C > 0$ *is independent of the choice of* $\mathcal{T}_\ell$*. Under the saturation assumption* (58)*, we can also prove reliability, i.e.*

$$\|u - u_\ell\|_{H^1(\Omega^{int})} + \|\phi - \phi_\ell\|_{H^{-1/2}(\Gamma)} \leq C(\mu_\ell + \mathrm{osc}_\ell).$$

A prove of this result can be found in [2].

9 Overview on the Implementation

The following implementation of the coupling methods from Section 6 and Section 7 allows to consider a more general model problem as (1).

$$\begin{aligned} -\mathrm{div}(A\nabla u^i) &= f && \text{in } \Omega^{int}, \\ -\Delta u^e &= 0 && \text{in } \Omega^{ext}, \\ u^i - u^e &= u_0 && \text{on } \Gamma, \\ (A\nabla u^i - \nabla u^e)\cdot n &= \phi_0 && \text{on } \Gamma, \\ u^e(x) &= \mathcal{O}(\log|x|) && \text{as } |x| \to \infty. \end{aligned} \tag{59}$$

As in Section 1, Ω^{int} is a bounded Lipschitz domain in $\mathbb{R}^2$ with boundary $\Gamma := \partial\Omega$ and outer unit normal vector n. The given data satisfy $f \in L^2(\Omega^{int})$, $u_0 \in H^{1/2}(\Gamma)$, and $\phi_0 \in H^{-1/2}(\Gamma)$. The implementation of the operator $A : (L^2(\Omega^{int}))^2 \to (L^2(\Omega^{int}))^2$ allows the following situations.

- *Linear Case:* A is the identity.
- *Nonlinear Case:* A is of the form

  $$A(x) = \rho(|x|)x \quad x \in \mathbb{R}^2,$$

  where $\rho : \mathbb{R} \to \mathbb{R}$.

The directories of the implementation are organized as follows:

```
FEMBEM/
    code/
    input/
    output/
    postproc/
    setup_scripts/
```

`code/` contains the source code of all functions, `input/` contains the input data for the problems to be computed. The results are stored in `output/`. `setup_scripts/` contains some scripts which allow to perform some numerical examples. The results of the so called post-processing, i.e. plotting the approximation error, or the computational time, are stored in `postproc/`.

10 Preliminaries

10.1 Triangulation

As in Section 8, we assume the domain Ω^{int} to be a polygon. By $\mathcal{T}_\ell$, we denote a particular triangulation of Ω^{int}, and by $\mathcal{E}_\ell$ we denote its restriction to the boundary, cf. Definition 8.1. We call the pair $(\mathcal{T}_\ell, \mathcal{E}_\ell)$ a mesh. By $(\widehat{\mathcal{T}}_\ell, \widehat{\mathcal{E}}_\ell)$, we denote the uniform refinement of the mesh $(\mathcal{T}_\ell, \mathcal{E}_\ell)$. The uniform refinement is done via bisec_3 as explained in Section 11.1.

10.2 Discrete Integral Operators

In Section 7.2, we defined the discrete operators V_ℓ, K_ℓ, K'_ℓ, and W_ℓ. These operators are used to compute the numerical approximation. The corresponding matrices (e.g. used in Section 11.2) are computed by the functions

buildV
buildK
buildW

These functions are part of the MATLAB BEM library `HILBERT`, cf. [1].

10.3 Discrete Spaces

The numerical solutions (u_ℓ, ϕ_ℓ) of the computations belong to the Hilbert space $H^1(\Omega^{int}) \times H^{-1/2}(\Gamma)$. As discrete subspaces X_ℓ and Y_ℓ, we choose

$$X_\ell := \mathcal{S}^1(\mathcal{T}_\ell) \quad \text{and} \quad Y_\ell := \mathcal{P}^0(\mathcal{E}_\ell).$$

We choose the canonical basis functions $(\eta_i)_{i=1}^n$ of $\mathcal{S}^1(\mathcal{T}_\ell)$ and $(\xi_i)_{i=1}^m$ of $\mathcal{P}^0(\mathcal{E}_\ell)$, i.e.

$$\eta_i(V_j) = \delta_{ij} \quad \text{and} \quad \xi_k(x) = \begin{cases} 1 & , x \in \tau_k \\ 0 & , x \in \mathcal{E}_\ell \setminus \tau_k \end{cases} \quad \text{for all } 1 \le i, j \le n,\ 1 \le k \le m,$$

where $(V_j)_{j=1}^n$ are the nodes of $\mathcal{T}_\ell$ and $(\tau_k)_{k=1}^m$ are the elements of $\mathcal{E}_\ell$.

10.4 Data Structures

10.4.1 The Arrays `coordinates`, `elements`, and `dirichlet`

The arrays `coordinates`, `elements`, and `dirichlet` represent the given meshes $\mathcal{T}_\ell$ and $\mathcal{E}_\ell$. The `size` of the arrays is denoted by

```
size(coordinates) = [nC,2],
   size(elements) = [nE,3],
  size(dirichlet) = [nD,2].
```

`coordinates(j,:)` contains the coordinates of the `j`-th node V_j of $\mathcal{T}_\ell$, `elements(s,:)` contains the numbers of the three nodes of $T_s \in \mathcal{T}_\ell$ and `dirichlet(s,:)` contains the numbers of the two endpoints of $E_s \in \mathcal{E}_\ell$. Particularly, `coordinates(elements(s,:),:)` is a 3×2-array which contains the coordinates of the three nodes of the s-th volume element $T_s \in \mathcal{T}_\ell$, as well as `coordinates(dirichlet(s,:),:)` is a 2×2-array which contains the coordinates of the two endpoints of the s-th boundary element $E_s \in \mathcal{E}_\ell$.

10.4.2 The Array x

After sorting the arrays `elements`, `dirichlet`, and `coordinates` with the function `sortBoundaryNodesLast`, the $(\mathtt{nC}+\mathtt{nD})\times 1$-array `x` represents the solution $(u_\ell, \phi_\ell) \in X_\ell \times Y_\ell$ in the following way

$$u_\ell = \sum_{i=1}^{\mathtt{nC}} \mathtt{x(i)}\eta_i \quad \text{and} \quad \phi_\ell = \sum_{i=\mathtt{nC}+1}^{\mathtt{nC}+\mathtt{nD}} \mathtt{x(i)}\xi_{i-\mathtt{nC}}. \tag{60}$$

11 Implementation

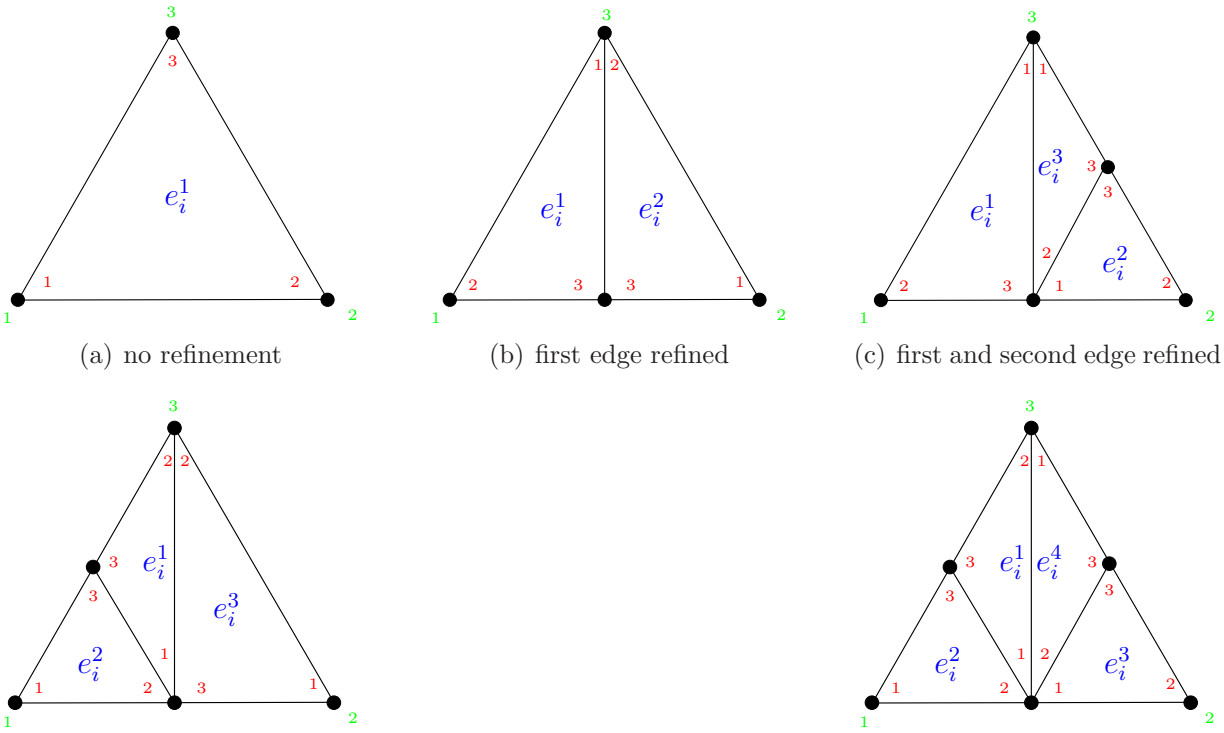
11.1 Local Mesh Refinement

The function `refineMesh` provides uniform and adaptive mesh refinement of the volume mesh $\mathcal{T}_\ell$ and the boundary mesh $\mathcal{E}_\ell$. The uniform refinement is done via bisec_3 refinement on the volume mesh and via bisection of the elements on the boundary mesh. If requested, the arrays `father2vol` and `father2dir` are provided. The `nC`$\times 4$-array `father2vol` links the refined elements represented by `elements_fine` and the original elements represented by `elements` as well as their nodes. We assume that the i-th element ($1 \le i \le$ `nC`) is refined with:

- *No refinement:* There holds, `father2vol(i,:)`$=(e_i^1, e_i^1, e_i^1, e_i^1)$, where $1 \le e_i^1 \le$ `size(elements_fine,1)` is the number of the new (=old) element.
- *First edge refined:* There holds, `father2vol(i,:)`$=(e_i^1, e_i^1, e_i^2, e_i^2)$, where $1 \le e_i^1, e_i^2 \le$ `size(elements_fine,1)` are the numbers of the new elements.
- *First and second edge refined:* There holds, `father2vol(i,:)`$=(e_i^1, e_i^1, e_i^2, e_i^3)$, where $1 \le e_i^1, e_i^2, e_i^3 \le$ `size(elements_fine,1)` are the numbers of the new elements.
- *First and third edge refined:* There holds, `father2vol(i,:)`$=(e_i^1, e_i^2, e_i^3, e_i^3)$, where $1 \le e_i^1, e_i^2, e_i^3 \le$ `size(elements_fine,1)` are the numbers of the new elements.
- *All edges refined:* There holds, `father2vol(i,:)`$=(e_i^1, e_i^2, e_i^3, e_i^4)$, where $1 \le e_i^1, e_i^2, e_i^3, e_i^4 \le$ `size(elements_fine,1)` are the numbers of the new elements.

In some applications, eg. Section 11.4, it is good to know the geometric position of the new nodes. Figure 4 states the position of the nodes of the refined mesh depending on the type of refinement. The variables j and r denote the respectively colored numbers in Figure 4.

- the numbers of the nodes of the i-th element of the original mesh are given by `elements(`i, r`)`.
- `elements_fine(`e_i^k, j`)` is the number of the j-th node in the element e_i^k of the corresponding diagram in Figure 4.



(a) no refinement (b) first edge refined (c) first and second edge refined

(d) first and third edge refined (e) all edges refined

Figure 4: Geometric position of the new nodes.

One remark is that in case of uniform refinement of all elements, the array `father2vol` has the following structure

$$\texttt{father2vol}(i,j) = 4(i-1)+j.$$

The structure of the `nD` $\times$ 2-array `father2dir` is much more simple.

- There holds, `father2dir(`i`,:)` $= (d_i^1, d_i^1)$ if the i-th element of $\mathcal{E}_\ell$ has not been refined.
- There holds, `father2dir(`i`,:)` $= (d_i^1, d_i^2)$ if the i-th element of $\mathcal{E}_\ell$ has been refined.

Here, d_i^1 and d_i^2 denote the numbers of the new elements. Note that the son elements $E_{d_i^1}$ and $E_{d_i^2}$ are located such that `coordinates(dirichlet(`$d_i^1, 2$`),:)` $=$ `coordinates(dirichlet(`$d_i^2, 1$`),:)`.

11.2 Solving with Symmetric Coupling

In this section, we deal with the solver `solveFEMBEMsymm`. The function is able to solve the reformulation of the model problem (1) with symmetric coupling (see Section 7) with the difference that the terms $\langle Wu_0, v_\ell\rangle_\Gamma$ and $\langle Ku_0, \psi_\ell\rangle_\Gamma$ are not computed analytically, but

we approximate the function $u_0 \in H^{1/2}(\Gamma)$ in $\mathcal{S}^1(\mathcal{T}_\ell)$ and proceed with discrete integral operators. Thus, we solve the following problem: Find $(u_\ell, \phi_\ell) \in X_\ell \times Y_\ell$ such that for all $(v_\ell, \psi_\ell) \in X_\ell \times Y_\ell$

$$\begin{array}{rcl} a(u_\ell, v_\ell) + \langle \big(K_\ell' - \frac{1}{2}\big)\, \phi_\ell \,, v_\ell \rangle_\Gamma + \langle W_\ell u_\ell \,, v_\ell \rangle_\Gamma & = & \langle f \,, v_\ell \rangle_{\Omega^{int}} + \langle \phi_0 + W_\ell u_{0,\ell} \,, v_\ell \rangle_\Gamma, \\ \langle \big(\frac{1}{2} - K_\ell\big)\, u_\ell \,, \psi_\ell \rangle_\Gamma + \langle V_\ell \phi_\ell \,, \psi_\ell \rangle_\Gamma & = & \langle \big(\frac{1}{2} - K_\ell\big)\, u_{0,\ell} \,, \psi_\ell \rangle_\Gamma, \end{array} \tag{61}$$

where $u_{0,\ell} = I_\ell u_0 \in \mathcal{S}(\mathcal{T}_\ell)$ denotes the nodal interpolant of u_0. The implementation of the form $a(\cdot, \cdot)$ allows several problems to be considered.

- *Linear Case:* The operator A in (59) is the identity. There holds

$$a(u_\ell, v_\ell) = \int_{\Omega^{int}} \nabla u_\ell \cdot \nabla v_\ell \, \mathrm{dx}.$$

- *Nonlinear Case:* The operator A in (59) is given by $A\nabla u = \rho(|\nabla u|)\nabla u, \ u \in H^1(\Omega^{int})$, where $\rho : \mathbb{R} \to \mathbb{R}$. In view of the application of Newton's method, we restrict to $\rho \in C^1(\mathbb{R})$. Note that existing convergence and solvability results only apply if the nonlinear operator A is Lipschitz continuous and strongly monotone, i.e.

  $$\|Au - Av\|_{L^2(\Omega^{int})} \lesssim \|u - v\|_{L^2(\Omega^{int})} \text{ as well as } \|u - v\|^2_{L^2(\Omega^{int})} \lesssim \langle Au - Av \,, u - v \rangle_{\Omega^{int}},$$

  for all $(u, v) \in (L^2(\Omega^{int}))^2$. There holds

$$a(u, v) = \int_{\Omega^{int}} \rho(|\nabla u_\ell|)\nabla u_\ell \cdot \nabla v_\ell \, \mathrm{dx}. \tag{62}$$

Let $(u_\ell, \phi_\ell) \in X_\ell \times Y_\ell$ denote a solution of (61) and $(u_\ell^*, \phi_\ell^*) \in X_\ell \times Y_\ell$ be a solution of (44). In the linear case, Theorem 7.6 and Corollary 7.7 state

$$\|u_\ell^* - u_\ell\|_{H^1(\Omega^{int})} + \|\phi_\ell^* - \phi_\ell\|_{H^{-1/2}(\Gamma)} \leq C \|u_{0,\ell} - u_0\|_{H^{1/2}(\Gamma)},$$

where $C > 0$ does not depend on X_ℓ or Y_ℓ. This result also holds for the nonlinear case. For a proof, we refer to [2, Lemma 2]. The (possibly) nonlinear system induced by (61) is solved by an undampted Newton method. With $\mathbf{x} \in \mathbb{R}^{n+m}$ being the coefficient vector of u_ℓ and ϕ_ℓ, i.e. $u_\ell = \sum_{i=1}^n x_i \eta_i$ and $\phi_\ell = \sum_{i=n+1}^{n+m} x_i \xi_{i-n}$, the variational form (61) can be rewritten as follows

$$\begin{pmatrix} N(\mathbf{x}) \\ 0 \end{pmatrix} + B\mathbf{x} = F. \tag{63}$$

Here, N denotes the multi-valued function

$$N : \mathbb{R}^{n+m} \to \mathbb{R}^n, \ \mathbf{x} \mapsto (a(u_\ell, \eta_i))_{i=1}^n.$$

The matrix $B \in \mathbb{R}^{(n+m)\times(n+m)}$ is given by

$$B = \begin{pmatrix} \mathbf{W} & \mathbf{K}^T - \frac{1}{2}\mathbf{M}^T \\ \frac{1}{2}\mathbf{M} - \mathbf{K} & \mathbf{V} \end{pmatrix}, \tag{64}$$

$$\text{where } \begin{cases} \mathbf{W}_{i,j} &= \langle W_\ell \eta_j \, , \eta_i \rangle_\Gamma, & 1 \le i,j \le n, \\ (\mathbf{K}^T - \frac{1}{2}\mathbf{M}^T)_{i,j} &= \langle \left(K'_\ell - \frac{1}{2}\right) \xi_j \, , \eta_i \rangle_\Gamma, & 1 \le i \le n,\ 1 \le j \le m \\ (\frac{1}{2}\mathbf{M} - \mathbf{K})_{i,j} &= \langle \left(\frac{1}{2} - K_\ell\right) \eta_j \, , \xi_i \rangle_\Gamma, & 1 \le i \le m,\ 1 \le j \le n \\ \mathbf{V}_{i,j} &= \langle V_\ell \xi_j \, , \xi_i \rangle_\Gamma, & 1 \le i,j \le m. \end{cases}$$

And finally, the vector $F \in \mathbb{R}^{n+m}$ reads

$$F_i = \begin{cases} \langle f \, , \eta_i \rangle_{\Omega^{int}} + \langle \phi_0 + W_\ell u_{0,\ell} \, , \eta_i \rangle_\Gamma, & 1 \le i \le n \\ \langle \left(\frac{1}{2} - K_\ell\right) u_{0,\ell} \, , \xi_{i-n} \rangle_\Gamma, & n+1 \le i \le n+m \end{cases}. \tag{65}$$

The implementation provides two possibilities for the evaluation of F , see Section 11.7.2 for details. The next step is to determine the Jacobian of the function, we want to drive to zero. Therefore, we need

$$\frac{\partial}{\partial x_k} N(\mathbf{x}) \in \mathbb{R}^{1 \times (n+m)}.$$

For $n+1 \le k \le n+m$, the function N does not depend on x_k. Thus, the derivative equals zero. We consider the case $1 \le k \le n$. Elementary calculus shows

$$\begin{aligned} \left(\frac{\partial}{\partial x_k} N(\mathbf{x})\right)_i &= \frac{\partial}{\partial x_k} \int_{\Omega^{int}} \rho(|\nabla u_\ell|) \nabla u_\ell \cdot \nabla \eta_i \, \mathrm{dx} \\ &= \int_{\Omega^{int}} \frac{\partial}{\partial x_k} \rho(|\nabla u_\ell|) \nabla u_\ell \cdot \nabla \eta_i \, \mathrm{dx} + \int_{\Omega^{int}} \rho(|\nabla u_\ell|) \nabla \eta_k \cdot \nabla \eta_i \, \mathrm{dx}, \end{aligned}$$

as well as

$$\frac{\partial}{\partial x_k} \rho(|\nabla u_\ell|) = \rho'(|\nabla u_\ell|) \frac{1}{|\nabla u_\ell|} \sum_{i=1}^{n} x_i \nabla \eta_i \nabla \eta_k = \rho'(|\nabla u_\ell|) \frac{u_\ell}{|\nabla u_\ell|} \nabla \eta_k.$$

The combination of both results yields

$$\left(\frac{\partial}{\partial x_k} N(\mathbf{x})\right)_i = \int_{\Omega^{int}} \frac{\rho'(|\nabla u_\ell|)}{|\nabla u_\ell|} \nabla \eta_k (\nabla u_\ell)^T \nabla u_\ell (\nabla \eta_i)^T \, \mathrm{dx} + \int_{\Omega^{int}} \rho(|\nabla u_\ell|) \nabla \eta_k \cdot \nabla \eta_i \, \mathrm{dx}. \tag{66}$$

Now, the Jacobian of the function $\mathbf{x} \mapsto \begin{pmatrix} N(\mathbf{x}) \\ 0 \end{pmatrix} + B\mathbf{x} - F$ reads

$$\mathcal{J}(\mathbf{x}) = \begin{pmatrix} J(\mathbf{x}) & 0 \\ 0 & 0 \end{pmatrix} + B \in \mathbb{R}^{(n+m)\times(n+m)}, \tag{67}$$

where $J(\mathbf{x}) \in \mathbb{R}^{n \times n}$ is given by $J(\mathbf{x})_{ki} = \left(\frac{\partial}{\partial x_k} N(\mathbf{x})\right)_i$. To solve the nonlinear system (63), we apply the Newton method given by

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{x}_k + \delta_k, \\ \delta_k &:= -\mathcal{J}(\mathbf{x}_k)^{-1} \left(\begin{pmatrix} N(\mathbf{x}_k) \\ 0 \end{pmatrix} + B\mathbf{x}_k - F \right) \quad k \in \mathbb{N}. \end{aligned} \tag{68}$$

The first part of the documentation of `solveFEMBEMsymm`, cf. Listing 1, describes some preliminary work done by the function.

- There are several ways to call the function

  ```
  [x [,opt.out]]=solveFEMBEMsymm(coordinates,elements,dirichlet,...
  f,u0,grad_u0,singularity,n_points_on_layer, n_layers,...
  phi0_direct_or_projected [,opt.in]),
  ```

  The output parameter `opt.out` is always optional. The following combinations for `opt.in` are allowed:

  - *Linear Case:* `opt.in = RHSdir` or `opt.in` is not given.
  - *Nonlinear Case:* `opt.in = xstart, rho, rhoprime` or `opt.in = xstart, rho, rhoprime, RHSdir`.

  The arrays `coordinates`, `elements`, and `dirichlet` represent the given mesh. These arrays have to be sorted by the function `sortBoundaryNodesLast`, cf. Section 11.7.3. The function handles `f`, `u0`, and `grad_u0` denote the volume force f, the trace jump u_0, as well as the jump of the normal derivative ϕ_0 in (59). The parameter `xstart` contains the starting value $\mathbf{x}_0$ for the Newton method (68). Note that `xstart` has to be sorted by `sortBoundaryNodesLast`. `rho` and `rhoprime` contain the function ρ and its derivative ρ' from (62). `RHSdir` contains a part of the right hand side. A more detailed explanation follows below. The parameters `singularity`, `n_points_on_layer`, and `n_layers` steer the hp-quadrature in Section 11.7.2.

- At first, we check the correct number of input parameters and if the optional output parameter is requested `compute_bdir=1` (Line 3-32). Further, the function determines if the problem is nonlinear (`lin=false`) and declares the respective variables `xstart`, `rho`, `rhoprime`. If parts of the right hand side are given we set `compute_b=0`.

- Note that the input parameter `xstart` is a (`nC`+`nD`) $\times$ 1-array. There holds $n =$`nC` and $m =$`nD`. (Line 35-37).

- (Line 39-41) The number of boundary nodes is determined. Note that equality of `nBN` and `nD` is not mandatory, e.g. in case of an open boundary.

- (Line 42-74) Some useful variables are declared. We allocate the solution vector `x` in Line 45. One can consider `x` as the implementation of the vector $\mathbf{x} \in \mathbb{R}^{n+m}$, cf. Section 10.4.2. In Line 48 we determine the first entry of `coordinates` which belongs to a node V_i of $\mathcal{T}_\ell$ with $V_i \in \Gamma$. To build the Integral operators, we need the numeration for the boundary elements to start with 1. Therefore, we define `dirichlet_modified` in Line 56.

Listing 1: Solving with Symmetric Coupling: Preliminaries

```
1 function [x,varargout] = solveFEMBEMsymm(coordinates,elements,dirichlet,f,u0,grad_u0,...
2     singularity,n_points_on_layer, n_layers,phi0_direct_or_projected, varargin)
```

```
3  if(nargin==13)
4      lin=false;
5      xstart=varargin{1};
6      rho=varargin{2};
7      rhoprime=varargin{3};
8      compute_b=1;
9  elseif(nargin==14)
10     lin=false;
11     xstart=varargin{1};
12     rho=varargin{2};
13     rhoprime=varargin{3};
14     RHSdir=varargin{4};
15     compute_b=0;
16 elseif(nargin==11)
17     lin=true;
18     RHSdir=varargin{1};
19     compute_b=0;
20 elseif(nargin==10)
21     lin=true;
22     compute_b=1;
23 else
24     error('solveFEMBEMsymm: Wrong number of input parameters');
25 end
26 if(nargout==2)
27     compute_bdir=1;
28 elseif(nargout==1)
29     compute_bdir=0;
30 else
31     error('solveFEMBEMsymm: Wrong number of output parameters');
32 end
33
34 %%% Some useful constants of mesh data
35 nE=size(elements,1); %number of elements
36 nC=size(coordinates,1); %number of nodes
37 nD=size(dirichlet,1); %number of boundary elements
38
39 %%% store the unique nodes of dirichlet in boundary_nodes
40 [boundary_nodes]=unique(dirichlet);
41 nBN=length(boundary_nodes); %number of boundary nodes
42
43
44 %%% Allocate solution vector
45 x=zeros(nC+nD,1);
46
47 %%% index of the first boundary node in coordinates
48 first_boundary_node=nC-nBN+1;
49
50 %%% index of the first boundary element in the solution
51 first_boundary_element=nC+1;
52
53 %%% degrees of freedom
```

```
54 degrees_of_freedom=nC+nD;
55
56 dirichlet_modified=dirichlet-first_boundary_node+1;
57
58 %%% First vertex of elements and corresponding edge vectors
59 c1=coordinates(elements(:,1),:);
60 d21=coordinates(elements(:,2),:)-c1;
61 d31=coordinates(elements(:,3),:)-c1;
62
63 %%% dBE stores the vectors of the boundary elements according to
64 %%% definition of orientation
65 dBE=coordinates(dirichlet(:,2),:)-coordinates(dirichlet(:,1),:);
66
67 %%% later on we will need  2*midpoints of the boundary elements
68 midpoints2_BE=coordinates(dirichlet(:,1),:)+coordinates(dirichlet(:,2),:);
69
70 %*** Vector of element areas 4*|T|
71 area4=2*(d21(:,1).*d31(:,2)-d21(:,2).*d31(:,1));
72
73 %%% vector that holds the lengths of boundary elements
74 lengthBE=sqrt(sum(dBE.*dBE,2));
```

The second part of `solveFEMBEMsymm` (cf. Listing 2) builds the integral operators and assembles the right and left hand side of the equations.

- (Line 2-7) The ingredients for the classical FEM Galerkin matrix

$$\mathbf{A}_{ij} = \int_{\Omega^{int}} \nabla\eta_j \nabla\eta_i \,\mathrm{d}x \quad 1 \le i,j \le \mathtt{nC} \tag{69}$$

  are prepared. The array `A` is organized as follows

$$\mathbf{A}^{T_s}_{i_s j_s} := \int_{T_s} \nabla\eta_j \nabla\eta_i \,\mathrm{d}x \quad 1 \le i_s, j_s \le 3,\ 1 \le s \le \mathtt{nE},$$

  where $j = \mathtt{elements}(s, j_s)$, $i = \mathtt{elements}(s, i_s)$ and $T_s \in \mathcal{T}_\ell$.

$$\mathtt{A} = \left(\cdots,\ \mathbf{A}^{T_s}_{11},\ \mathbf{A}^{T_s}_{21},\ \mathbf{A}^{T_s}_{31},\ \mathbf{A}^{T_s}_{12},\ \mathbf{A}^{T_s}_{22},\ \mathbf{A}^{T_s}_{32},\ \mathbf{A}^{T_s}_{13},\ \mathbf{A}^{T_s}_{23},\ \mathbf{A}^{T_s}_{33},\ \ldots \right)^T.$$

- (Line 11-30) The discrete integral operators V_ℓ, W_ℓ, and K_ℓ are assembled. Due to the storage of the matrices with MATLAB's `sparse` function, we need three arrays per operator: One that contains the entries of the matrix, and two which provide information, where (number of row and column) to store the entries. The detailed position is explained below.

- (Line 17-20) Here, the $\frac{1}{2}$ matrix from $\langle(\frac{1}{2}-K)\eta_j\,,\xi_i\rangle_\Gamma$ is built. Note that $\eta_j|_{E_i}$, $E_i \in \mathcal{E}_\ell$ is, if not zero, a linear function with $\eta_j(a) = 0$ and $\eta_j(b) = 1$, where a, b are the endpoints of E_i. There holds $\frac{1}{2}\langle\eta_j\,,\xi_i\rangle_\Gamma = \frac{1}{4}\mathrm{diam}(E_i)$ exactly if η_j is associated either with a or with b.

- (Line 20) The matrix `OnehalfMat` is used later on to build the right hand side.
- (Line 33-35) Now, we assemble the big `sparse` matrix `SysMat`. The position of the entries can be seen in (64), but note that for $1 \leq i \leq$ `firstBoundaryNode` $-$ 1 holds $\eta_i|_\Gamma \equiv 0$. Thus, the discrete integral operators are only built for η_i, `firstBoundaryNode` $\leq i \leq$ `nC` $= n$, see also the respective row and column arrays in Line 12-13, 23-24, 28-29.
- We proceed with the assembly of the right hand side `b`. The (`nC` + `nD`) $\times$ 1-array `b` is organized similar to the vector F in (65), only the ordering of $(\xi_i)_{i=1}^m$ is determined by the array `dirichlet`. Again, there holds $\langle \cdot, \eta_i \rangle_\Gamma = 0$ for all $1 \leq i \leq$ `firstBoundaryNode` $- 1$, e.g. see the row entries in Line 66-67.
- (Line 41) We use a one point quadrature rule (midpoint rule) for the evaluation of $\langle f, \eta_i \rangle_{\Omega^{int}}$.
- (Line 44) As seen in (61), the trace jump u_0 is approximated by its nodal interpolant in $\mathcal{S}^1(\mathcal{T}_\ell)$.
- (Line 47-53) The variable `phi0_direct_or_projected` decides whether $\langle \phi_0, \eta_i \rangle_\Gamma$ is computed with `computePhi0Rhs` or `computeProjPhi0`, see Section 11.7.2 for details.
- (Line 56-74) If `RHSdir` is not given and `opt.out` not requested, i.e. `compute_b=0` and `compute_bdir=0` ("usual case"), the array `b` is assembled by `accumarray`. If the Dirichlet part of the right hand side (e.g. for post-processing) is requested (`compute_bdir=1`) we split `b` in two parts

$$\texttt{bvol(i)} = \langle f, \eta_{i+\texttt{firstBoundaryNode}} \rangle_{\Omega^{int}}, \quad 1 \leq i \leq \texttt{nBN},$$

$$\texttt{bdir(i)} = \begin{cases} \langle \phi_0 + W u_{0,\ell}, \eta_{i+\texttt{firstBoundaryNode}} \rangle_\Gamma, & 1 \leq i \leq \texttt{nBN} \\ \langle \left(\frac{1}{2} - K\right) u_{0,\ell}, \xi_{i-\texttt{nBN}} \rangle_\Gamma, & \texttt{nBN}+1 \leq i \leq \texttt{nBN}+\texttt{nD} \end{cases},$$

  and add them in Line 63.
- (Line 65) The output parameter `RHSdir` is set.
- (Line 73-75) If the Dirichlet part of the right hand side is given, we just have to assemble `bvol` and add `RHSdir`.

Listing 2: Solving with Symmetric Coupling: Part 2

```
1 %%% rows, cols and values for the stiffness matrix of -div(grad u)
2 rowsA=reshape(elements(:,[1 2 3 1 2 3 1 2 3])',9*nE,1);
3 colsA=reshape(elements(:,[1 1 1 2 2 2 3 3 3])',9*nE,1);
4 a=(sum(d21.*d31,2)./area4)';
```

```
b=(sum(d31.*d31,2)./area4)';
c=(sum(d21.*d21,2)./area4)';
A=[-2*a+b+c;a-b;a-c;a-b;b;-a;a-c;-a;c];

%%% ---- Assembly of the left hand side ----

%%% assembly of W: hypersingular integral operator
rowsW=repmat([first_boundary_node:nC]',nBN,1);
colsW=reshape(repmat([first_boundary_node:nC],nBN,1),nBN*nBN,1);
W=buildW(coordinates(first_boundary_node:nC,:),dirichlet_modified,1);

%%% assembly of "1/2" in context with K
rowsOnehalf=repmat([first_boundary_element:degrees_of_freedom]',2,1);
colsOnehalf=dirichlet(:);
Onehalf=0.25*repmat(lengthBE,2,1);
OnehalfMat=sparse(repmat([1:nD]',2,1),dirichlet_modified(:),Onehalf);

%%% assembly of K: double layer potential
rowsK=repmat([first_boundary_element:degrees_of_freedom]',nBN,1);
colsK=reshape(repmat([first_boundary_node:nC],nD,1),nD*nBN,1);
K=buildK(coordinates(first_boundary_node:nC,:),dirichlet_modified,1);

%%% assembly of V: single layer potential
rowsV=repmat([first_boundary_element:degrees_of_freedom]',nD,1);
colsV=reshape(repmat([first_boundary_element:degrees_of_freedom],nD,1),nD*nD,1);
V=buildV(coordinates(first_boundary_node:nC,:),dirichlet_modified,1);

%***Assembly of the Matrix for the linear part of the operator (the BEM-part)
SysMat=sparse([rowsW; rowsOnehalf; rowsK; colsOnehalf; colsK; rowsV],...
    [colsW; colsOnehalf; colsK; rowsOnehalf; rowsK; colsV],...
    [W(:); Onehalf; -K(:); -Onehalf; K(:); V(:)]);


%%% ---- Assembly of the right-hand side ----

%%% Values of the volume force f
fsT=feval(f,c1+(d21+d31)/3);

%%% compute u0h:=I_h(u0) =^ the nodal interpolant of u0
u0h=feval(u0,coordinates(first_boundary_node:nC,:));

%%% computing of phi0 for the rhs
if strcmp(phi0_direct_or_projected,'projected')
    phi0_rhs=repmat(computeProjPhi0(coordinates,dirichlet,grad_u0,singularity,...
        n_points_on_layer,n_layers).*lengthBE/2,1,2);
else
    phi0_rhs=computePhi0Rhs(coordinates,dirichlet,grad_u0,singularity,...
        n_points_on_layer,n_layers);
end

%%% build the right hand side
```

```
56 if(compute_b)
57     if(compute_bdir)
58     bvol = accumarray([elements(:)],[repmat(12\area4.*fsT,3,1)],[nC,1]);
59     bdir= accumarray([[1:nBN]';dirichlet(:)-first_boundary_node+1;...
60         [first_boundary_element:degrees_of_freedom]'-first_boundary_node+1],...
61         [W*u0h;reshape(phi0_rhs,2*nD,1);...
62         (OnehalfMat-K)*u0h],[degrees_of_freedom-first_boundary_node+1,1]);
63     b=[bvol(1:first_boundary_node-1);bvol(first_boundary_node:nC)+...
64         bdir(1:nBN);bdir(nBN+1:end)];
65     varargout{1}=bdir;
66     else
67         b=accumarray([[elements(:)];[first_boundary_node:nC]';dirichlet(:);...
68         [first_boundary_element:degrees_of_freedom]'],...
69         [[repmat(12\area4.*fsT,3,1)];W*u0h;reshape(phi0_rhs,2*nD,1);...
70         (OnehalfMat-K)*u0h],[degrees_of_freedom,1]);
71     end
72 else
73     bvol = accumarray([elements(:)],[repmat(12\area4.*fsT,3,1)],[nC,1]);
74     b=[bvol(1:first_boundary_node-1);RHSdir(1:nBN)+...
75         bvol(first_boundary_node:nC);RHSdir(nBN+1:end)];
```

The final part of `solveFEMBEMsymm` (Listing 3) solves the (possibly) nonlinear system. For the ease of notation, we define

$$u_\ell = \sum_{i=1}^{\mathsf{nC}} x_i^k \, \eta_i \quad \text{and} \quad \phi_\ell = \sum_{i=1}^{\mathsf{nD}} x_{i+\mathsf{nC}}^k \, \xi_i,$$

where $\mathbf{x}_k$ is the current Newton approximation.

- (Line 2) If `lin=false`, the preparations for the Newton method are done.
- (Line 8) The loop for the Newton iteration.
- (Line 10-12) The element wise gradient of the current Newton approximation `x` (corresponds to $\mathbf{x}_k$ in (68)) is computed.
- (Line 13) `normgrad` contains the norm of the gradient of `x` on each element.
- (Line 15-18) Some ingredients for Jacobian are declared. `rhovec` contains the evaluations of ρ at each entry of `normgrad`. `AMat(j,i)` corresponds to the last term in (66) as well as `SysVec` corresponds to $N(\mathbf{x}_n)$.
- (Line 20-22) The array `tmp` contains the entries of $\begin{pmatrix} N(\mathbf{x}_n) \\ 0 \end{pmatrix} + B\mathbf{x}_n$ from (63).
- (Line 24-28) `rhoprimevec` contains the evaluations of ρ' at each entry of `normgrad`. The arrays `rhovec2` and `area4` are reshaped such that their entries correspond to the entries of `A`.

- (Line 31) With $\tilde{\mathbf{x}}_k := (x^k_{r_1}, x^k_{r_2}, x^k_{r_3})^T$, one can write formally

$$\texttt{A2(i:i+3)} = \mathbf{A}^{T_s}\tilde{\mathbf{x}}_k, = \left(\int_{T_s} \nabla\eta_j(\nabla u_\ell)^T \,\mathrm{dx}\right)_{j=(r_1,r_2,r_3)} \qquad T_s \in \mathcal{T}_\ell. \tag{70}$$

  There holds $(r_1, r_2, r_3) =$ `elements(s,:)` as well as $i = 3(s-1)+1$. To avoid building up the whole matrix, this product is performed on the level of the entries.

- (Line 32-33) We want to realize

$$v^s_{i_s j_s} = \int_{T_s} \nabla\eta_j(\nabla u_\ell)^T \nabla u_\ell (\nabla\eta_i)^T \,\mathrm{dx} \quad 1 \le i_s, j_s \le 3,\ 1 \le s \le \texttt{nE},$$

  cf. (66), where $i =$ `elements(s,`i_s`)` and $j =$ `elements(s,`j_s`)`. Note that the integrand in (70) is constant, therefore we can put together $v_{i_k j_k}$ with elements from `A2`. The arrays `A2` and `A3` contain two different orders of `A2` from Line 31 such that there holds formally

$$\texttt{A2.*A3} = \int_{T_s} 1 \,\mathrm{dx}\ (\ldots, v^s_{11}, v^s_{21}, v^s_{31}, v^s_{12}, v^s_{22}, v^s_{32}, v^s_{13}, v^s_{23}, v^s_{33}, \ldots)^T \quad 1 \le s \le \texttt{nE}.$$

- (Line 35) Consequently, we are able to assemble the first part of the Jacobian `AMat2` which corresponds to the first term of (66).

- (Line 37-38) `SysMat2` finally contains the Jacobian (cf. (67)).

- (Line 40-50) We perform the Newton step. Theory predicts the sequence $(\delta_k)_{k\ge k_0}$ to be monotone for x_{k_0} being sufficiently close to the solution. Thus, we check if `norm(delta)` is at least five orders of magnitude lower than the first Newton residual stored in `norm_delta_first`. We stop the iteration if for the first time holds

$$\|\delta_{k-1}\| \le 10^{-5}\|\delta_0\| \quad \text{and} \quad \|\delta_k\| > \|\delta_{k-1}\|.$$

- (Line 54-59) In case of a linear problem, we assemble the FEM-Galerkin matrix `AMat` (cf. (69)) and add it to the existing Galerkin matrix `SysMat`. The linear system is solved by MATLAB's backslash operator.

Listing 3: Solving with Symmetric Coupling: The Newton Method

```
1 %%% ---- Computation of (S1/P0)-FEMBEM approximation ----
2 if(lin==false)
3     x=xstart;
4     norm_delta_old=1;
5     %*** Newton's Method
6     fprintf('norm of residual:\n');
7     step=1;
```

```
8       while(1)
9           %*** Elementwise gradient of x
10          u21 = repmat( 2*(x(elements(:,2))-x(elements(:,1)))./area4, 1,2);
11          u31 = repmat( 2*(x(elements(:,3))-x(elements(:,1)))./area4, 1,2);
12          grad = (d31.*u21 - d21.*u31)*[0 -1 ; 1 0];
13          normgrad=sqrt(sum(grad.*grad,2));
14          %*** Evaluate the nonlinear part of the Operator (the FEM-part)
15          rhovec=rho(normgrad);
16          rhovec=reshape(repmat(rhovec',9,1),9*nE,1);
17          AMat=sparse(rowsA,colsA,rhovec.*A(:));
18          SysVec=AMat*x(1:nC);
19          %***Evaluate the linear part of the Operator
20          tmp=SysMat*x;
21          %***Finally, adding the two parts...
22          tmp=[tmp(1:nC) + SysVec; tmp(nC+1:end)];
23          %%% ---Assembly of the Jacobian---
24          rhoprimevec=rhoprime(normgrad);
25          rhovec2=rhoprimevec./normgrad;
26          rhovec2=reshape(repmat(rhovec2',9,1),9*nE,1);
27          area4new=reshape(repmat(area4',9,1),9*nE,1);
28          xvec=reshape(x(elements(:,[1 2 3 1 2 3 1 2 3]))',9*nE,1);
29          %***Compute integral of grad(phi_j)*grad(u)'*grad(u)*grad(phi_i)'
30          A2=(A(:).*xvec);
31          A2=sum(reshape(A2,3,3*nE),1);
32          A3=reshape(repmat(A2,3,1),9*nE,1);
33          A2=reshape(repmat(reshape(A2,3,nE),3,1),9*nE,1);
34          %***Assembly of the difficult part (Jacobian of the nonlinear part)
35          AMat2=sparse(rowsA,colsA,4.*rhovec2.*(A3.*A2)./(area4new));
36          %***Adding the easy part (Jacobian of the linear part)
37          SysMat2=[AMat+AMat2+SysMat(1:nC,1:nC),SysMat(1:nC,nC+1:end);...
38              SysMat(nC+1:end,1:nC),SysMat(nC+1:end,nC+1:end)];
39          %***Perform the famous step...
40          delta=SysMat2\(b-tmp);
41          norm_delta_new=norm(delta);
42          if(step==1)
43              norm_delta_first=norm_delta_new;
44          end
45          if(norm_delta_old < norm_delta_new &&...
46                  norm_delta_old <norm_delta_first*(1e-5) && step>1)
47              break;
48          end
49          norm_delta_old=norm_delta_new;
50          x=x+delta;
51          fprintf('step %d: %d\n',step,norm_delta_old)
52          step=step+1;
53      end
54  else
55      AMat=sparse(rowsA,colsA,A(:));
56      SysMat=[SysMat(1:nC,1:nC)+AMat,SysMat(1:nC,nC+1:end);...
57          SysMat(nC+1:end,1:nC),SysMat(nC+1:end,nC+1:end)];
58      x=SysMat\b;
```

```
59 end
```

11.3 Solving with Johnson-Nédélec Coupling

The implementation of the *Johnson-Nédélec* coupling method in `solveFEMBEMJohNed` is very similar to `solveFEMBEMsymm` discussed in Section 11.2. Now, we want to solve the following problem: Find $(u_\ell, \phi_\ell) \in X_\ell \times Y_\ell$ such that

$$\begin{aligned} a(u_\ell, v_\ell) - \langle \phi_\ell \,, v_\ell \rangle_\Gamma &= \langle f \,, v_\ell \rangle_{\Omega^{int}} + \langle \phi_0 \,, v_\ell \rangle, \\ \langle \big(\tfrac{1}{2} - K_\ell\big)\, u_\ell \,, \psi_\ell \rangle_\Gamma + \langle V_\ell \phi_\ell \,, \psi_\ell \rangle_\Gamma &= \langle \big(\tfrac{1}{2} - K_\ell\big)\, u_{0,\ell} \,, \psi_\ell \rangle_\Gamma, \end{aligned} \tag{71}$$

for all $(v_\ell, \psi_\ell) \in X_\ell \times Y_\ell$. As in the previous section $u_{0,\ell} \in \mathcal{S}(\mathcal{T}_\ell)$ denotes the nodal interpolant of u_0. Again, the implementation of the form $a(\cdot, \cdot)$ allows several problems to be considered.

- *Linear Case:* The operator A in (59) is the identity. There holds

$$a(u_\ell, v_\ell) = \int_{\Omega^{int}} \nabla u_\ell \cdot \nabla v_\ell \, \mathrm{dx}.$$

- *Nonlinear Case:* The operator A in (59) is given by $A\nabla u = \rho(|\nabla u|)\nabla u,\ u \in H^1(\Omega^{int})$, where $\rho : \mathbb{R} \to \mathbb{R}$. In view of the application of Newton's method, we restrict to $\rho \in C^1(\mathbb{R})$. There holds

$$a(u, v) = \int_{\Omega^{int}} \rho(|\nabla u_\ell|)\nabla u_\ell \cdot \nabla v_\ell \, \mathrm{dx}. \tag{72}$$

Let $(u_\ell, \phi_\ell) \in X_\ell \times Y_\ell$ denote a solution of (71) and $(u_\ell^*, \phi_\ell^*) \in X_\ell \times Y_\ell$ be a solution of (37). In the linear case, Theorem 6.3 and Corollary 6.4 state

$$\|u_\ell^* - u_\ell\|_{H^1(\Omega^{int})} + \|\phi_\ell^* - \phi_\ell\|_{H^{-1/2}(\Gamma)} \leq C \|u_{0,\ell} - u_0\|_{H^{1/2}(\Gamma)},$$

where $C > 0$ does not depend on X_ℓ or Y_ℓ. In the nonlinear case, no such results are available. We rewrite the variational form (71) as

$$\begin{pmatrix} N(x) \\ 0 \end{pmatrix} + B\mathbf{x} = F. \tag{73}$$

As in Section 11.2, the function N is given by

$$N : \mathbb{R}^{n+m} \to \mathbb{R}^n,\ \mathbf{x} \to (a(u_\ell, \eta_i))_{i=1}^n.$$

The matrix $B \in \mathbb{R}^{(n+m)\times(n+m)}$ reads

$$B = \begin{pmatrix} \mathbf{W} & -\mathbf{M}^T \\ \frac{1}{2}\mathbf{M} - \mathbf{K} & \mathbf{V} \end{pmatrix}, \tag{74}$$

$$\text{where} \quad \begin{cases} \mathbf{W}_{i,j} &= \langle W_\ell \eta_j \,, \eta_i \rangle_\Gamma, & 1 \le i,j \le n, \\ \mathbf{M}^T_{i,j} &= \langle \xi_j \,, \eta_i \rangle_\Gamma, & 1 \le i \le n,\ 1 \le j \le m \\ (\frac{1}{2}\mathbf{M} - \mathbf{K})_{i,j} &= \langle \left(\frac{1}{2} - K_\ell\right) \eta_j \,, \xi_i \rangle_\Gamma, & 1 \le i \le m,\ 1 \le j \le n \\ \mathbf{V}_{i,j} &= \langle V_\ell \xi_j \,, \xi_i \rangle_\Gamma, & 1 \le i,j \le m. \end{cases}$$

And finally, the vector $F \in \mathbb{R}^{n+m}$ is given by

$$F_i = \begin{cases} \langle f \,, \eta_i \rangle_{\Omega^{int}} + \langle \phi_0 \,, \eta_i \rangle_\Gamma, & 1 \le i \le n \\ \langle \left(\frac{1}{2} - K_\ell\right) u_{0,\ell} \,, \xi_{i-n} \rangle_\Gamma, & n+1 \le i \le n+m \end{cases}. \tag{75}$$

Again, we define the Newton method as

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{x}_k + \delta_k, \\ \delta_k &:= -\mathcal{J}(\mathbf{x}_k)^{-1} \left(\begin{pmatrix} N(\mathbf{x}_k) \\ 0 \end{pmatrix} + B\mathbf{x}_k - F \right) \quad k \in \mathbb{N}. \end{aligned} \tag{76}$$

Now, we want to discuss the small differences to the code of `solveFEMBEMsymm`.

- (Line 4-18) The discrete integral operators V_ℓ and K_ℓ are assembled. Note that we do not need to build W_ℓ, which is the main advantage of *Johnson-Nédélec coupling*.
- (Line 21-23) The sparse matrix `SysMat` is built. The position of the entries can be seen in (74).

```
1  %%% ---- Assembly of the left hand side ----
2
3  %%% assembly of "1/2" in context with K
4  rowsOnehalf=repmat([first_boundary_element:degrees_of_freedom]',2,1);
5  colsOnehalf=dirichlet(:);
6  Onehalf=0.25*repmat(lengthBE,2,1);
7
8  OnehalfMat=sparse(repmat([1:nD]',2,1),dirichlet_modified(:),Onehalf);
9
10 %%% assembly of K: double layer potential
11 rowsK=repmat([first_boundary_element:degrees_of_freedom]',nBN,1);
12 colsK=reshape(repmat([first_boundary_node:nC],nD,1),nD*nBN,1);
13 K=buildK(coordinates(first_boundary_node:nC,:),dirichlet_modified,1);
14
15 %%% assembly of V: single layer potential
16 rowsV=repmat([first_boundary_element:degrees_of_freedom]',nD,1);
17 colsV=reshape(repmat([first_boundary_element:degrees_of_freedom],nD,1),nD*nD,1);
18 V=buildV(coordinates(first_boundary_node:nC,:),dirichlet_modified,1);
19
20 %***Assembly of the Matrix for the linear part of the operator (the BEM-part)
21 SysMat=sparse([colsOnehalf; rowsOnehalf; rowsK; rowsV],...
22     [rowsOnehalf; colsOnehalf; colsK; colsV],...
23     [-2*Onehalf; Onehalf; -K(:); V(:)]);
```

- The right-hand side is built. The array `b` is organized similar to the vector F in (75). For detailed explanation on the variables `compute_b` and `compute_bdir`, see Section 11.2.

```
if(compute_b)
    if(compute_bdir)
    bvol = accumarray([elements(:)],[repmat(12\area4.*fsT,3,1)],[nC,1]);
    bdir= accumarray([dirichlet(:)-first_boundary_node+1;...
        [first_boundary_element:degrees_of_freedom]'-first_boundary_node+1],...
            [reshape(phi0_rhs,2*nD,1);(OnehalfMat-K)*u0h],...
            [degrees_of_freedom-first_boundary_node+1,1]);
    b=[bvol(1:first_boundary_node-1);bvol(first_boundary_node:nC)+...
        bdir(1:nBN);bdir(nBN+1:end)];
    varargout{1}=bdir;
    else
        b = accumarray([elements(:);dirichlet(:);...
            [first_boundary_element:degrees_of_freedom]'],...
            [repmat(12\area4.*fsT,3,1);reshape(phi0_rhs,2*nD,1);...
            (OnehalfMat-K)*u0h],[degrees_of_freedom,1]);
    end
else
    bvol = accumarray([elements(:)],[repmat(12\area4.*fsT,3,1)],[nC,1]);
    b=[bvol(1:first_boundary_node-1);RHS(1:nBN)+...
        bvol(first_boundary_node:nC);RHS(nBN+1:end)];
end
```

11.4 Error Estimators

In this section, we discuss the implementation of the error estimator μ_ℓ from Section 8.2. In view of the steering of our adaptive algorithm, we additionally define the error estimator

$$\varrho_\ell := \mu_\ell + \operatorname{osc}_\ell,$$

as well as the local contributions

$$\begin{aligned} \varrho_\ell(T)^2 &= \|(1 - I_\ell^{\mathcal{T}})\widehat{u}_\ell\|^2_{H^1(T)} && T \in \mathcal{T}_\ell, \\ \varrho_\ell(E)^2 &= \|h_\ell^{1/2}(1 - \Pi_\ell)\widehat{\phi}_\ell\|^2_{L^2(E)} + \|h_\ell^{1/2}(u_0 - u_{0,\ell})'\|^2_{L^2(E)} && E \in \mathcal{E}_\ell. \end{aligned}$$

The computation of μ_ℓ is split into several functions. We start with `computeEstVolume` (Listing 4).

- The function is called with the arrays `coordinates` and `elements` representing the uniformly refined mesh $\widehat{\mathcal{T}}_\ell$. The array `father2vol` links $\widehat{\mathcal{T}}_\ell$ with $\mathcal{T}_\ell$. The (`nC`+`nD`)×1-array `x` denotes the solution w.r.t the uniformly refined mesh. Output of the function are the two arrays `tmu1` and `tmu2` which contain

$$\begin{aligned} \texttt{tmu1} &= \left(\|\nabla\left((1 - I_\ell^{\mathcal{T}})\widehat{u}_\ell\right)\|^2_{L^2(T_s)}\right)_{1\le s\le \texttt{nE}}, \\ \texttt{tmu2} &= \left(\|(1 - I_\ell^{\mathcal{T}})\widehat{u}_\ell\|^2_{L^2(T_s)}\right)_{1\le s\le \texttt{nE}}, \end{aligned}$$

where `nE` is defined in Line 2 and $T_s \in \mathcal{T}_\ell$.

- (Line 5-9) The vectors of the triangle sides of the fine mesh $\widehat{\mathcal{T}}_\ell$ are determined. `area2` contains $2(\int_{\widehat{T}_s} 1\,\mathrm{dx})_{s=1}^{4\mathtt{nE}}$, $\widehat{T}_s \in \widehat{\mathcal{T}}_\ell$.
- (Line 11) Due to the simple structure of `father2vol`, we use the array `idx` instead. Here, `idx` $+ j$, $j = 0, \dots, 3$ corresponds to `father2vol(:,`$j+1$`)`.
- (Line 12-18) We compute $I_\ell^{\mathcal{T}} \widehat{u}_\ell$. Therefore, we need to know the geometric position of the nodes $\widehat{\mathcal{T}}_\ell$, see Section 11.1 for details.
- (Line 21-26) We compute $\|(1 - I_\ell^{\mathcal{T}})\widehat{u}_\ell\|^2_{L^2(T_s)}$ over each element $T_s \in \mathcal{T}_\ell$. Therefore, we use the fact that $\big((1 - I_\ell^{\mathcal{T}})\widehat{u}_\ell\big)\big|_{\widehat{T}_s}$ is an affine function of the form

  $$L(y_1, y_2) = ay_1 + by_2 + d.$$

  The coefficients are determined in Line 22-24, and in Line 25 we compute the integral $\|(1 - I_\ell^{\mathcal{T}})\widehat{u}_\ell\|^2_{L^2(\widehat{T}_s)}$ analytically. Finally, we sum over the elements to obtain

  $$\|(1 - I_\ell^{\mathcal{T}})\widehat{u}_\ell\|^2_{L^2(T_s)} = \sum_{i=1}^{4} \|(1 - I_\ell^{\mathcal{T}})\widehat{u}_\ell\|^2_{L^2(\widehat{T}_{s_i})},$$

  for all elements $T_s \in \mathcal{T}_\ell$. There holds $(s_1, s_2, s_3, s_4) =$ `father2vol(`s`,:)`.
- (Line 29-36) The element wise gradient of $\widehat{u}_\ell$ and its nodal interpolant $I_\ell^{\mathcal{T}} \widehat{u}_\ell$ is computed.
- (Line 39-41) The quantity `tmu1` $= (\|\nabla\left((1 - I_\ell^{\mathcal{T}})\widehat{u}_\ell\right)\|^2_{L^2(T_s)})_{1 \le s \le \mathtt{nE}}$ is assembled. Here, we use that $\nabla\left((1 - I_\ell^{\mathcal{T}})\widehat{u}_\ell\right)$ is constant on each element $\widehat{T}_s \in \widehat{\mathcal{T}}_\ell$.

Listing 4: Estimate the Error Contributions of the Volume Mesh

```
1  function [tmu1,tmu2] = computeEstVolume(x,coordinates,elements,father2vol)
2  nE = size(elements,1)/4;
3
4  %*** First vertex of elements and corresponding edge vectors
5  c1  = coordinates(elements(:,1),:);
6  d21 = coordinates(elements(:,2),:) - c1;
7  d31 = coordinates(elements(:,3),:) - c1;
8  %*** Vector of element volumes 2*|T|
9  area2 = d21(:,1).*d31(:,2)-d21(:,2).*d31(:,1);
10 %*** Compute nodal-interpolation of uh/2 on coarse mesh
11 idx = 1:4:4*nE;
12 Ihx=zeros(2,1);
13 Ihx(elements(idx+1,1))=x(elements(idx+1,1));
14 Ihx(elements(idx+1,2))=(x(elements(idx+1,1))+x(elements(idx+2,2)))/2;
```

```
15 Ihx(elements(idx+2,2))=x(elements(idx+2,2));
16 Ihx(elements(idx+2,3))=(x(elements(idx+2,2))+x(elements(idx+3,1)))/2;
17 Ihx(elements(idx+3,1))=x(elements(idx+3,1));
18 Ihx(elements(idx,3))=(x(elements(idx,2))+x(elements(idx+1,1)))/2;
19 %*** Compute L2-Norm of (1-Ih)uh/2
20 nC=size(coordinates,1);
21 diffx=x(1:nC)-Ihx;
22 d=diffx(elements(:,3));
23 a=diffx(elements(:,2))-diffx(elements(:,3));
24 b=diffx(elements(:,1))-diffx(elements(:,3));
25 l2norm=area2.*(1/12*(a.*a+a.*b+b.*b)+1/3*(d.*a+b.*d)+1/2*(d.*d));
26 tmu2=sum(reshape(l2norm,4,nE),1)';
27
28 %*** Elementwise gradient of uh/2
29 u21 = repmat( (x(elements(:,2))-x(elements(:,1)))./area2, 1,2);
30 u31 = repmat( (x(elements(:,3))-x(elements(:,1)))./area2, 1,2);
31 grad = (d31.*u21 - d21.*u31)*[0 -1 ; 1 0];
32
33 %*** Elementwise gradient of nodal interpolant of uh/2
34 u21 = repmat( (Ihx(elements(:,2))-Ihx(elements(:,1)))./area2, 1,2);
35 u31 = repmat( (Ihx(elements(:,3))-Ihx(elements(:,1)))./area2, 1,2);
36 Ihgrad = (d31.*u21 - d21.*u31)*[0 -1 ; 1 0];
37
38 %*** Compute L2-norm of grad[(1-Ih)uh/2]
39 diffgrad = grad-Ihgrad;
40 norm2 = sum(diffgrad.^2,2);
41 tmu1= 2*area2(1:4:4*nE).*sum(reshape(norm2,4,nE),1)'/4;
```

We proceed with the main function to compute the error estimator `computeHH2mu` (Listing 5).

- There are two ways to call the function:
  - In case of uniform refinement, we call

    `[vol1,vol2,boundary,osc]=computeHH2mu(x,coordinates,elements,...`
    `father2vol,father2dir,coordinatesold,dirichletold,u0)`.
  - In case of adaptive refinement, we additionally need the array `dirichlet`

    `[vol1,vol2,boundary,osc]=computeHH2mu(x,coordinates,elements,...`
    `father2vol,father2dir,coordinatesold,dirichletold,u0,dirichlet)`.
- The `*old` input parameters correspond to a coarse mesh $(\mathcal{T}_\ell, \mathcal{E}_\ell)$. The parameters `coordinates`, `elements`, and optionally `dirichlet` correspond to a uniformly refined mesh $(\widehat{\mathcal{T}}_\ell, \widehat{\mathcal{E}}_\ell)$. `x` represents the solution w.r.t the fine mesh. `father2vol` and `father2dir` denote the usual linking arrays. `u0` is a function handle for the Dirichlet data u_0 from (59).

- The output parameters contain the parts of ϱ_ℓ, i.e.

$$\begin{aligned} \texttt{vol1} &= \texttt{tmu1}, \\ \texttt{vol2} &= \texttt{tmu2}, \\ \texttt{boundary} &= \left(\|h_\ell^{1/2}(1-\Pi_\ell)\widehat{\phi}_\ell\|^2_{L^2(E_s)}\right)_{1\le s\le \texttt{nD}}, \\ \texttt{osc} &= \left(\|h_\ell^{1/2}(u_0-u_{0,\ell})'\|^2_{L^2(E_s)}\right)_{1\le s\le \texttt{nD}}. \end{aligned}$$

- In case of adaptive refinement (Line 9-17), we compute the data oscillations with respect to the fine mesh, i.e.

$$\texttt{osc} = \left(\|h_\ell^{1/2}(u_0-u_{0,\ell})'\|^2_{L^2(\widehat{E}_{s_1})} + \|h_\ell^{1/2}(u_0-u_{0,\ell})'\|^2_{L^2(\widehat{E}_{s_2})}\right)_{1\le s\le \texttt{nD}},$$

  where $\widehat{E}_{s_1}$ and $\widehat{E}_{s_2}$ denote the sons of E_s ($\texttt{father2dir(s,:)} = (s_1, s_2)$).

Listing 5: Compute the Error Estimator

```
1  function [vol1,vol2,boundary,osc]=computeHH2mu(x,coordinates,elements,...
2      father2vol,father2dir,coordinatesold,dirichletold,u0,varargin)
3  if(nargin==8)
4      nC=size(coordinates,1);
5      [vol1,vol2]=computeEstVolume(x,coordinates,elements,father2vol);
6      boundary = computeEstSlpMuTilde(coordinatesold,dirichletold,...
7          father2dir,x(nC+1:end));
8      osc = computeOscDirichlet(coordinatesold,dirichletold,u0);
9  else
10     dirichlet=varargin{1};
11     nC=size(coordinates,1);
12     [vol1,vol2]=computeEstVolume(x,coordinates,elements,father2vol);
13     boundary = computeEstSlpMuTilde(coordinatesold,dirichletold,...
14         father2dir,x(nC+1:end));
15     osc_fine = computeOscDirichlet(coordinates,dirichlet,u0);
16     osc=osc_fine(father2dir(:,1))+osc_fine(father2dir(:,2));
17 end
```

11.5 Adaptive and Uniform Algorithm

Despite the misleading name, the function `adaptiveAlgorithmFEMBEM` contains both, the uniform and the adaptive strategy to compute a good approximation (u_ℓ, ϕ_ℓ). In view of post-processing, we perform some extra operations which wouldn't be mandatory for the adaptive or the uniform algorithm. Additionally, we do some time measurement with `tic` and `toc`, see Section 12 for details. The "pure" adaptive Algorithm reads as follows: **INPUT:** Initial meshes $(\mathcal{T}_0, \mathcal{E}_0)$ for $\ell = 0$, adaptivity parameter $\theta \in (0, 1)$.

1. Generate uniformly refined meshes $\widehat{\mathcal{T}}_\ell$, $\widehat{\mathcal{E}}_\ell$.
2. Compute discrete solution $(\widehat{u}_\ell, \widehat{\phi}_\ell) \in \widehat{X}_\ell \times \widehat{Y}_\ell$.
3. Compute refinement indicators $\varrho_\ell(\tau)$ for all $\tau \in \mathcal{T}_\ell \cup \mathcal{E}_\ell$.
4. Determine set $\mathcal{M}_\ell \subseteq \mathcal{T}_\ell \cup \mathcal{E}_\ell$ which satisfies Dörfler marking.
5. Mark triangles $T \in \mathcal{T}_\ell \cap \mathcal{M}_\ell$ and boundary elements $E \in \mathcal{E}_\ell \cap \mathcal{M}_\ell$ for refinement.
6. Generate new meshes $(\mathcal{T}_{\ell+1}, \mathcal{E}_{\ell+1})$, increase counter $\ell \mapsto \ell + 1$, and go to (1).

OUTPUT: Sequence of error estimators $(\varrho_\ell)_{\ell\in\mathbb{N}}$ and discrete solutions $(\widehat{u}_\ell, \widehat{\phi}_\ell)_{\ell\in\mathbb{N}}$.
At first, we discuss the ways to call the function (Listing 6).

- The first three parameters `singularity`, `n_points_on_layer`, and `n_layers` steer the hp-quadrature used in Section 11.7.2. `phi_direct_or_projected` is explained in Section 11.7.2. The parameter `adptrho` corresponds to the adaptivity parameter θ. It also decides which refinement strategy is used, i.e. $0 < \theta < 1$ for adaptive refinement and $1 + \varepsilon \leq \theta$ for uniform refinement. We perform `nPreRefinementSteps` steps of uniform refinement, before we start the algorithm. There holds `lin` $= 1$ if the problem to be computed is linear and `lin` $= 0$ if not. The variable `method` contains the chosen solving method, i.e
  - `method=1` for the use of *Symmetric Coupling*, cf. Section 11.2
  - `method=2` for the use of *Johnson-Nédélec Coupling*, cf. Section 11.3.

  Finally, `nEmax` contains the maximal number of volume elements.

Listing 6: Adaptive and Uniform Algorithm, Part 1

```
1 function adaptiveAlgorithmFEMBEM(singularity,n_points_on_layer,n_layers,...
2     phi0_direct_or_projected,adptrho,nPreRefinementSteps,lin,method,nEmax)
```

We proceed with the documentation of the Uniform Algorithm (Listing 7).

- (Line 1-5) The variable `solveFEMBEM` contains the function handle of the chosen solving method.
- (Line 8-10) We load the data of the current example.
- (Line 12-18) We prepare the data and rotate the domain to get a symmetric solution.
- (Line 21-24) Perform the pre-refinement steps.
- (Line 25-42) Declare some system variables, e.g. for post-processing or time measurement.

- (Line 51-52) We rearrange the arrays `coordinates`, `elements`, `dirichlet`, and `ustart` such that the nodes V of $\mathcal{T}_\ell$ with $V \in \Gamma$ are last. The array `ustart` contains a part of the initial guess $\mathbf{x}_0$ of the Newton method. The entries of `ustart` correspond to the array `coordinates`, thus we have to sort `ustart` too.
- (Line 55-57) Save the data structures for post-processing.
- (Line 60-67) Depending on the problem (linear or nonlinear), we call the respective solving function.
- (Line 69) Save the computational time for the current step.
- (Line 72) Save the solution `x` for post-processing.
- (Line 73-86) We compute the error estimator ϱ_ℓ for the previous mesh, because there holds $\widehat{\mathcal{T}}_\ell = \mathcal{T}_{\ell+1}$ and $\widehat{\mathcal{E}}_\ell = \mathcal{E}_{\ell+1}$ in case of uniform refinement. Note that we only compute the error estimators for numerical experiments.
- (Line 88-90) The stopping criterion. We stop the iteration if the maximal number of elements is reached.
- (Line 92-93) To compute the error estimators, we need the "old" mesh $(\mathcal{T}_{\ell-1}, \mathcal{E}_{\ell-1})$.
- (Line 95-96) We uniformly refine the mesh, i.e. we generate $(\mathcal{T}_{\ell+1}, \mathcal{E}_{\ell+1})$. The time for mesh refinement is stored separately. The function `refineMesh` also prolongates the solution `x` to the refined mesh $(\mathcal{T}_{\ell+1}, \mathcal{E}_{\ell+1})$.

Listing 7: Adaptive and Uniform Algorithm, Part 2

```
1  if(method==1)
2      solveFEMBEM=@solveFEMBEMsymm;
3  elseif(method==2)
4      solveFEMBEM=@solveFEMBEMJohNed;
5  end
6
7  %*** load mesh data
8  load coordinates.dat
9  load elements.dat
10 load dirichlet.dat
11 %*** rotate the domain
12 theta=pi/2;
13 D=[cos(theta),-sin(theta);sin(theta),cos(theta)];
14 %*** delete first column (i.e. row numeration)
15 coordinates(:,1)=[];
16 elements(:,1)=[];
17 dirichlet(:,1)=[];
18 coordinates=(D*coordinates')';
19
```

```
%*** pre-refinement: #nPreRefinementSteps steps of uniform refinement
for i=1:nPreRefinementSteps
    [coordinates,elements,dirichlet] = ...
        refineMesh(coordinates,elements,dirichlet);
end
nMesh=0;
%%% adaptive processing routine
estvol1=[];
estvol2=[];
estdir=[];
estosc=[];
comptime=[];
refinetime=[];
save('err_est.mat','estvol1');
save('comptime.mat','comptime');
save('comptime.mat','refinetime','-append');
nC=size(coordinates,1);
nD=size(dirichlet,1);
ustart=rand(nC,1);
phistart=rand(nD,1);
mark1=0;
mark2=0;
systime=0;
while 1
    fprintf(['mesh_',int2str(nMesh),': number of elements=',...
        int2str(size(elements,1)),'\n']);
    if(adptrho>=1)
        %%%*************************
        %%%****UNIFORM-REFINEMENT****
        %%%*************************
        tic;
        [coordinates,elements,dirichlet,ustart] =...
            sortBoundaryNodesLast(coordinates,elements,dirichlet,ustart);
        systime=toc;
        %%% save data structures
        save(['mesh_' int2str(nMesh) '.mat'], 'coordinates');
        save(['mesh_' int2str(nMesh) '.mat'], 'elements','-append');
        save(['mesh_' int2str(nMesh) '.mat'], 'dirichlet','-append');
        tic;
        %%% Compute discrete solution
        if(lin)
            x=solveFEMBEM(coordinates,elements,dirichlet,@f,@u0,@grad_u0,...
                singularity,n_points_on_layer,n_layers,phi0_direct_or_projected);
        else
            x=solveFEMBEM(coordinates,elements,dirichlet,@f,@u0,@grad_u0,...
                singularity,n_points_on_layer,n_layers,phi0_direct_or_projected,...
                [ustart;phistart],@rho,@rhoprime);
        end
        systime=systime+toc;
        comptime=[comptime,systime];
        save('comptime.mat','comptime','-append');
```

```
71          %%% save solution
72          save(['mesh_' int2str(nMesh) '.mat'], 'x','-append');
73          if(nMesh>0)
74              %%% Compute error estimators for the previous step
75              [vol1,vol2,dir,osc]=computeHH2mu(x,coordinates,elements,...
76                  father2vol,father2dir,coordinatesold,dirichletold,@u0);
77              %***Save them for post-processing
78              estvol1=[estvol1,sum(vol1)];
79              estvol2=[estvol2,sum(vol2)];
80              estdir=[estdir,sum(dir)];
81              estosc=[estosc,sum(osc)];
82              save('err_est.mat','estvol1','-append');
83              save('err_est.mat','estvol2','-append');
84              save('err_est.mat','estdir','-append');
85              save('err_est.mat','estosc','-append');
86          end
87          %%% ---- Stopping criterion ----
88          if size(elements,1)>=nEmax
89              break;
90          end
91          %%%uniform refinement
92          coordinatesold=coordinates;
93          dirichletold=dirichlet;
94          tic;
95          [coordinates,elements,dirichlet,father2vol,father2dir,x] = ...
96              refineMesh(coordinates,elements,dirichlet,x);
97          refinetime=[refinetime,toc];
98          save('comptime.mat','refinetime','-append');
99          %%%prepare next step
100         nC=size(coordinates,1);
101         nD=size(dirichlet,1);
102         ustart=x(1:nC);
103         phistart=x(nC+1:end);
104     else
```

Now, we discuss the adaptive algorithm (Listing 8).

- We consider five different meshes: the original mesh $(\mathcal{T}_\ell, \mathcal{E}_\ell)$, its uniform refinement $(\widehat{\mathcal{T}}_\ell, \widehat{\mathcal{E}}_\ell)$, the old mesh $(\mathcal{T}_{\ell-1}, \mathcal{E}_{\ell-1})$, its uniform refinement $(\widehat{\mathcal{T}}_{\ell-1}, \widehat{\mathcal{E}}_{\ell-1})$, and the adaptive refinement of the original mesh $(\mathcal{T}_{\ell+1}, \mathcal{E}_{\ell+1})$.
- (Line 10-15) In case of a nonlinear problem, we need some data of the mesh $(\widehat{\mathcal{T}}_{\ell-1}, \widehat{\mathcal{E}}_{\ell-1})$ as well as its link to $(\mathcal{T}_{\ell-1}, \mathcal{E}_{\ell-1})$ stored in `father2coarse` and `father2coarse_dir`.
- (Line 16-17) We generate the mesh $(\widehat{\mathcal{T}}_\ell, \widehat{\mathcal{E}}_\ell)$.
- (Line 22-23) Again, we need the boundary nodes to be sorted last.
- (Line 24-31) The initial guess for the Newton method (`x_fine`) is generated. A random vector in the first step, and the prolongation of the solution w.r.t. $(\widehat{\mathcal{T}}_{\ell-1}, \widehat{\mathcal{E}}_{\ell-1})$

to $(\widehat{\mathcal{T}}_\ell, \widehat{\mathcal{E}}_\ell)$ in the remaining steps, see Section 11.6 for details.

- (Line 32-41) We compute the fine mesh solution $(\widehat{u}_\ell, \widehat{\phi}_\ell) \in \widehat{X}_\ell \times \widehat{Y}_\ell$.
- (Line 45-59) To give a fair comparison of adaptive and uniform strategy, we also compute the coarse mesh solution. We use the Dirichlet data from the fine mesh and project it onto the coarse mesh (cf. Section 11.7.1). Note that this step is not needed for the adaptive algorithm.
- (Line 61-68) Save the data for post-processing.
- (Line 71-72) We compute ϱ_ℓ for adaptive mesh refinement.
- (Line 76-84) Again, we save some data for post-processing.
- (Line 86-88) The algorithm is stopped if the maximal number of elements is reached.
- (Line 91) Determine the set $\mathcal{M}_\ell$ which satisfies Dörfler marking. The numbers of the elements in $\mathcal{M}_\ell \cap \mathcal{T}_\ell$ are stored in `mark1` as well as the numbers of the elements in $\mathcal{M}_\ell \cap \mathcal{E}_\ell$ are stored in `mark2`.
- (Line 94-95) We generate the adaptively refined mesh $(\mathcal{T}_{\ell+1}, \mathcal{E}_{\ell+1})$.

Listing 8: Adaptive and Uniform Algorithm, Part 3

```
1     else
2         %%%**************************
3         %%%****ADAPTIVE-REFINEMENT****
4         %%%**************************
5         tic;
6         nC_coarse=size(coordinates,1);
7         nD_coarse=size(dirichlet,1);
8         %%% Compute refinement indicators
9         %***Compute the h/2 solution
10        if(nMesh>0 && not(lin))
11            elements_coarse=elements_fine;
12            father2coarse=father2vol;
13            father2coarse_dir=father2dir;
14            nC_coarse2=size(coordinates_fine,1);
15        end
16        [coordinates_fine,elements_fine,dirichlet_fine,father2vol,father2dir] = ...
17            refineMesh(coordinates,elements,dirichlet);
18
19        nC_fine=size(coordinates_fine,1);
20        nD_fine=size(dirichlet_fine,1);
21
22        [coordinates_fine,elements_fine,dirichlet_fine] =...
23            sortBoundaryNodesLast(coordinates_fine,elements_fine,dirichlet_fine);
24        if(nMesh>0 && not(lin))
```

```
25              x_fine=prolongate(x_fine,elements_coarse,elements_fine,...
26                  father2coarse,father2adpt,father2vol,father2coarse_dir,...
27                  father2adpt_dir,father2dir,nC_coarse2,nC_fine,nD_fine);
28          else
29              x_fine=rand(nC_fine+nD_fine,1);
30              x=rand(nC_coarse+nD_coarse,1);
31          end
32          if(lin)
33             [x_fine,bdir_fine]=solveFEMBEM(coordinates_fine,elements_fine,...
34                 dirichlet_fine,@f,@u0,@grad_u0,singularity,n_points_on_layer,...
35                 n_layers,phi0_direct_or_projected);
36          else
37              [x_fine,bdir_fine]=solveFEMBEM(coordinates_fine,elements_fine,...
38                  dirichlet_fine,@f,@u0,@grad_u0,singularity,n_points_on_layer,...
39                  n_layers,phi0_direct_or_projected,x_fine,@rho, @rhoprime);
40          end
41
42          systime=systime+toc;
43
44          %***Compute coarse-mesh solution for postprocessing
45          [coordinates,elements,dirichlet] =...
46              sortBoundaryNodesLast(coordinates,elements,dirichlet);
47
48          bdir_coarse=computeRHScoarse(bdir_fine,dirichlet_fine,dirichlet,...
49              father2dir,nC_fine,nC_coarse);
50
51          if(lin)
52              x=solveFEMBEM(coordinates,elements,dirichlet,@f,@u0,@grad_u0,...
53                  singularity,n_points_on_layer,n_layers,...
54                  phi0_direct_or_projected,bdir_coarse);
55          else
56              x=solveFEMBEM(coordinates,elements,dirichlet,@f,@u0,@grad_u0,...
57                  singularity,n_points_on_layer,n_layers,phi0_direct_or_projected,...
58                  x,@rho, @rhoprime,bdir_coarse);
59          end
60           %%% save data structures
61          save(['mesh_' int2str(nMesh) '.mat'], 'coordinates');
62          save(['mesh_' int2str(nMesh) '.mat'], 'elements','-append');
63          save(['mesh_' int2str(nMesh) '.mat'], 'dirichlet','-append');
64          save(['mesh_' int2str(nMesh) '.mat'], 'coordinates_fine','-append');
65          save(['mesh_' int2str(nMesh) '.mat'], 'elements_fine','-append');
66          save(['mesh_' int2str(nMesh) '.mat'], 'dirichlet_fine','-append');
67          save(['mesh_' int2str(nMesh) '.mat'], 'x','-append');
68          save(['mesh_' int2str(nMesh) '.mat'], 'x_fine','-append');
69          tic;
70          %***Compute the error estimators
71          [vol1,vol2,dir,osc]=computeHH2mu(x_fine,coordinates_fine,elements_fine,...
72              father2vol,father2dir,coordinates,dirichlet,@u0,dirichlet_fine);
73          systime=systime+toc;
74          comptime=[comptime,systime];
75          %***Save them for post-processing
```

```
76          estvol1=[estvol1,sum(vol1)];
77          estvol2=[estvol2,sum(vol2)];
78          estdir=[estdir,sum(dir)];
79          estosc=[estosc,sum(osc)];
80          save('err_est.mat','estvol1','-append');
81          save('err_est.mat','estvol2','-append');
82          save('err_est.mat','estdir','-append');
83          save('err_est.mat','estosc','-append');
84          save('comptime.mat','comptime','-append');
85          %%% ---- Stopping criterion ----
86          if size(elements,1)>=nEmax
87              break;
88          end
89          tic;
90          %%% Mark elements for refinement
91          [mark1,mark2]=markElements(adptrho,vol1+vol2,dir+osc);
92
93          %%%Refine marked elements
94          [coordinates,elements,dirichlet,father2adpt,father2adpt_dir,x] = ...
95              refineMesh(coordinates,elements,dirichlet,mark1,mark2,x);
96          %[coordinates,elements,dirichlet,father2adpt,xadpt] = ...
97          %    refineMesh(coordinates,elements,dirichlet,mark1,mark2,x);
98          systime=toc;
99      end
```

11.6 Prolongate the Solution

Theory predicts quadratic convergence of the Newton method only locally. Thus, we try to determine the initial guess $\mathbf{x}_0$ as close to the solution as possible. The main effort in the adaptive algorithm goes into computing the solution w.r.t. the uniformly refined mesh $(\widehat{\mathcal{T}}_\ell, \widehat{\mathcal{E}}_\ell)$. One idea is to consider the preceding solution $(\widehat{u}_{\ell-1}, \widehat{\phi}_{\ell-1}) \in \widehat{X}_{\ell-1} \times \widehat{Y}_{\ell-1}$ and find its representation in $\widehat{X}_\ell \times \widehat{Y}_\ell$. Therefore, we need some link between the two meshes, which is given by the `father2son`-arrays as seen in Figure 5. The documentation of `prolongate` reads as follows (Listing 9).

- The `nC_coarse` $\times$ 3-array `elements_coarse` represents the mesh $\widehat{\mathcal{T}}_{\ell-1}$ as well as the `nC_fine` $\times$ 3-array `elements_fine` represents $\widehat{\mathcal{T}}_\ell$. `x` is the solution w.r.t. the coarse mesh. The `father2son`-arrays link the two meshes as shown in Figure 5. `nD_fine` denotes the number of boundary elements of the fine mesh $\widehat{\mathcal{E}}_\ell$. The output parameter `xnew` is a (`nC_fine` + `nD_fine`) $\times$ 1-array and contains the prolonged solution. Note that the function needs the respective arrays to be sorted by `sortBoundaryNodesLast` to work.
- (Line 5-13) We check the refinement type of the elements in $\mathcal{T}_{\ell-1}$. The `logical` arrays `none`, `bisec1`, `bisec12`, `bisec13`, and `bisec123` mark the elements with the respective refinement type.

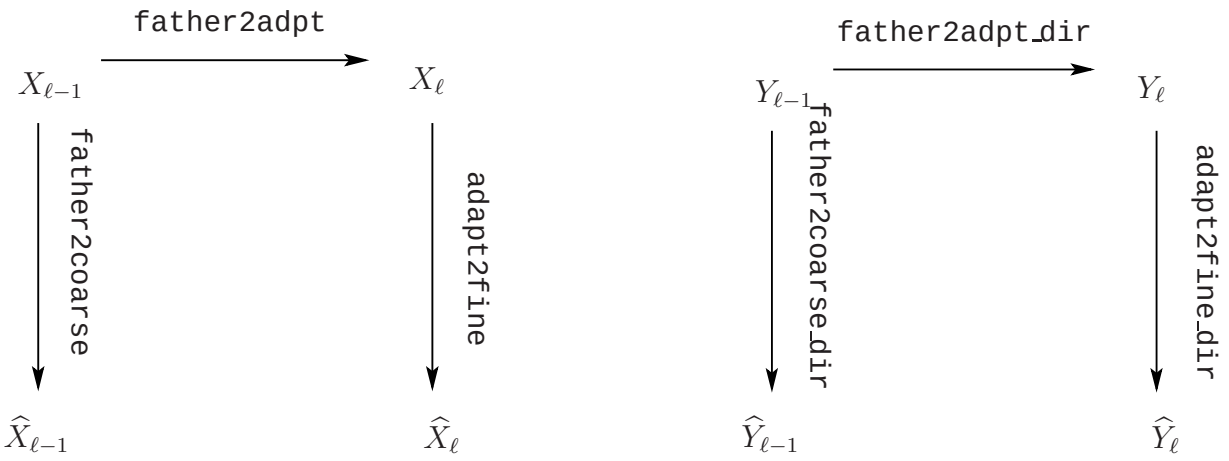


Figure 5: Illustration of the link between the different meshes used in `prolongate`.

- (Line 15) The `logical` array `bisecdir` marks the boundary elements of $\mathcal{E}_{\ell-1}$ which are refined in $\mathcal{E}_\ell$.
- We define $(\widehat{u}_{\text{new}}, \widehat{\phi}_{\text{new}})$ as

$$\widehat{u}_{\text{new}} = \sum_{i=1}^{\texttt{nC\_fine}} \texttt{xnew(i)}\eta_i \quad \text{and} \quad \widehat{\phi}_{\text{new}} = \sum_{i=\texttt{nC\_fine}+1}^{\texttt{nC\_fine+nD\_fine}} \texttt{xnew(i)}\xi_{i-\texttt{nC\_fine}}.$$

- (Line 14-39) We prolongate the solution on the boundary. If the element $\tau \in \mathcal{E}_{\ell-1}$ hasn't been adaptively refined, we know that τ is refined in two elements $\tau_1, \tau_2 \in \widehat{\mathcal{E}}_{\ell-1} \cap \widehat{\mathcal{E}}_\ell$. Consequently, we assign the values of `x` on the elements τ_1 and τ_2 to `xnew`. If the element $\tau \in \mathcal{E}_{\ell-1}$ has been refined, we obtain the following situation:

$$\begin{aligned} \overline{\tau} &= \overline{\tau_1} \cup \overline{\tau_2} \quad \tau_1, \tau_2 \in \widehat{\mathcal{E}}_{\ell-1}, \\ \overline{\tau_i} &= \overline{\tau_1^i} \cup \overline{\tau_2^i}, \quad \tau_1^i, \tau_2^i \in \widehat{\mathcal{E}}_\ell, \ i = 1, 2. \end{aligned}$$

  Due to the geometric situation, we assign a weighted sum to `xnew` on the elements τ_2^1 and τ_1^2 , i.e.

$$\begin{aligned} \widehat{\phi}_{\text{new}}\Big|_{\tau_2^1} &= \frac{3}{4}\,\widehat{\phi}_{\ell-1}\Big|_{\tau_1} + \frac{1}{4}\,\widehat{\phi}_{\ell-1}\Big|_{\tau_2}, \\ \widehat{\phi}_{\text{new}}\Big|_{\tau_1^2} &= \frac{1}{4}\,\widehat{\phi}_{\ell-1}\Big|_{\tau_1} + \frac{3}{4}\,\widehat{\phi}_{\ell-1}\Big|_{\tau_2}. \end{aligned}$$

- (Line 47-58) The values of $\widehat{u}_\ell$ on the elements of $\mathcal{T}_{\ell-1}$ which haven't been adaptively refined are the same as the values of $\widehat{u}_{\ell-1}$.
- (Line 59-200) The rest of the code considers the elements which have been adaptively refined. Any new node $\widehat{V}$ of $\widehat{\mathcal{T}}_\ell$ is located on an edge between the nodes V_1, V_2 of the

coarse mesh $\widehat{\mathcal{T}}_{\ell-1}$, cf. Figure 4. Consequently, we interpolate the values, i.e.

$$\widehat{u}_{\text{new}}(V) = \frac{1}{2}\left(\widehat{u}_{\ell-1}(V_1) + \widehat{u}_{\ell-1}(V_2)\right).$$

- We give an example for an element $T_s \in \mathcal{T}_{\ell-1}$ whose first and second edge is refined in $\mathcal{T}_\ell$, i.e. `bisec12(s)` $= 1$ (cf. Figure 6). We want to generate $\widehat{u}_{\text{new}}$ on the element in $\widehat{\mathcal{T}}_\ell$ with the known values of $\widehat{u}_{\ell-1}$ on the element in $\widehat{\mathcal{T}}_{\ell-1}$. Therefore, we will discuss two exemplary situations.

  - To generate the value of $\widehat{u}_{\text{new}}$ on the blue marked node in $\widehat{\mathcal{T}}_\ell$, we just take the value of $\widehat{u}_{\ell-1}$ on the corresponding blue marked node of $\mathcal{T}_{\ell-1}$. To find the entry in $\widehat{u}_{\ell-1}$ resp. `x`, we start with the array `father2coarse`. As seen in Figure 4 the value of $\widehat{u}_{\ell-1}$ of the blue marked node in $\widehat{\mathcal{T}}_{\ell-1}$ can be obtained by

    ```
    x(elements_coarse(father2coarse(s,1),3)).
    ```
    (77)

    We proceed with the value of $\widehat{u}_{\text{new}}$ on the blue marked node in $\widehat{\mathcal{T}}_\ell$. `father2adpt(s,1)` is the number of the blue marked element in $\mathcal{T}_\ell$ and `adapt2fine(father2adpt(s,1),2)` contains the number of the blue marked element in $\widehat{\mathcal{T}}_\ell$. Thus, the value in (77) has to be stored in

    ```
    xnew(elements_fine(adapt2fine(father2adpt(s,1),2),2)),
    ```

    cf. Line 92-93 and see also Figure 4 for the position of the nodes.

  - To generate the value of $\widehat{u}_{\text{new}}$ on the green marked node in $\widehat{\mathcal{T}}_\ell$, we have to interpolate the values of $\widehat{u}_{\ell-1}$ on the green marked nodes in $\widehat{\mathcal{T}}_{\ell-1}$. As before, we navigate to the green marked nodes in $\widehat{\mathcal{T}}_{\ell-1}$ with

    ```
    xnew(elements_coarse(father2coarse(s,4),1)) and
    xnew(elements_coarse(father2coarse(s,4),3)).
    ```

    `father2adpt(s,4)` contains the number of the green marked element in $\mathcal{T}_\ell$ and `adapt2fine(father2adpt(s,4),1)` contains the number of the green marked element in $\widehat{\mathcal{T}}_\ell$. Consequently, we may generate the value of $\widehat{u}_{\text{new}}$ on the green marked node in $\widehat{\mathcal{T}}_\ell$ as follows

    ```
    xnew(elements_fine(adapt2fine(father2adpt(s,4),1),3)) =
          1/2 (xnew(elements_coarse(father2coarse(s,4),1))
          +xnew(elements_coarse(father2coarse(s,4),3))),
    ```

    cf. Line 119-121.

Listing 9: Prolongate the Solution

```
1 function xnew=prolongate(x,elements_coarse,elements_fine,father2coarse,...
2     father2adpt,adapt2fine,father2coarse_dir,father2adpt_dir,...
3     adapt2fine_dir,nC_coarse,nC_fine,nD_fine)
```

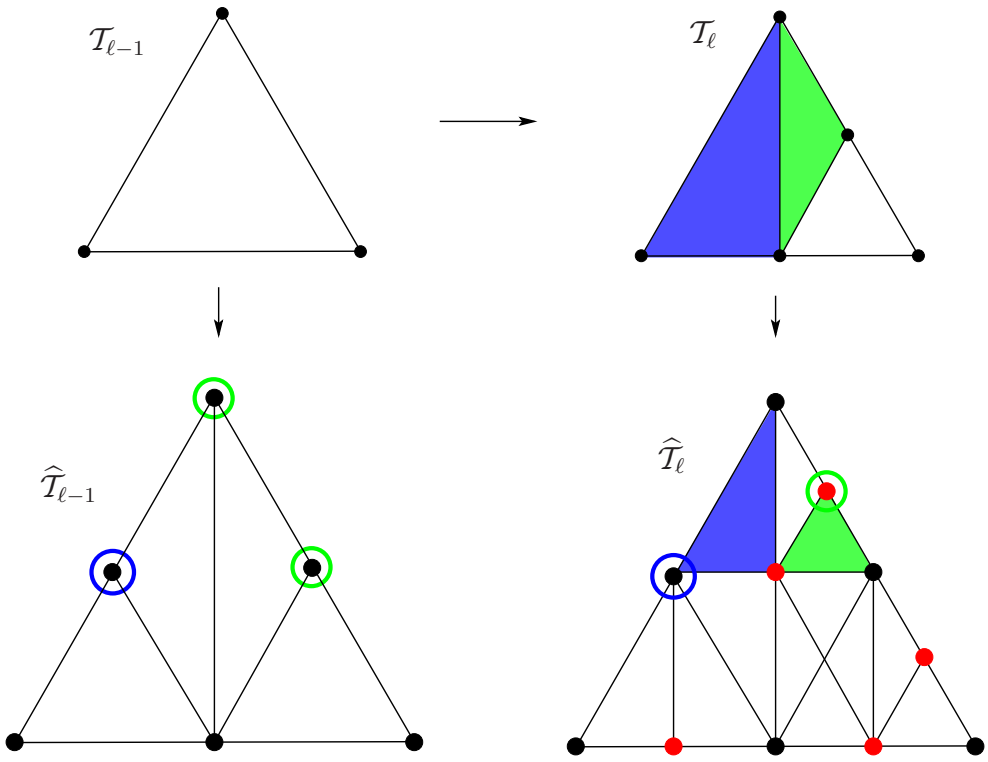


Figure 6: Example for prolongation.

```
%***Check refinement type in adaptive refined mesh
none=(father2adpt(:,1)==father2adpt(:,2)) & (father2adpt(:,3)==father2adpt(:,4))...
    & (father2adpt(:,2)==father2adpt(:,3));
bisec1=(father2adpt(:,1)==father2adpt(:,2)) &...
    (father2adpt(:,3)==father2adpt(:,4)) & ~none;
bisec12=(father2adpt(:,1)==father2adpt(:,2)) &...
    (father2adpt(:,3) ~= father2adpt(:,4));
bisec13=(father2adpt(:,1) ~= father2adpt(:,2)) &...
    (father2adpt(:,3)==father2adpt(:,4));
bisec123=not(bisec1 | bisec12  | bisec13 | none);

bisecdir=(father2adpt_dir(:,1) ~= father2adpt_dir(:,2));

xnew=zeros(nC_fine+nD_fine,1);

%%%************************************************
%%% prolongate x on dirichlet mesh, i.e x(nC+1:end)
%%%************************************************

%***no adaptive refinement
xnew(nC_fine+adapt2fine_dir(father2adpt_dir(~bisecdir,1),1))=...
    x(nC_coarse+father2coarse_dir(~bisecdir,1));
xnew(nC_fine+adapt2fine_dir(father2adpt_dir(~bisecdir,1),2))=...
    x(nC_coarse+father2coarse_dir(~bisecdir,2));
```

```
28
29  %***bisection of the boundary elements
30  xnew(nC_fine+adapt2fine_dir(father2adpt_dir(bisecdir,1),1))=...
31      x(nC_coarse+father2coarse_dir(bisecdir,1));
32  xnew(nC_fine+adapt2fine_dir(father2adpt_dir(bisecdir,1),2))=...
33      0.25*(3*x(nC_coarse+father2coarse_dir(bisecdir,1))...
34      +x(nC_coarse+father2coarse_dir(bisecdir,2)));
35  xnew(nC_fine+adapt2fine_dir(father2adpt_dir(bisecdir,2),1))=...
36      0.25*(x(nC_coarse+father2coarse_dir(bisecdir,1))...
37      +3*x(nC_coarse+father2coarse_dir(bisecdir,2)));
38  xnew(nC_fine+adapt2fine_dir(father2adpt_dir(bisecdir,2),2))=...
39      x(nC_coarse+father2coarse_dir(bisecdir,2));
40
41
42  %%%************************************************
43  %%% prolongate x on volume mesh, i.e x(1:nC)
44  %%%************************************************
45
46  %***no adaptive refinement
47  xnew(elements_fine(adapt2fine(father2adpt(none,1),1),2))=...
48      x(elements_coarse(father2coarse(none,1),2));
49  xnew(elements_fine(adapt2fine(father2adpt(none,1),1),3))=...
50      x(elements_coarse(father2coarse(none,1),3));
51  xnew(elements_fine(adapt2fine(father2adpt(none,1),2),1))=...
52      x(elements_coarse(father2coarse(none,2),1));
53  xnew(elements_fine(adapt2fine(father2adpt(none,1),2),2))=...
54      x(elements_coarse(father2coarse(none,2),2));
55  xnew(elements_fine(adapt2fine(father2adpt(none,1),3),2))=...
56      x(elements_coarse(father2coarse(none,3),2));
57  xnew(elements_fine(adapt2fine(father2adpt(none,1),3),3))=...
58      x(elements_coarse(father2coarse(none,3),3));
59
60  %***first edge refined
61  xnew(elements_fine(adapt2fine(father2adpt(bisec1,1),2),1))=...
62      x(elements_coarse(father2coarse(bisec1,1),2));
63  xnew(elements_fine(adapt2fine(father2adpt(bisec1,1),2),2))=...
64      x(elements_coarse(father2coarse(bisec1,1),3));
65  xnew(elements_fine(adapt2fine(father2adpt(bisec1,1),3),2))=...
66      x(elements_coarse(father2coarse(bisec1,2),1));
67  xnew(elements_fine(adapt2fine(father2adpt(bisec1,1),4),1))=...
68      x(elements_coarse(father2coarse(bisec1,4),2));
69
70  xnew(elements_fine(adapt2fine(father2adpt(bisec1,3),2),1))=...
71      x(elements_coarse(father2coarse(bisec1,3),2));
72  xnew(elements_fine(adapt2fine(father2adpt(bisec1,3),2),2))=...
73      x(elements_coarse(father2coarse(bisec1,3),3));
74  xnew(elements_fine(adapt2fine(father2adpt(bisec1,3),3),2))=...
75      x(elements_coarse(father2coarse(bisec1,4),1));
76  xnew(elements_fine(adapt2fine(father2adpt(bisec1,3),4),1))=...
77      x(elements_coarse(father2coarse(bisec1,4),2));
78
```

```
xnew(elements_fine(adapt2fine(father2adpt(bisec1,1),1),3))=...
    0.5*(x(elements_coarse(father2coarse(bisec1,1),1))...
    +x(elements_coarse(father2coarse(bisec1,1),2)));
xnew(elements_fine(adapt2fine(father2adpt(bisec1,1),4),3))=...
    0.5*(x(elements_coarse(father2coarse(bisec1,2),1))...
    +x(elements_coarse(father2coarse(bisec1,2),2)));
xnew(elements_fine(adapt2fine(father2adpt(bisec1,3),1),3))=...
    0.5*(x(elements_coarse(father2coarse(bisec1,3),1))...
    +x(elements_coarse(father2coarse(bisec1,3),2)));

%***first and second edge refined
xnew(elements_fine(adapt2fine(father2adpt(bisec12,1),2),1))=...
    x(elements_coarse(father2coarse(bisec12,1),2));
xnew(elements_fine(adapt2fine(father2adpt(bisec12,1),2),2))=...
    x(elements_coarse(father2coarse(bisec12,1),3));
xnew(elements_fine(adapt2fine(father2adpt(bisec12,1),3),2))=...
    x(elements_coarse(father2coarse(bisec12,2),1));
xnew(elements_fine(adapt2fine(father2adpt(bisec12,1),4),1))=...
    x(elements_coarse(father2coarse(bisec12,4),2));

xnew(elements_fine(adapt2fine(father2adpt(bisec12,3),3),2))=...
    x(elements_coarse(father2coarse(bisec12,3),2));
xnew(elements_fine(adapt2fine(father2adpt(bisec12,3),4),1))=...
    x(elements_coarse(father2coarse(bisec12,3),3));

xnew(elements_fine(adapt2fine(father2adpt(bisec12,1),1),3))=...
    0.5*(x(elements_coarse(father2coarse(bisec12,1),1))...
    +x(elements_coarse(father2coarse(bisec12,1),2)));
xnew(elements_fine(adapt2fine(father2adpt(bisec12,1),3),3))=...
    0.5*(x(elements_coarse(father2coarse(bisec12,2),1))...
    +x(elements_coarse(father2coarse(bisec12,2),2)));
xnew(elements_fine(adapt2fine(father2adpt(bisec12,3),1),1))=...
    0.5*(x(elements_coarse(father2coarse(bisec12,3),1))...
    +x(elements_coarse(father2coarse(bisec12,3),2)));
xnew(elements_fine(adapt2fine(father2adpt(bisec12,3),1),3))=...
    0.5*(x(elements_coarse(father2coarse(bisec12,3),1))...
    +x(elements_coarse(father2coarse(bisec12,3),3)));
xnew(elements_fine(adapt2fine(father2adpt(bisec12,3),4),3))=...
    0.5*(x(elements_coarse(father2coarse(bisec12,3),3))...
    +x(elements_coarse(father2coarse(bisec12,3),2)));
xnew(elements_fine(adapt2fine(father2adpt(bisec12,4),1),3))=...
    0.5*(x(elements_coarse(father2coarse(bisec12,4),3))...
    +x(elements_coarse(father2coarse(bisec12,4),1)));

%***first and third edge refined
xnew(elements_fine(adapt2fine(father2adpt(bisec13,1),3),2))=...
    x(elements_coarse(father2coarse(bisec13,1),2));
xnew(elements_fine(adapt2fine(father2adpt(bisec13,1),1),2))=...
    x(elements_coarse(father2coarse(bisec13,1),3));
xnew(elements_fine(adapt2fine(father2adpt(bisec13,1),2),1))=...
    x(elements_coarse(father2coarse(bisec13,2),2));
```

```
130
131 xnew(elements_fine(adapt2fine(father2adpt(bisec13,2),2),1))=...
132     x(elements_coarse(father2coarse(bisec13,2),1));
133
134 xnew(elements_fine(adapt2fine(father2adpt(bisec13,3),2),1))=...
135     x(elements_coarse(father2coarse(bisec13,3),2));
136 xnew(elements_fine(adapt2fine(father2adpt(bisec13,3),2),2))=...
137     x(elements_coarse(father2coarse(bisec13,3),3));
138
139 xnew(elements_fine(adapt2fine(father2adpt(bisec13,1),1),1))=...
140     0.5*(x(elements_coarse(father2coarse(bisec13,1),1))...
141     +x(elements_coarse(father2coarse(bisec13,1),2)));
142 xnew(elements_fine(adapt2fine(father2adpt(bisec13,2),1),1))=...
143     0.5*(x(elements_coarse(father2coarse(bisec13,2),1))...
144     +x(elements_coarse(father2coarse(bisec13,2),2)));
145 xnew(elements_fine(adapt2fine(father2adpt(bisec13,3),1),3))=...
146     0.5*(x(elements_coarse(father2coarse(bisec13,3),1))...
147     +x(elements_coarse(father2coarse(bisec13,3),2)));
148 xnew(elements_fine(adapt2fine(father2adpt(bisec13,1),4),3))=...
149     0.5*(x(elements_coarse(father2coarse(bisec13,1),3))...
150     +x(elements_coarse(father2coarse(bisec13,1),2)));
151 xnew(elements_fine(adapt2fine(father2adpt(bisec13,1),1),3))=...
152     0.5*(x(elements_coarse(father2coarse(bisec13,1),1))...
153     +x(elements_coarse(father2coarse(bisec13,1),3)));
154 xnew(elements_fine(adapt2fine(father2adpt(bisec13,2),1),3))=...
155     0.5*(x(elements_coarse(father2coarse(bisec13,2),1))...
156     +x(elements_coarse(father2coarse(bisec13,2),3)));
157
158 %***all edges refined
159 xnew(elements_fine(adapt2fine(father2adpt(bisec123,1),3),2))=...
160     x(elements_coarse(father2coarse(bisec123,1),2));
161 xnew(elements_fine(adapt2fine(father2adpt(bisec123,1),1),2))=...
162     x(elements_coarse(father2coarse(bisec123,1),3));
163 xnew(elements_fine(adapt2fine(father2adpt(bisec123,1),2),1))=...
164     x(elements_coarse(father2coarse(bisec123,2),2));
165
166 xnew(elements_fine(adapt2fine(father2adpt(bisec123,2),2),1))=...
167     x(elements_coarse(father2coarse(bisec123,2),1));
168
169 xnew(elements_fine(adapt2fine(father2adpt(bisec123,3),3),2))=...
170     x(elements_coarse(father2coarse(bisec123,3),2));
171 xnew(elements_fine(adapt2fine(father2adpt(bisec123,3),4),1))=...
172     x(elements_coarse(father2coarse(bisec123,3),3));
173
174 xnew(elements_fine(adapt2fine(father2adpt(bisec123,1),1),1))=...
175     0.5*(x(elements_coarse(father2coarse(bisec123,1),1))...
176     +x(elements_coarse(father2coarse(bisec123,1),2)));
177 xnew(elements_fine(adapt2fine(father2adpt(bisec123,2),1),1))=...
178     0.5*(x(elements_coarse(father2coarse(bisec123,2),1))...
179     +x(elements_coarse(father2coarse(bisec123,2),2)));
180 xnew(elements_fine(adapt2fine(father2adpt(bisec123,1),4),3))=...
```

```
181     0.5*(x(elements_coarse(father2coarse(bisec123,1),3))...
182     +x(elements_coarse(father2coarse(bisec123,1),2)));
183 xnew(elements_fine(adapt2fine(father2adpt(bisec123,1),1),3))=...
184     0.5*(x(elements_coarse(father2coarse(bisec123,1),1))...
185     +x(elements_coarse(father2coarse(bisec123,1),3)));
186 xnew(elements_fine(adapt2fine(father2adpt(bisec123,2),1),3))=...
187     0.5*(x(elements_coarse(father2coarse(bisec123,2),1))...
188     +x(elements_coarse(father2coarse(bisec123,2),3)));
189 xnew(elements_fine(adapt2fine(father2adpt(bisec123,3),1),3))=...
190     0.5*(x(elements_coarse(father2coarse(bisec123,3),1))...
191     +x(elements_coarse(father2coarse(bisec123,3),3)));
192 xnew(elements_fine(adapt2fine(father2adpt(bisec123,3),1),1))=...
193     0.5*(x(elements_coarse(father2coarse(bisec123,3),1))...
194     +x(elements_coarse(father2coarse(bisec123,3),2)));
195 xnew(elements_fine(adapt2fine(father2adpt(bisec123,3),4),3))=...
196     0.5*(x(elements_coarse(father2coarse(bisec123,3),2))...
197     +x(elements_coarse(father2coarse(bisec123,3),3)));
198 xnew(elements_fine(adapt2fine(father2adpt(bisec123,4),1),3))=...
199     0.5*(x(elements_coarse(father2coarse(bisec123,4),3))...
200     +x(elements_coarse(father2coarse(bisec123,4),1)));
```

11.7 Auxiliary Functions

11.7.1 `computeRHScoarse`

This function is used to compute the right hand side of the Symmetric Coupling Method on a coarse mesh, but with the discrete integral operators computed on a uniformly refined mesh. We use it in Section 11.5. The documentation reads as follows (Listing 10).

- The function is called with the array `dirichlet` representing the coarse mesh $\mathcal{E}_\ell$, the array `dirichlet_fine` representing the uniformly refined mesh $\widehat{\mathcal{E}}_\ell$, the number of the nodes of the respective meshes $\mathcal{T}_\ell$, $\widehat{\mathcal{T}}_\ell$, and the Dirichlet part of the fine-mesh right hand side `b_fine`. To obtain `b_fine`, use the function `solveFEMBEMsymm` (Section 11.2).

- (Line 5-11) We determine the number of the first node V of $\mathcal{T}_\ell$ or $\widehat{\mathcal{T}}_\ell$ with $V \in \Gamma$. This information is stored in `fBN_coarse` and `fBN_fine`.

- (Line 13-15) We build the index field `coarse2fine`. With `k=coarse2fine(j)`, there holds that the j-th boundary node of $\mathcal{T}_\ell$ has the number k in $\widehat{\mathcal{T}}_\ell$.

- (Line 19) Assume that the boundary element E_s of the coarse mesh $\mathcal{E}_\ell$ has the endpoints V_1 and V_2. We know that $V_m := \frac{1}{2}(V_1 + V_2)$ is a node in $\widehat{\mathcal{T}}_\ell$. The index field `dirichlet2midpoint` contains this information, i.e.

  `m=dirichlet2midpoint(s)`,

  where m is the number of the node V_m and s is the number of the element E_s.

- (Line 23-26) For a node V_k of the fine mesh $\widehat{\mathcal{T}}_\ell$ exist unique elements E_i, E_j with

$$V_k = \overline{E_i} \cap \overline{E_j} \quad E_i, E_j \in \mathcal{E}_\ell.$$

  Consequently, we build the index field `node2dirichlet` as `[i,j]=node2dirichlet(k)`.

- (Line 29-33) With the index fields defined above, one can represent a test function $\eta_i \in \mathcal{S}^1(\mathcal{T}_\ell)$ as follows

$$\eta_i|_\Gamma = \widehat{\eta}_k|_\Gamma + \frac{1}{2}\,\widehat{\eta}_m|_\Gamma + \frac{1}{2}\,\widehat{\eta}_n|_\Gamma \quad \widehat{\eta}_k, \widehat{\eta}_m, \widehat{\eta}_n \in \mathcal{S}^1(\widehat{\mathcal{T}}_\ell),$$

  where `k=coarse2fine(i)` and `[m,n]=dirichlet2midpoint(node2dirichlet(i))`. A test function $\xi_j \in \mathcal{P}^0(\mathcal{E}_\ell)$ can be represented as

$$\xi_j = \widehat{\xi}_p + \widehat{\xi}_q \quad \widehat{\xi}_p, \widehat{\xi}_q \in \mathcal{P}^0(\widehat{\mathcal{E}}_\ell),$$

  where `[p,q]=node2dirichlet(dirichlet2midpoint(i))`.

Listing 10: `computeRHScoarse`

```
function b_coarse=computeRHScoarse(b_fine,dirichlet_fine,dirichlet,...
    father2dir,nC_fine,nC_coarse)
%*** build index field k = coarse2fine (j) such that j-th node of coarse
%*** mesh coincides with k-th nodes of fine mesh
boundary_nodes=unique(dirichlet);
nBN_coarse=length(boundary_nodes);
fBN_coarse=nC_coarse-nBN_coarse+1;

boundary_nodes=unique(dirichlet_fine);
nBN_fine=length(boundary_nodes);
fBN_fine=nC_fine-nBN_fine+1;

coarse2fine = zeros(nBN_coarse,1);
coarse2fine(dirichlet-fBN_coarse+1) = [ dirichlet_fine(father2dir(:,1),1) ...
    dirichlet_fine(father2dir(:,2),2) ]-fBN_fine+1;

%*** build index field k = element2midpoint(j) such that midpoint of j-th
%*** element of coarse mesh is k-th node of fine mesh
dirichlet2midpoint = dirichlet_fine(father2dir(:,1),2)-fBN_fine+1;

%*** build index field [i,j] = node2element(k) such that
%*** intersection(Ei,Ej) = zk
nD_ = size(dirichlet,1);
node2dirichlet = zeros(nBN_coarse,2);
node2dirichlet(dirichlet(:,1)-fBN_coarse+1,2) = (1:nD_)';
node2dirichlet(dirichlet(:,2)-fBN_coarse+1,1) = (1:nD_)';

```

```
%*** obtain coarse-mesh RHS from fine-mesh RHS
b_coarse=zeros(nBN_coarse+nD_,1);
b_coarse(1:nBN_coarse) = b_fine(coarse2fine) + ...
    0.5*sum(b_fine(dirichlet2midpoint(node2dirichlet)),2);
b_coarse(nBN_coarse+1:end)=(b_fine(nBN_fine+father2dir(:,1))+...
    b_fine(nBN_fine+father2dir(:,2)));
```

11.7.2 Compute $\langle \phi_0 , \eta_i \rangle_\Gamma$

There are two strategies to compute the quantity $\langle \phi_0 , \eta_i \rangle_\Gamma$. The first idea is to compute the integral $\int_\Gamma \phi_0(y)\, \eta_i|_\Gamma\,(y)\, \mathrm{d}\Gamma(y)$ directly with some quadrature rule. The function `computePhi0Rhs` follows this approach. The function is called with

```
phi0_rhs=computePhi0Rhs(coordinates,dirichlet,...
gradient_u0,singularity,n_points_on_layer,n_layers)
```

The arrays `coordinates` and `dirichlet` represent the boundary mesh $\mathcal{E}_\ell$. `gradient_u0` is a function handle for the function ϕ_0 from (59). The variables `singularity`, `n_points_on_layer`, and `n_layers` steer the quadrature rule. The output of the function is the `nD` $\times$ 1-array `phi0_rhs` which contains

$$\texttt{phi0\_rhs(i)} = \langle \phi_o , \eta_{i+\texttt{firstBoundaryNode}} \rangle_\Gamma.$$

Another idea is to replace ϕ_0 by its L^2-orthogonal projection on $\mathcal{P}^0(\mathcal{E}_\ell)$, as implemented in the function `computeProjPhi0`. The function is called with

```
proj_phi0=computeProjPhi0(coordinates,dirichlet,...
gradient_u0,singularity,n_points_on_layer,n_layers)
```

Again, the arrays `coordinates` and `dirichlet` represent the boundary mesh $\mathcal{E}_\ell$. `gradient_u0` is a function handle for the function ϕ_0 from (59). The variables `singularity`, `n_points_on_layer`, and `n_layers` steer the quadrature rule. The output of the function is the `nD` $\times$ 1-array `proj_phi0` which contains

$$\texttt{proj\_phi0(s)} = \mathrm{length}(E_s)^{-1} \langle \phi_0 , 1 \rangle_{E_s}.$$

Consequently, we approximate

$$\langle \phi_0 , \eta_i \rangle \approx \frac{1}{2}(\mathrm{length}(E_n)\texttt{proj\_phi0(n)} + \mathrm{length}(E_m)\texttt{proj\_phi0(m)}),$$

where $E_n, E_m \in \mathcal{E}_\ell$ with $\overline{E_n \cup E_m} = \mathrm{supp}(\eta_i) \cap \Gamma$.
If we follow the second idea, we have to consider a different data oscillation term in Section 11.4. Thus, for numerical experiments, we prefer the function `computePhi0Rhs`.

11.7.3 Sorting Nodes

In view of solving the linear systems which arise in Section 11.2 and 11.3, we want to study the structure of the corresponding matrices. In both cases, one has to solve

$$\mathcal{J}(\mathbf{x}_k)z = f \quad z, f \in \mathbb{R}^{n+m},$$

to obtain the vector $\mathcal{J}(\mathbf{x}_k)^{-1}f$. Here, f denotes the respective right-hand sides in (68) and (76). The matrix $\mathcal{J}(\mathbf{x}_k)$ has the form

$$\mathcal{J}(\mathbf{x}) = \begin{pmatrix} J(\mathbf{x}) & 0 \\ 0 & 0 \end{pmatrix} + B \in \mathbb{R}^{(n+m)\times(n+m)}.$$

$J(\mathbf{x})$ is the FEM-Galerkin sparse matrix with small bandwidth. Thus, MATLAB's backslash-operator should not have any problems. However, B is a dense matrix. To overcome this problem, we sort the nodes V of the mesh $\mathcal{T}_\ell$ which are located on the boundary, i.e. $V \in \Gamma$, last. There holds, $\eta_i|_\Gamma = 0$ for any node $V_i \notin \Gamma$, therefore B has the following structure:

$$B = \begin{pmatrix} 0 & 0 \\ 0 & \widetilde{B} \end{pmatrix}, \tag{78}$$

where $\widetilde{B} \in \mathbb{R}^{2m\times 2m}$ (see the definition of B in (64) and (74)). Note that there holds approximately $m \approx \sqrt{n}$, thus $2m \times 2m \ll (m+n)\times(m+n)$. The main advantage is that the structure of B in (78) is optimal for fill-in, i.e. the LU-decomposition maintains the structure of B.
Namely, we want to reach the following numbering of the nodes.

- The nodes V with $V \in \Omega^{int} \setminus \Gamma$ are denoted by

$$V_1, \ldots, V_{\texttt{firstBoundaryNode}-1}.$$

- The nodes V with $V \in G$ are denoted by

$$V_{\texttt{firstBoundaryNode}}, \ldots, V_{\texttt{nC}}.$$

There holds, `firstBoundaryNode=nC-nBN+1`. We proceed with the documentation of `sortBoundaryNodesLast` (Listing 11).

- There are two ways to call the function
  - `[coordinates,elements,dirichlet]=...`
    `sortBoundaryNodesLast(coordinates,elements,dirichlet)`
  - `[coordinates,elements,dirichlet,u_sorted]=...`
    `sortBoundaryNodesLast(coordinates,elements,dirichlet,u)`

The arrays `coordinates`, `elements`, and `dirichlet` represent the mesh $(\mathcal{T}_\ell, \mathcal{E}_\ell)$. The optional `nC` $\times$ 1-array `u` contains the first `nC` entries of the array `x`. Because these entries correspond to the nodes, we have to sort them to. (e.g. used to generate the initial guess for the Newton iteration, see Section 11.5)

- (Line 4-6) We determine the number of boundary nodes `nBN`. The array `boundary_nodes` contains the numbers of all nodes V with $V \in \Gamma$.
- (Line 9-10) We determine the nodes which are located on Γ and provide the index field `coordinates2newcoordinates` to link the old with the new numbers of the nodes.
- (Line 13) The array `coordinates` is reordered.
- (Line 15-18) If the parameter `u` is given, we reorder it, too.
- (Line 20-21) The arrays `elements` and `dirichlet` refer to the nodes, thus we have to update the numbers.

Listing 11: Sorting Nodes

```
1  function [coordinates,elements,dirichlet,varargout]=...
2    sortBoundaryNodesLast(coordinates,elements,dirichlet,varargin)
3  %%% mesh quantities
4  nC=size(coordinates,1);
5  boundary_nodes=unique(dirichlet);
6  nBN=length(boundary_nodes);
7
8  %%% renumeration of nodes
9  ind=[setdiff([1:nC],boundary_nodes)';boundary_nodes];
10 coordinates2newCoordinates(ind)=[[1:(nC-nBN)] [(nC-nBN+1):nC]];
11
12 %%% reorder coordinates
13 coordinates=coordinates(ind,:);
14 %%% reorder coordinates from a prolonged solution xold
15 if(nargin==4)
16    u=varargin{1};
17    varargout{1}=u(ind);
18 end
19 %%% assign the new node numbers to elements and dirichlet
20 elements=coordinates2newCoordinates(elements);
21 dirichlet=coordinates2newCoordinates(dirichlet);
```

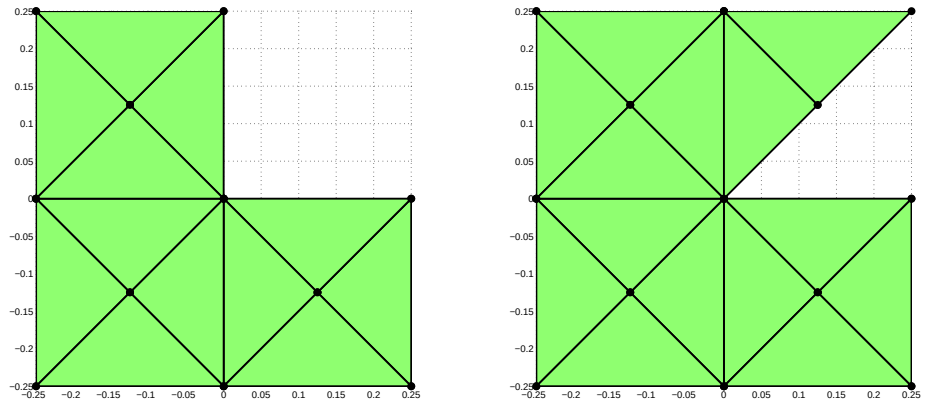


Figure 7: L-shaped (left) and Z-shaped domain (right) and initial triangulations $\mathcal{T}_0$ for the numerical experiments.

12 Numerical Experiments

In this section, we present three numerical examples from [6] to demonstrate the performance of symmetric FEM-BEM coupling. Recall the definition of (u_ℓ, ϕ_ℓ), (u, ϕ), (u_ℓ^*, ϕ_ℓ^*), as well as osc_ℓ and $u_{0,\ell}$ from Section 8.2. In all experiments, we prescribe the exact solution $(u^{\mathrm{int}}, u^{\mathrm{ext}})$ of the transmission problem (59), and the data (u_0, ϕ_0, f) are computed thereof. The error is measured in the energy norm

$$|||(u,\phi)||| := \|u\|_{H^1(\Omega^{int})} + \langle \phi\,, V\phi\rangle_\Gamma \quad \text{for all } (u,\phi) \in H^1(\Omega^{int}) \times H^{-1/2}(\Gamma),$$

which is an equivalent norm on the product space $H^1(\Omega^{int}) \times H^{-1/2}(\Gamma)$. This is a simple consequence of the continuity and ellipticity of V. Note that the contribution $\langle \phi - \phi_\ell\,, V(\phi - \phi_\ell)\rangle_\Gamma$ to the error $|||(u^i - u_\ell, \phi - \phi_\ell)|||$ can hardly be computed analytically. However, according to Proposition 8.7 and the quasi optimality result in Theorem 7.10, there holds

$$\begin{aligned} |||(u-u_\ell, \phi-\phi_\ell)||| &\lesssim |||(u-u_\ell^\star, \phi-\phi_\ell^\star)||| + \mathrm{osc}_\ell \\ &\lesssim \|u-u_\ell\|_{H^1(\Omega^{int})} + \min_{\psi_\ell \in \mathcal{P}^0(\mathcal{E}_\ell)} \langle \phi - \psi_\ell\,, V(\phi-\psi_\ell)\rangle_\Gamma + \mathrm{osc}_\ell. \end{aligned}$$

In all experiments, the exterior normal derivative has additional regularity $\phi \in L^2(\Gamma)$. We may therefore use an approximation result from [5, Theorem 4.1, Lemma 4.3] to obtain

$$\begin{aligned} \min_{\psi_\ell \in \mathcal{P}^0(\mathcal{E}_\ell)} \langle \phi - \psi_\ell\,, V(\phi-\psi_\ell)\rangle_\Gamma &\le \langle (1-\Pi_\ell)\phi\,, V(1-\Pi_\ell)\phi\rangle_\Gamma \\ &\lesssim \|h_\ell^{1/2}(1-\Pi_\ell)\phi\|_{L^2(\Gamma)} \le \|h_\ell^{1/2}(\phi-\phi_\ell)\|_{L^2(\Gamma)} \end{aligned}$$

with $\Pi_\ell : L^2(\Gamma) \to \mathcal{P}^0(\mathcal{T}_\ell)$ being the L^2-orthogonal projection. Altogether, we see that

$$\begin{aligned} |||(u-u_\ell, \phi-\phi_\ell)||| &\lesssim \|u-u_\ell\|_{H^1(\Omega^{int})} + \|h_\ell^{1/2}(\phi-\phi_\ell)\|_{L^2(\Gamma)} + \mathrm{osc}_\ell \\ &=: \mathrm{err}_\ell(u) + \mathrm{err}_\ell(\phi) + \mathrm{osc}_\ell \end{aligned} \tag{79}$$

provides a computable upper bound for the energy error. In the same spirit, the error estimator μ_ℓ is split into

$$\mu_\ell^2 = \sum_{T\in\mathcal{T}_\ell} \mu_\ell(T)^2 + \sum_{E\in\mathcal{E}_\ell} \mu_\ell(E)^2 =: \mu_\ell(u)^2 + \mu_\ell(\phi)^2. \tag{80}$$

Recall that Theorem 8.8 predicts

$$\mu_\ell(u) + \mu_\ell(\phi) + \mathrm{osc}_\ell \lesssim \|u-u_\ell\|_{H^1(\Omega^{int})} + \|\phi-\phi_\ell\|_{H^{-1/2}(\Gamma)} + \mathrm{osc}_\ell \lesssim \mu_\ell(u) + \mu_\ell(\phi) + \mathrm{osc}_\ell,$$

whence

$$\mu_\ell(u) + \mu_\ell(\phi) + \mathrm{osc}_\ell \lesssim |||(u-u_\ell,\phi-\phi_\ell)||| + \mathrm{osc}_\ell \lesssim \mu_\ell(u) + \mu_\ell(\phi) + \mathrm{osc}_\ell,$$

where the upper bound holds under the saturation assumption (58).
In the following, we plot the five quantities $\mathrm{err}_\ell(u)$, $\mathrm{err}_\ell(\phi)$, $\mu_\ell(u)$, $\mu_\ell(\phi)$, and osc_ℓ from (79)–(80) over the number $N = \#\mathcal{T}_\ell$ of triangles, where both axes are scaled logarithmically. We consider uniform mesh-refinement $\mathcal{T}_\ell = \mathcal{T}_\ell^{(\mathrm{unif})}$ with $\mathcal{T}_\ell^{(\mathrm{unif})} := \widehat{\mathcal{T}}_{\ell-1}$, as well as adaptive mesh-refinement, where the meshes a generated with `adaptiveAlgorithmFEMBEM` from Section 11.5, with the adaptivity parameter $\theta = 0.25$. Note that a decay with slope $-\alpha$ indicates some dependence $\Omega(N^{-\alpha})$. For uniform meshes with mesh-size h, this corresponds to $\Omega(h^{2\alpha})$. We stress that, by theory, an overall slope of $\alpha = 1/2$ is thus optimal with P1-finite elements.
For the adaptive mesh-refinement, recall that all integral operators have to be computed with respect to the fine mesh $\widehat{\mathcal{T}}_\ell$. Therefore, one usually takes the improved approximation $\widehat{u}_{0,\ell} \in \mathcal{S}^1(\widehat{\mathcal{T}}_\ell|_\Gamma)$ instead of $u_{0,\ell} \in \mathcal{S}^1(\mathcal{T}_\ell|_\Gamma)$. Consequently, we then consider $\widehat{\mathrm{osc}}_\ell = \|h_\ell^{1/2}(u_0 - \widehat{u}_{0,\ell})'\|_{L^2(\Gamma)}$ instead of osc_ℓ. Moreover, although u_ℓ is not needed by the adaptive algorithm in Section 11.5, we nevertheless plot err_ℓ to give a fair comparison of uniform and adaptive mesh-refinement.
Besides the experimental convergence rates, we plot $\mathrm{err}_\ell(u)$, $\mathrm{err}_\ell(\phi)$, $\mu_\ell(u)$, $\mu_\ell(\phi)$, and osc_ℓ (resp. $\widehat{\mathrm{osc}}_\ell$) over the computational time t_ℓ.

- For uniform mesh-refinement, $t_\ell = t_\ell^{(\mathrm{unif})}$ is the time needed for ℓ uniform refinements of the initial mesh $\mathcal{T}_0$ to obtain $\mathcal{T}_\ell$, plus the time for building and solving the Galerkin system with respect to $X_\ell \times Y_\ell$.

For adaptive mesh, refinement, the mesh $\mathcal{T}_\ell$ depends on the entire history of preceding meshes (and solutions). Therefore, the computational time has to be defined differently, where $t_{-1}^{(\mathrm{adap})} := 0$.

- For adaptive mesh-refinement, $t_\ell = t_\ell^{(\mathrm{adap})}$ is the sum of the time $t_{\ell-1}^{(\mathrm{adap})}$ elapsed in prior steps of the adaptive algorithm, plus the time for generating the fine mesh $\widehat{\mathcal{T}}_\ell$, building and solving the Galerkin system with respect to $\widehat{X}_\ell \times \widehat{Y}_\ell$, computing the local contributions of the data oscillations $\widehat{\mathrm{osc}}_\ell$ and the error estimator μ_ℓ, element marking, and local refinement of $\mathcal{T}_\ell$ to generate $\mathcal{T}_{\ell+1}$.

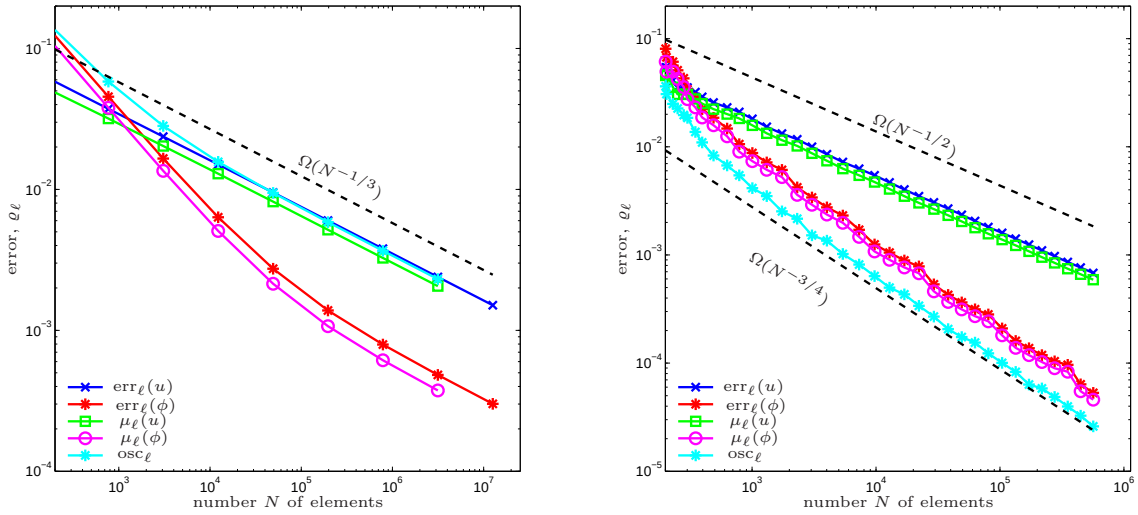


Figure 8: Estimators $\mathrm{err}_\ell(u)$, $\mathrm{err}_\ell(\phi)$, $\mu_\ell(u)$ and $\mu_\ell(\phi)$ from (79)–(80) as well as data oscillations osc_ℓ in linear Experiment 12.1, plotted over the number $N = \#\mathcal{T}_\ell$ of triangles for uniform (left) and adaptive mesh-refinement (right).

Although this definition seems to favour uniform mesh-refinement, we think that it provides a fair comparison between uniform and adaptive mesh-refinement.
All experiments are conducted by use of MATLAB (Release 2009b) running on a common 64 Bit Linux system with 32 GB of RAM. Throughout, the occuring linear systems are solved by use of the MATLAB backslash operator. For the computation of the boundary integral operators, we use the MATLAB BEM library `HILBERT`, cf. [1]; see `http://www.asc.tuwien.ac.at/abem/hilbert/`

12.1 Linear Problem on L-Shaped Domain

We consider the L-shaped domain visualized in Figure 7. With $A : L^2(\Omega)^2 \to L^2(\Omega)^2$ being the identity, we prescribe the exact solution of (59) as

$$\begin{aligned} u^{\mathrm{int}}(x,y) &= r^{2/3}\sin(\tfrac{2}{3}\varphi) && \text{in } \Omega^{\mathrm{int}}, \\ u^{\mathrm{ext}}(x,y) &= \log\big((x+1/8)^2+(y+1/8)^2\big)^{1/2} && \text{in } \Omega^{\mathrm{ext}}, \end{aligned} \tag{81}$$

where (r,φ) are the polar coordinates of $(x,y) \in \mathbb{R}^2$ with respect to $(0,0)$. Clearly, the identity satisfies the assumptions of our model problem, and the FEM-BEM coupling (61) is linear. Recall that (u,ϕ) denotes the exact solution of (61) and note that $u = u^{\mathrm{int}}$ has a generic singularity at the reentrant corner, whereas $\phi = \nabla u^{\mathrm{ext}} \cdot n$ is piecewise smooth.
In Figure 8, we plot the convergence of the error quantities from (79)–(80). Since the interior solution has a generic singularity at the reentrant corner, uniform mesh-refinement leads to a suboptimal order of convergence $\alpha = 1/3$, i.e. we observe $\Omega(h^{2/3})$. For $\mathrm{err}_\ell(u)$

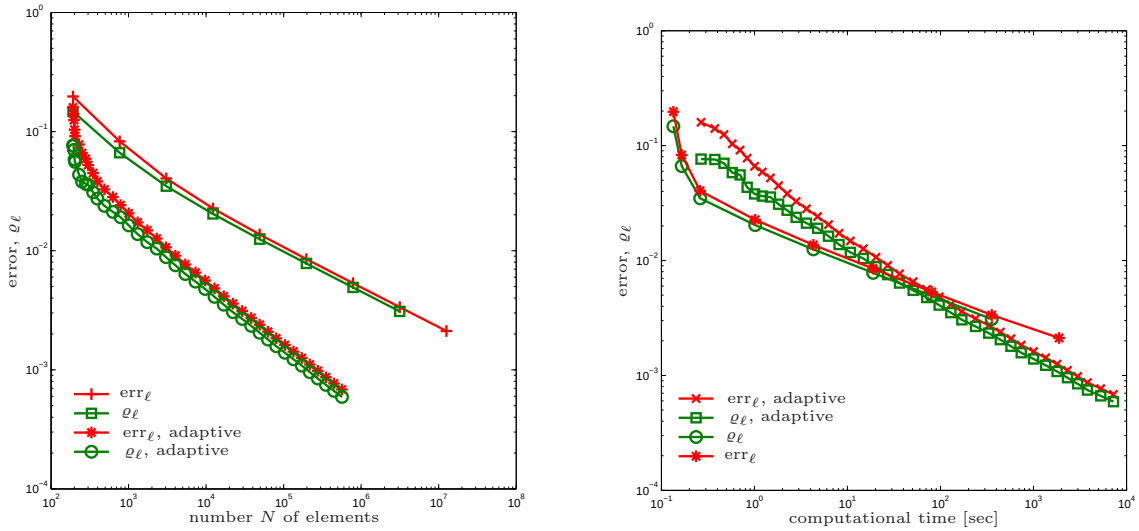


Figure 9: Comparison of uniform and adaptive mesh-refining in linear Experiment 12.1, where the error bound err_ℓ and the estimator ϱ_ℓ from (82) are plotted over the number $N = \#\mathcal{T}_\ell$ of triangles (left) and over the computational time (right).

and $\mu_\ell(u)$, this asymptotics is observed already on coarse meshes. For $\mathrm{err}_\ell(\phi)$ and $\mu_\ell(\phi)$, a preasymptotic phase occurs. For adaptive mesh-refinement, we observe the optimal order of convergence $\alpha = 1/2$ for $\mathrm{err}_\ell(u)$ and $\mu_\ell(u)$. Moreover, the terms $\mathrm{err}_\ell(\phi)$ and $\mu_\ell(\phi)$ even converge with order $\alpha = 3/2$ which is optimal for the approximation of a smooth function by piecewise constants with respect to the $H^{-1/2}(\Gamma)$-norm.
Figure 9 provides comparisons between uniform and adaptive mesh-refinement. We plot

$$\mathrm{err}_\ell := \big(\mathrm{err}_\ell(u)^2 + \mathrm{err}_\ell(\phi)^2 + \mathrm{osc}_\ell^2\big)^{1/2} \quad \text{and} \quad \varrho_\ell = \big(\mu_\ell(u)^2 + \mu_\ell(\phi)^2 + \mathrm{osc}_\ell^2\big)^{1/2} \tag{82}$$

over the number $N = \#\mathcal{T}_\ell$ of elements as well as over the computational time.

12.2 Nonlinear Problem on L-Shaped Domain

We consider the L-shaped domain visualized in Figure 7. We define

$$\rho(t) = 2 + \frac{1}{1+t} \quad \text{for } t > 0 \tag{83}$$

and note that the derivative satisfies $-1 \le \rho'(t) < 0$. The nonlinear operator A is then defined by

$$A(x) = \rho(|x|)\, x \quad \text{for } x \in \mathbb{R}^2. \tag{84}$$

We stress that A is strongly monotone and Lipschitz continuous on $L^2(\Omega)^2$. We prescribe the same solution (81) as for the linear experiment from the previous section. The volume

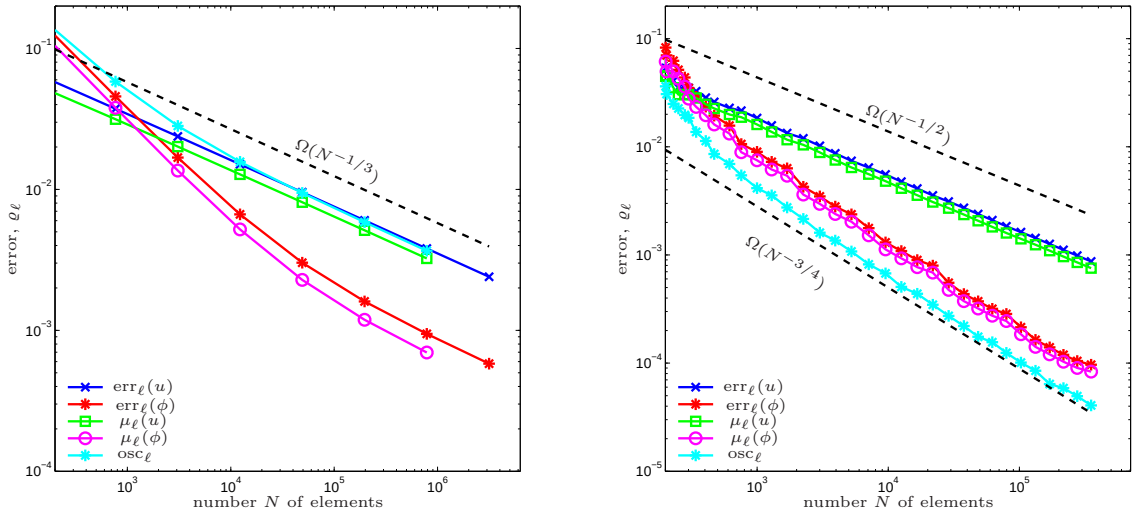


Figure 10: Estimators $\mathrm{err}_\ell(u)$, $\mathrm{err}_\ell(\phi)$, $\mu_\ell(u)$ and $\mu_\ell(\phi)$ from (79)–(80) as well as data oscillations osc_ℓ in nonlinear Experiment 12.2 on the L-shaped domain, plotted over the number $N = \#\mathcal{T}_\ell$ of triangles for uniform (left) and adaptive mesh-refinement (right).

force f then reads

$$f(x,y) := -\frac{4}{27} r^{-5/3} \frac{\sin(\frac{2}{3}\varphi)}{(1+\frac{2}{3}r^{-1/3})^2},$$

where (r,φ) are the polar coordinates of $(x,y) \in \mathbb{R}^2$ with respect to $(0,0)$.
Figure 10 and 11 provide the experimental convergence results. The observations are the same as for the linear experiment in Section 12.1. Figure 12 shows some adaptively generated meshes. We observe a strong mesh-refinement towards the reentrant corner, where u^{int} is singular.

12.3 Nonlinear Problem on Z-Shaped Domain

In the final example, Ω^{int} is the Z-shaped domain, shown in Figure 7. The exact solution reads

$$\begin{aligned} u^{\mathrm{int}}(x,y) &= r^{4/7}\sin(\tfrac{4}{7}\varphi) && \text{in } \Omega^{\mathrm{int}}, \\ u^{\mathrm{ext}}(x,y) &= \log\big((x+1/8)^2+(y+1/8)^2\big)^{1/2} && \text{in } \Omega^{\mathrm{ext}}, \end{aligned} \tag{85}$$

with (r,φ) being the polar coordinates of $(x,y) \in \mathbb{R}^2$. Again, u^{int} has a generic singularity at the reentrant corner. With the nonlinear operator A from Section 12.2, the right-hand side f becomes

$$-\mathrm{div}(\rho(|\nabla u^{\mathrm{int}}|)\nabla u^{\mathrm{int}}) = f,$$

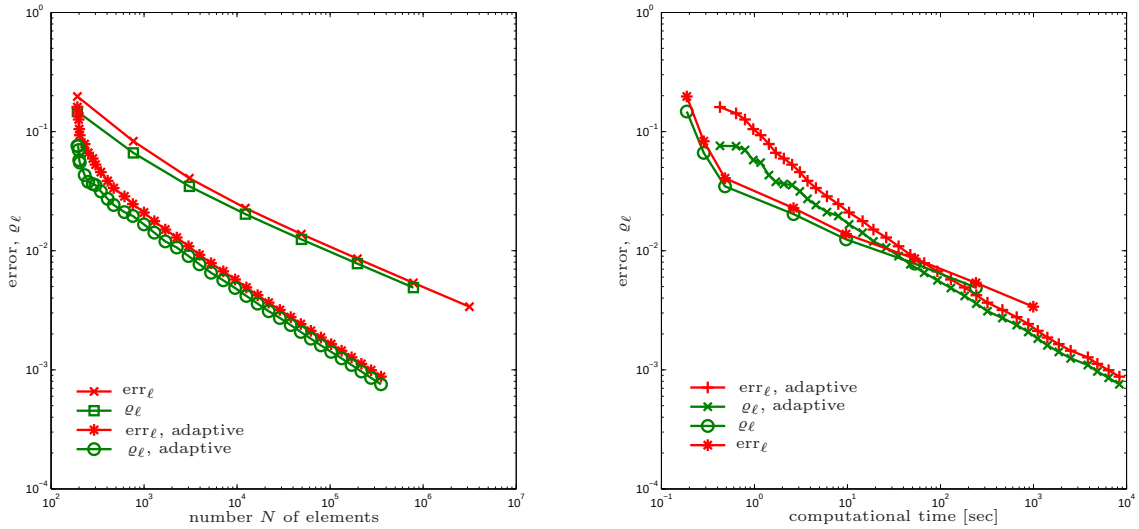


Figure 11: Comparison of uniform and adaptive mesh-refining in nonlinear Experiment 12.2 on the L-shaped domain, where the error bound err_ℓ and the estimator ϱ_ℓ from (82) are plotted over the number $N = \#\mathcal{T}_\ell$ of triangles (left) and over the computational time (right).

where

$$f(r,\varphi) := -\frac{48}{343} r^{-13/7} \frac{\sin(\frac{4}{7}\varphi)}{(1+\frac{4}{7}r^{-3/7})^2}.$$

Figure 13 and 14 provide the experimental convergence results. As before, the observations are the same as for the linear experiment in Section 12.1. Figure 15 shows some adaptively generated meshes which show a strong mesh-refinement towards the reentrant corner.

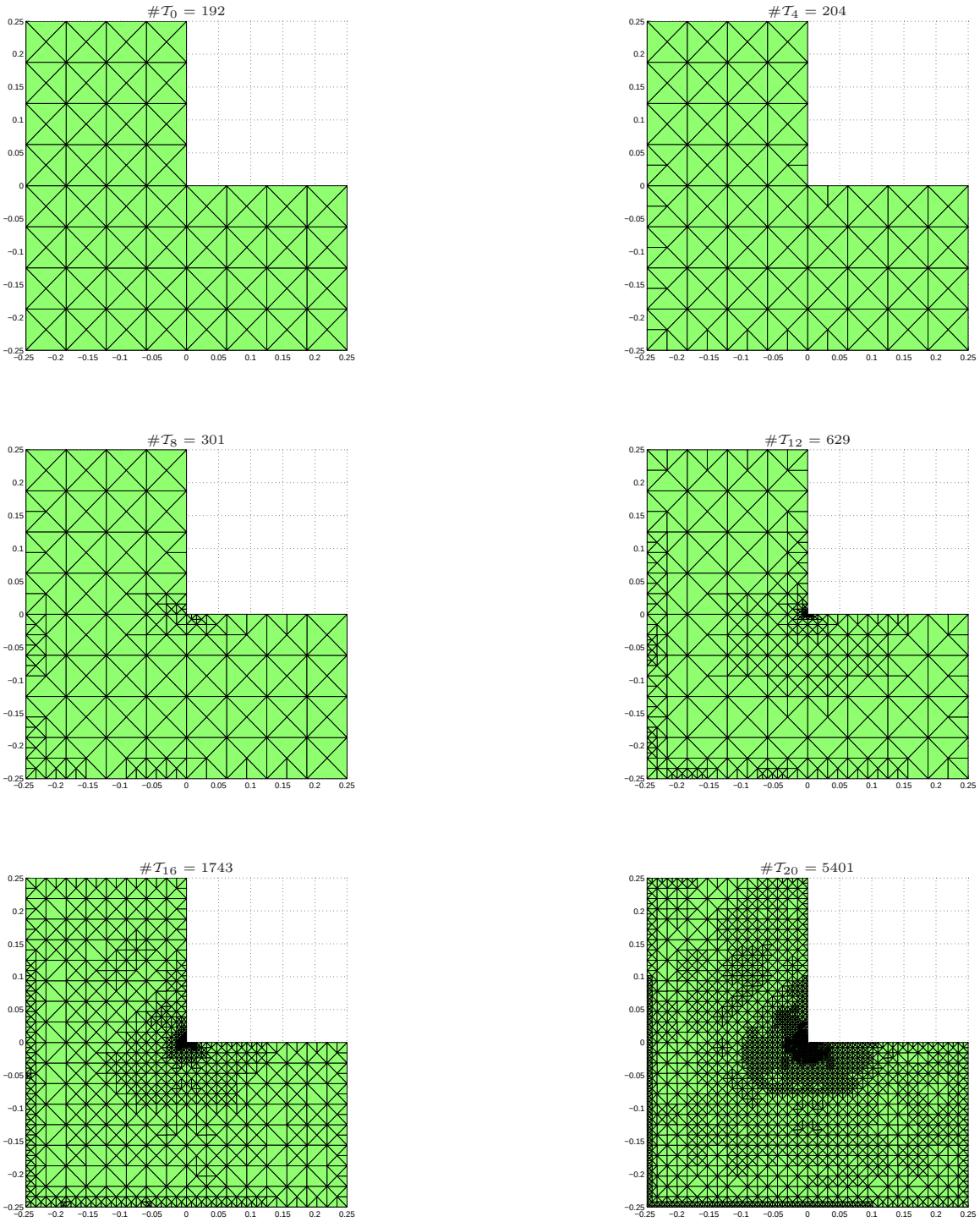


Figure 12: Adaptively generated meshes $\mathcal{T}_\ell$ in nonlinear Experiment 12.2 on the L-shaped domain.

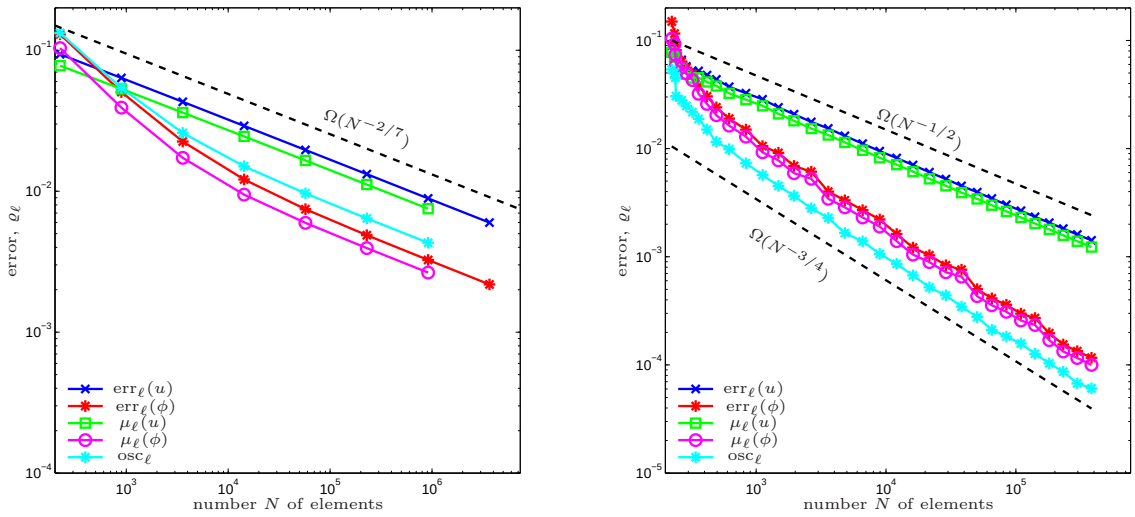


Figure 13: Estimators $\mathrm{err}_\ell(u)$, $\mathrm{err}_\ell(\phi)$, $\mu_\ell(u)$ and $\mu_\ell(\phi)$ from (79)–(80) as well as data oscillations osc_ℓ in nonlinear Experiment 12.3 on the Z-shaped domain, plotted over the number $N = \#\mathcal{T}_\ell$ of triangles for uniform (left) and adaptive mesh-refinement (right).

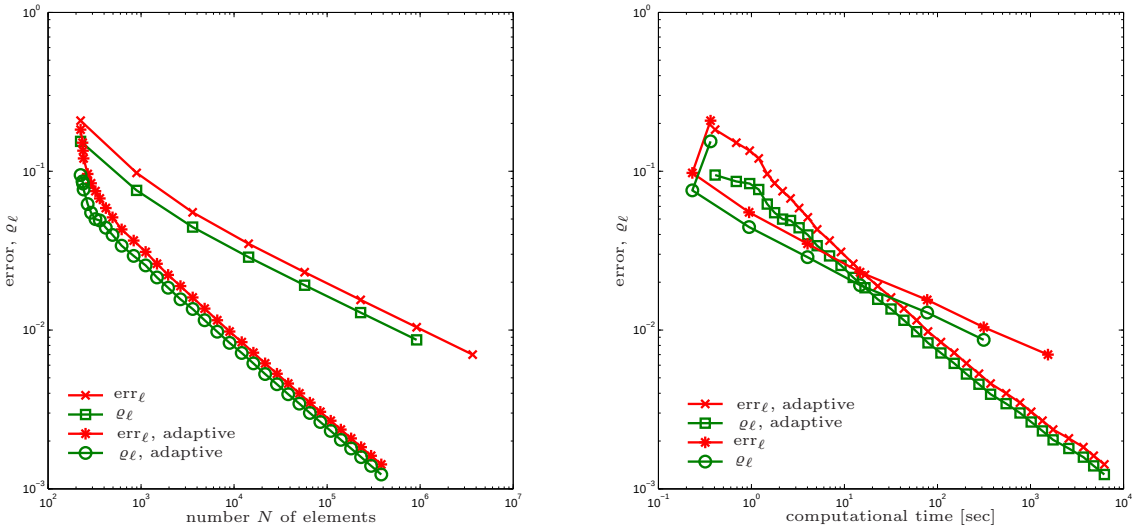


Figure 14: Comparison of uniform and adaptive mesh-refining in nonlinear Experiment 12.3 on the Z-shaped domain, where the error bound err_ℓ and the estimator ϱ_ℓ from (82) are plotted over the number $N = \#\mathcal{T}_\ell$ of triangles (left) and over the computational time (right).

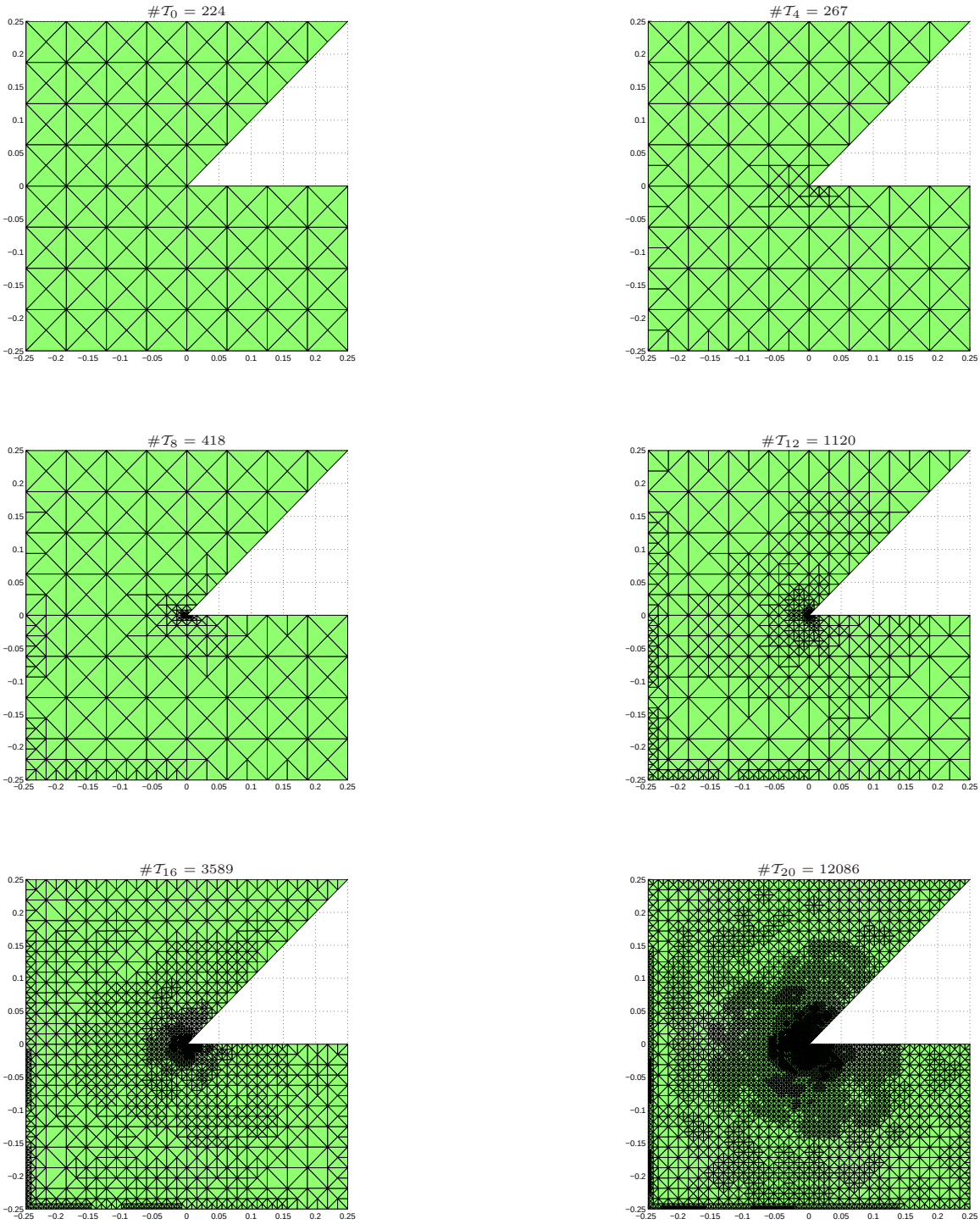


Figure 15: Adaptively generated meshes $\mathcal{T}_\ell$ in nonlinear Experiment 12.3 on the Z-shaped domain.

References

[1] M. Aurada, M. Ebner, S. Ferraz-Leite, M. Mayr, P. Goldenits, M. Karkulik, D. Praetorius: *HILBERT — A* Matlab *implementation of adaptive BEM*, ASC Report **44/2009**, Institute for Analysis and Scientific Computing, Vienna University of Technology, Wien, 2009, software download at `http://www.asc.tuwien.ac.at/abem/hilbert/`

[2] M. Aurada, M. Feischl, D. Praetorius: Convergence of some adaptive FEM-BEM coupling, ASC Report 06/2010, Institute for Analysis and Scientific Computing, Vienna University of Technology, Wien, 2010, ISBN 978-3-902627-03-2.

[3] J. Bielak, R.C. MacCamy: An exterior interface problem in two-dimensional elastodynamics, *Quart. Appl. Math.* **41**, 143-159 (1983).

[4] C. Carstensen: A posteriori error estimate for the symmetric coupling of finite elements and boundary elements. Computing 57 (1996), **no. 4**, 301–322. (Reviewer: J. T. Oden)

[5] 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.

[6] C. Carstensen, E. Stephan: Adaptive coupling of boundary elements and finite elements, Math. Model. Numer. Anal. **29** (1995), 779â817.

[7] P. Clement: Approximation by finite element functions using local regularization, Rairo, Sér. Rouge Anal. Numér., **R-2**, pp. 77-84 (1975).

[8] M. Costabel: A symmetric method for the coupling of finite elements and boundary elements, *The Mathematics of Finite Elements and Applications IV, MAFELAP 1987* (Edited by J.R. Whiteman),Academic Press, London, pp. 281-288 (1988).

[9] J. Johnson, J.C. Nedelec: On the coupling of boundary integral and finite element methods, *Math. Comp.* **35**, 1063-1079 (1980).

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

[11] D. Praetorius: Finite Element Method, Lecture Notes.

[12] D. Praetorius: Introduction to Boundary Element Method, lecture held at Humboldt-University of Berlin from July 16-20, 2007.

[13] W. Rudin: Functional analysis. Second edition. International Series in Pure and Applied Mathematics. McGraw-Hill, Inc., New York, 1991. xviii+424 pp.

[14] F.-J. Sayas: The validity of Johnson-Nédélec's BEM-FEM coupling on polygonal interfaces. SIAM J. Numer. Anal. 47 (2009), **no. 5**, 3451–3463.

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

[16] D. Werner: Funktionalanalysis. Third, revised and extended edition. Springer-Verlag, Berlin, 2000. xii+501 pp. ISBN: 3-540-67645-7