



TECHNISCHE
UNIVERSITÄT
WIEN

Vienna University of Technology

BACHELORARBEIT

A-posteriori Fehlerschätzer für die Simm'sche Integralgleichung

Ausgeführt am Institut für
Analysis und Scientific Computing
der Technischen Universität Wien

unter der Anleitung von
Ao. Univ.Prof. Dipl.-Math. Dr.techn. Dirk Praetorius
und
Dipl.-Ing. Dr.techn. Markus Aurada

durch
Michael Ebner
Steinackergasse 18/20
1120 Wien

Wien, Februar 2012

Inhaltsverzeichnis

Abbildungsverzeichnis	v
1 Einleitung	1
2 Vorbereitungen	3
2.1 Diskretisierung des Randes	3
2.2 Funktionen am Rand	3
2.3 Randintegrale	4
2.4 Bogenlängenableitung	5
3 Symm'sche Integralgleichung	6
3.1 Variationsformulierung und eindeutige Lösbarkeit	6
3.2 Galerkin-Diskretisierung	9
3.3 Dirichlet-Datenoszillationen	10
3.4 Konvergenzrate und zuverlässige Schranke des Fehlers $\ \phi - \Phi_\ell\ _V$	17
4 Adaptive BEM	22
4.1 Dörfler Markierung für lokale Netzverfeinerung	22
4.2 Adaptiver Netzverfeinerungs-Algorithmus	24
5 Fehlerschätzer	26
5.1 Notation	26
5.2 Kanonischer $(h - h/2)$ -Fehlerschätzer η_ℓ bzw. η_ℓ^*	27
5.3 Weitere $(h - h/2)$ -Fehlerschätzer $\tilde{\eta}_\ell, \mu_\ell, \tilde{\mu}_\ell$	30
5.4 Implementierung der globalen $(h - h/2)$ -Fehlerschätzer $\eta_\ell, \tilde{\eta}_\ell$ in MATLAB	32
5.4.1 Fehlerschätzer η_ℓ	32
5.4.2 Fehlerschätzer $\tilde{\eta}_\ell$	33
5.5 Implementierung der lokalen $(h - h/2)$ -Fehlerschätzer $\mu_\ell, \tilde{\mu}_\ell$ in MATLAB	34
5.5.1 Fehlerschätzer μ_ℓ	34
5.5.2 Fehlerschätzer $\tilde{\mu}_\ell$	36
5.6 Two-Level-Fehlerschätzer τ_ℓ	37
5.7 Implementierung des lokalen Fehlerschätzers τ_ℓ in MATLAB	38
5.8 Gewichteter Residualschätzer ϱ_ℓ	40
5.9 Implementierung des gewichteten Residualschätzers ϱ_ℓ in MATLAB	44
5.10 Faermann-Residualschätzer φ_ℓ	47
5.11 Implementierung des Faermann-Residualschätzers φ_ℓ in MATLAB	52
5.12 Steinbach-Schätzer σ_ℓ	56
5.12.1 Steinbach-Schätzer über Einfachschichtpotential V	65
5.12.2 Steinbach-Schätzer über Multilevel-Operator	66
5.13 Implementierung der Steinbach-Schätzer $\sigma_{\ell,V}$ und $\sigma_{\ell,A}$ in MATLAB	70
5.13.1 Steinbach-Schätzer $\sigma_{\ell,V}$	70
5.13.2 Steinbach-Schätzer $\sigma_{\ell,A}$	75

6	Numerische Resultate	81
6.1	Vorüberlegungen für die Berechnung auf den einzelnen Gebieten	85
6.1.1	L-Shape	85
6.1.2	Z-Shape	86
6.1.3	Schlitz	87
6.2	Parameterwahl der Steinbach-Schätzer	89
6.2.1	L-Shape: Parameterschätzung für Steinbach-Schätzer $\sigma_{\ell,V}$	89
6.2.2	L-Shape: Parameterschätzung für Steinbach-Schätzer $\sigma_{\ell,A}$	90
6.2.3	Z-Shape: Parameterüberprüfung der Steinbach-Schätzer $\sigma_{\ell,V}, \sigma_{\ell,A}$	97
6.2.4	Die Steinbach-Fehlerindikatoren und der Steinbach-Schätzer $\sigma_{\ell,V}$	100
6.3	Analyse der lokalen a-posteriori Fehlerschätzer	102
6.3.1	L-Shape	106
6.3.2	Z-Shape	123
6.3.3	Schlitz-Problem	134
A	Anhang	147
A.1	Programmcode Faermann-Residualschätzer für das Schlitz-Problem	147
A.2	Programmcode des $\sigma_{\ell,V}$ -adaptiv gesteuerten BEM-Algorithmus	148
	Literatur	152

Abbildungsverzeichnis

1	Beispiel-Gebiet mit 8 Elementen	4
2	Faermann-Residualschätzer: Überlappender Bereich $\omega_j = E_j \cup E_k$	47
3	Faermann-Residualschätzer: Überlappender Bereich $\omega_j \cup \omega_{\pi(j)}$	49
4	Steinbach-Schätzer $\sigma_{\ell,A}$: L^2 -Projektion auf verfeinerte Netze	79
5	L-Shape mit 8 Elementen	83
6	Z-Shape mit 9 Elementen	83
7	Schlitz-Problem mit 4 Elementen	84
8	Dirichlet- und Neumann-Daten auf L-Shape	86
9	Dirichlet- und Neumann-Daten auf Z-Shape	87
10	Parameterschätzung $\sigma_{\ell,V}$ auf L-Shape (3-Punkt Gaußquadratur)	91
11	Parameterschätzung $\sigma_{\ell,V}$ auf L-Shape (5-Punkt Gaußquadratur)	92
12	Parameterschätzung $\sigma_{\ell,V}$ auf L-Shape (Vergleich 3- und 5-Punkt Gaußquadratur)	93
13	Parameterschätzung $\sigma_{\ell,A}$ auf L-Shape (3-Punkt Gaußquadratur)	94
14	Parameterschätzung $\sigma_{\ell,A}$ auf L-Shape (1-Punkt Gaußquadratur)	95
15	Parameterschätzung $\sigma_{\ell,A}$ auf L-Shape (Vergleich 1- und 3-Punkt Gaußquadratur)	96
16	Parameterüberprüfung $\sigma_{\ell,V}$ auf Z-Shape (3- und 5-Punkt Gaußquadratur)	98
17	Parameterüberprüfung $\sigma_{\ell,A}$ auf Z-Shape (1- und 3-Punkt Gaußquadratur)	99
18	Vergleich des Steinbach-Schätzers $\sigma_{\ell,V}$ mit $\check{\sigma}_{\ell,V}$	101
19	L-Shape: Fehlerschätzer und Schranke error_ℓ bei unif. Netzverfeinerung	109
20	L-Shape: Fehlerschätzer und Schranke error_ℓ bei μ_ℓ -adaptiver Steuerung	109
21	L-Shape: Fehlerschätzer und Schranke error_ℓ bei $\tilde{\mu}_\ell$ -adaptiver Steuerung	110
22	L-Shape: Fehlerschätzer und Schranke error_ℓ bei τ_ℓ -adaptiver Steuerung	110
23	L-Shape: Fehlerschätzer und Schranke error_ℓ bei ϱ_ℓ -adaptiver Steuerung	111
24	L-Shape: Fehlerschätzer und Schranke error_ℓ bei φ_ℓ -adaptiver Steuerung	111
25	L-Shape: Fehlerschätzer und Schranke error_ℓ bei $\sigma_{\ell,A}$ -adaptiver Steuerung	112
26	L-Shape: Fehlerschätzer und Schranke error_ℓ bei $\sigma_{\ell,V}$ -adaptiver Steuerung	112
27	L-Shape: Fehlerschätzer μ_ℓ bei allen adaptiven Steuerungen	113
28	L-Shape: Fehlerschätzer $\tilde{\mu}_\ell$ bei allen adaptiven Steuerungen	113
29	L-Shape: Fehlerschätzer τ_ℓ bei allen adaptiven Steuerungen	114
30	L-Shape: Fehlerschätzer ϱ_ℓ bei allen adaptiven Steuerungen	114
31	L-Shape: Fehlerschätzer φ_ℓ bei allen adaptiven Steuerungen	115
32	L-Shape: Fehlerschätzer $\sigma_{\ell,A}$ bei allen adaptiven Steuerungen	115
33	L-Shape: Fehlerschätzer $\sigma_{\ell,V}$ bei allen adaptiven Steuerungen	116
34	L-Shape: Alle Fehlerschätzer bei eigener adaptiver Steuerung	117
35	L-Shape: Schranke error_ℓ bei allen adaptiven Steuerungen	118
36	L-Shape: Berechnungsdauer aller Verfeinerungsstrategien	119
37	L-Shape: Zeitmessung bei μ_ℓ -adaptiver Netzverfeinerung	119
38	L-Shape: Zeitmessung bei $\tilde{\mu}_\ell$ -adaptiver Netzverfeinerung	120
39	L-Shape: Zeitmessung bei τ_ℓ -adaptiver Netzverfeinerung	120
40	L-Shape: Zeitmessung bei ϱ_ℓ -adaptiver Netzverfeinerung	121
41	L-Shape: Zeitmessung bei φ_ℓ -adaptiver Netzverfeinerung	121
42	L-Shape: Zeitmessung bei $\sigma_{\ell,A}$ -adaptiver Netzverfeinerung	122

43	L-Shape: Zeitmessung bei $\sigma_{\ell,V}$ -adaptiver Netzverfeinerung	122
44	Z-Shape: Fehlerschätzer und Schranke error_{ℓ} bei unif. Netzverfeinerung	124
45	Z-Shape: Fehlerschätzer und Schranke error_{ℓ} bei μ_{ℓ} -adaptiver Steuerung	124
46	Z-Shape: Fehlerschätzer und Schranke error_{ℓ} bei $\tilde{\mu}_{\ell}$ -adaptiver Steuerung	125
47	Z-Shape: Fehlerschätzer und Schranke error_{ℓ} bei τ_{ℓ} -adaptiver Steuerung	125
48	Z-Shape: Fehlerschätzer und Schranke error_{ℓ} bei ϱ_{ℓ} -adaptiver Steuerung	126
49	Z-Shape: Fehlerschätzer und Schranke error_{ℓ} bei φ_{ℓ} -adaptiver Steuerung	126
50	Z-Shape: Fehlerschätzer und Schranke error_{ℓ} bei $\sigma_{\ell,A}$ -adaptiver Steuerung	127
51	Z-Shape: Fehlerschätzer und Schranke error_{ℓ} bei $\sigma_{\ell,V}$ -adaptiver Steuerung	127
52	Z-Shape: Fehlerschätzer μ_{ℓ} bei allen adaptiven Steuerungen	128
53	Z-Shape: Fehlerschätzer $\tilde{\mu}_{\ell}$ bei allen adaptiven Steuerungen	128
54	Z-Shape: Fehlerschätzer τ_{ℓ} bei allen adaptiven Steuerungen	129
55	Z-Shape: Fehlerschätzer ϱ_{ℓ} bei allen adaptiven Steuerungen	129
56	Z-Shape: Fehlerschätzer φ_{ℓ} bei allen adaptiven Steuerungen	130
57	Z-Shape: Fehlerschätzer $\sigma_{\ell,A}$ bei allen adaptiven Steuerungen	130
58	Z-Shape: Fehlerschätzer $\sigma_{\ell,V}$ bei allen adaptiven Steuerungen	131
59	Z-Shape: Alle Fehlerschätzer bei eigener adaptiver Steuerung	132
60	Z-Shape: Schranke error_{ℓ} bei allen adaptiven Steuerungen	133
61	Schlitz-Problem: Fehlerschätzer und Schranke error_{ℓ} bei unif. Netzverfeinerung	136
62	Schlitz-Problem: Fehlerschätzer und Schranke error_{ℓ} bei μ_{ℓ} -adaptiver Steuerung	136
63	Schlitz-Problem: Fehlerschätzer und Schranke error_{ℓ} bei $\tilde{\mu}_{\ell}$ -adaptiver Steuerung	137
64	Schlitz-Problem: Fehlerschätzer und Schranke error_{ℓ} bei τ_{ℓ} -adaptiver Steuerung	137
65	Schlitz-Problem: Fehlerschätzer und Schranke error_{ℓ} bei ϱ_{ℓ} -adaptiver Steuerung	138
66	Schlitz-Problem: Fehlerschätzer und Schranke error_{ℓ} bei φ_{ℓ} -adaptiver Steuerung	138
67	Schlitz-Problem: Fehlerschätzer μ_{ℓ} bei allen adaptiven Steuerungen	139
68	Schlitz-Problem: Fehlerschätzer $\tilde{\mu}_{\ell}$ bei allen adaptiven Steuerungen	139
69	Schlitz-Problem: Fehlerschätzer τ_{ℓ} bei allen adaptiven Steuerungen	140
70	Schlitz-Problem: Fehlerschätzer ϱ_{ℓ} bei allen adaptiven Steuerungen	140
71	Schlitz-Problem: Fehlerschätzer φ_{ℓ} bei allen adaptiven Steuerungen	141
72	Schlitz-Problem: Alle Fehlerschätzer bei eigener adaptiver Steuerung	142
73	Schlitz-Problem: Schranke error_{ℓ} bei allen adaptiven Steuerungen	143
74	Schlitz-Problem: Berechnungsdauer aller Verfeinerungsstrategien	144
75	Schlitz-Problem: Zeitmessung bei μ_{ℓ} -adaptiver Netzverfeinerung	144
76	Schlitz-Problem: Zeitmessung bei $\tilde{\mu}_{\ell}$ -adaptiver Netzverfeinerung	145
77	Schlitz-Problem: Zeitmessung bei τ_{ℓ} -adaptiver Netzverfeinerung	145
78	Schlitz-Problem: Zeitmessung bei ϱ_{ℓ} -adaptiver Netzverfeinerung	146
79	Schlitz-Problem: Zeitmessung bei φ_{ℓ} -adaptiver Netzverfeinerung	146

1 Einleitung

Die Randelementmethode, kurz **BEM** für **B**oundary **E**lement **M**ethod, ist ein Diskretisierungsverfahren für elliptische Differentialgleichungen der Form

$$Lu(\tilde{x}) = f(\tilde{x}) \quad \text{für } \tilde{x} \in \Omega$$

mit einem elliptischen und selbstadjungierten Differentialoperator L und einem beschränkten, einfach zusammenhängenden Gebiet Ω mit Lipschitz-Rand $\Gamma = \partial\Omega$.

Als Modellproblem dient uns die Poisson-Gleichung

$$-\Delta u(\tilde{x}) = f(\tilde{x}) \quad \text{für } \tilde{x} \in \Omega \quad (1.1)$$

auf einem beschränkten Lipschitz-Gebiet $\Omega \subseteq \mathbb{R}^2$. Die Fundamentallösung des Laplace-Operators im Pol y ist durch

$$U(\tilde{x}, y) := -\frac{1}{2\pi} \log |\tilde{x} - y|$$

gegeben. Mit Hilfe der **dritten Greenschen Formel** kann nun (1.1) in ein äquivalentes Problem

$$u(\tilde{x}) = \int_{\Omega} U(\tilde{x}, y) f(y) dy + \int_{\Gamma} U(\tilde{x}, y) \phi(y) d\Gamma(y) - \int_{\Gamma} \frac{\partial}{\partial n_y} U(\tilde{x}, y) g(y) d\Gamma(y) \quad \text{für } \tilde{x} \in \Omega \quad (1.2)$$

übergeführt werden, siehe [18, Kapitel 5]. Dabei stellt n_y den äußeren Einheitsnormalenvektor von Ω im Punkt y , $\phi := \frac{\partial}{\partial n} u$ die Normalenableitung und $g := u|_{\Gamma}$ die Spur von u auf Γ dar.

Mit den linearen Integraloperatoren

$$\begin{aligned} \tilde{N}f(\tilde{x}) &:= \int_{\Omega} U(\tilde{x}, y) f(y) dy = -\frac{1}{2\pi} \int_{\Omega} \log |\tilde{x} - y| f(y) dy, \\ \tilde{V}\phi(\tilde{x}) &:= \int_{\Gamma} U(\tilde{x}, y) \phi(y) d\Gamma(y) = -\frac{1}{2\pi} \int_{\Gamma} \log |\tilde{x} - y| \phi(y) d\Gamma(y), \end{aligned} \quad (1.3)$$

$$\tilde{K}g(\tilde{x}) := \int_{\Gamma} \frac{\partial}{\partial n_y} U(\tilde{x}, y) g(y) d\Gamma(y) = -\frac{1}{2\pi} \int_{\Gamma} \frac{(y - \tilde{x}) \cdot n_y}{|y - \tilde{x}|^2} g(y) d\Gamma(y) \quad (1.4)$$

können wir die Darstellungsformel (1.2) mittels

$$u(\tilde{x}) = \tilde{N}f(\tilde{x}) + \tilde{V}\phi(\tilde{x}) - \tilde{K}g(\tilde{x}) \quad \text{für alle } \tilde{x} \in \Omega$$

kompakter anschreiben. Anders ausgedrückt bedeutet dies, dass die Lösung u von (1.1) bekannt ist, sobald die **Cauchy-Daten** (g, ϕ) am Rand Γ bekannt sind. Durch Bilden des Grenzüberganges (vgl. [18, Abschnitt 6.6]) $\tilde{x} \rightarrow x \in \Gamma$ und Anwendung der Normalenableitung auf die Darstellungsformel (1.2) erhalten wir ein System von 2 Randintegralgleichungen

$$\begin{pmatrix} g \\ \phi \end{pmatrix} (x) = \begin{pmatrix} u|_{\Gamma} \\ \frac{\partial}{\partial n} u \end{pmatrix} (x) = \begin{pmatrix} 1/2 - K & V \\ W & K' + 1/2 \end{pmatrix} \begin{pmatrix} g \\ \phi \end{pmatrix} (x) + \begin{pmatrix} Nf \\ N_1 f \end{pmatrix} (x) \quad \text{für } x \in \Gamma. \quad (1.5)$$

Die beiden linearen Gleichungen in (1.5) werden als das **Calderón System** bezeichnet. Dazu sind sechs lineare Operatoren, welche nur auf dem Rand Γ agieren, nötig: das **Einfachschichtpotential** V , das **Doppelschichtpotential** K mit **adjungiertem Doppelschichtpotentialoperator** K' , dem **hypersingulären Integraloperator** W und die **Spur** N und die **Normalenableitung** N_1 des **Newton-Potentials** $\tilde{N}$.

Ziel ist es also, die fehlenden Cauchy-Daten am Rand Γ näherungsweise durch die BEM als numerisches Diskretisierungsverfahren zu bestimmen. Danach kann mit Hilfe der Darstellungsformel (1.2) eine Approximation der Lösung von u einfach erhalten werden.

Wir werden uns in dieser Arbeit ausschließlich mit dem Dirichlet-Randwertproblem

$$\begin{aligned} -\Delta u &= 0 & \text{in } \Omega \\ u &= g & \text{auf } \Gamma \end{aligned} \tag{1.6}$$

befassen, d.h. wir suchen die (approximierte) Lösung der Normalenableitung $\phi = \frac{\partial}{\partial n}u$ auf Γ .

Eine dafür sehr effiziente Berechnungsmethode stellt die **Adaptive BEM**, vorgestellt in Kapitel 4, dar. Dabei ist die Kenntnis des Fehlerverhaltens zwischen der approximierten Lösung und der exakten (unbekannten) Lösung unerlässlich. Durch sogenannte **Fehlerschätzer** kann dies hinreichend genau geschätzt werden. In Kapitel 5 werden wir uns mit verschiedenen a-posteriori Fehlerschätzern beschäftigen. Diese sind

- $(h - h/2)$ -basierte Fehlerschätzer inkl. dem Two-Level-Fehlerschätzer,
- der gewichtete Residualschätzer,
- der Faermann-Residualschätzer und
- der Steinbach-Schätzer.

Abschließend werden wir in Kapitel 6 sämtliche Fehlerschätzer auf drei Beispielgebieten eingehend hinsichtlich Stabilität, Konvergenzrate, Genauigkeit und Berechnungsdauer mit Hilfe von numerischen Simulationen untersuchen.

2 Vorbereitungen

Um uns mit dem Modellproblem (1.6) beschäftigen zu können, brauchen wir noch einige Notationen und Vorüberlegungen. Wir gehen wie in [1, Kapitel 1 f.] vor.

2.1 Diskretisierung des Randes

Wir setzen den Rand Γ durchwegs als stückweisen affinen Rand eines polygonalen Lipschitz-Gebietes $\Omega \subseteq \mathbb{R}^2$ voraus.

Das **Netz** oder die **Triangulierung** $\mathcal{E}_\ell$ stellt dabei eine endliche Menge von Elementen $E_j \in \mathcal{E}_\ell$ mit $\Gamma = \bigcup_{j=1}^N E_j$ dar, d.h. $\mathcal{E}_\ell = \{E_1, \dots, E_N\}$, wobei

$$E_j = [a_j, b_j] := \text{conv}\{a_j, b_j\}$$

mit $a_j, b_j \in \mathbb{R}^2$ und $a_j \neq b_j$ gilt.

Weiters bezeichnet $\mathcal{K}_\ell = \{z_1, \dots, z_N\}$ die Menge der Knoten der Triangulierung $\mathcal{E}_\ell$. Für einen geschlossenen Rand Γ gilt $\#\mathcal{E}_\ell = \#\mathcal{K}_\ell$.

Datenstrukturen

Für die Implementierung in MATLAB müssen wir geeignete Datenstrukturen definieren. Die Menge der Knoten $\mathcal{K}_\ell = \{z_1, \dots, z_N\}$ der Triangulierung $\mathcal{E}_\ell = \{E_1, \dots, E_N\}$ wird dargestellt als ein $N \times 2$ -Array **coordinates**. Die j -te Zeile von **coordinates** enthält die Koordinaten des j -ten Knotens $z_j = (x_j, y_j) \in \mathbb{R}^2$,

$$\text{coordinates}(j, :) = [x_j \ y_j].$$

Das i -te Randelement $E_i = [z_j, z_k]$ mit den Knoten $z_j, z_k \in \mathcal{K}_\ell$ ist gespeichert als

$$\text{elements}(i, :) = [j \ k],$$

wobei die Knoten entgegen dem Uhrzeigersinn angeordnet sind, d.h. die Parametrisierung des Randes $E_i \subseteq \Gamma$ ist mathematisch positiv. Damit ist gewährleistet, dass der äußere Einheitsnormalenvektor $n_i \in \mathbb{R}^2$ auf Γ auf einem Element $E_i = [z_j, z_k]$ durch

$$n_i = \frac{1}{|z_k - z_j|} \begin{pmatrix} y_k - y_j \\ x_j - x_k \end{pmatrix} \quad \text{mit } z_j = (x_j, y_j), z_k = (x_k, y_k)$$

gegeben ist, wobei $|\cdot|$ die euklidische Länge darstellt. Siehe auch Abbildung 1.

2.2 Funktionen am Rand

Für alle Elemente $E_j = [a_j, b_j] \in \mathcal{E}_\ell$ wenden wir die affine Parametrisierung

$$\gamma_j : [-1, 1] \rightarrow E_j, \quad \gamma_j(s) = \frac{1}{2}(a_j + b_j + s(b_j - a_j)) \quad (2.1)$$

an, welche das Referenzelement $[-1, 1] \subseteq \mathbb{R}$ bijektiv auf $E_j = [a_j, b_j] \subseteq \mathbb{R}^2$ abbildet.

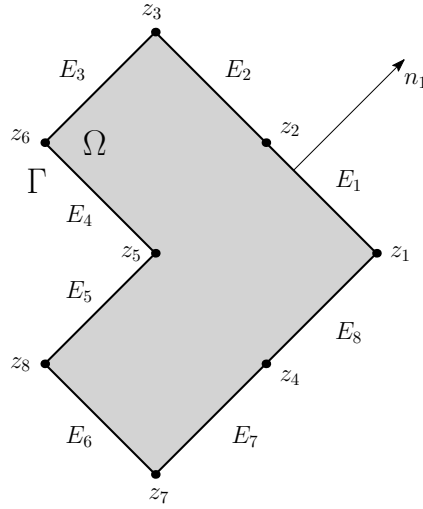


Abbildung 1: Beispiel-Gebiet mit 8 Elementen

Sei $\mathcal{P}^p(\mathcal{E}_\ell)$ der **Raum aller $\mathcal{E}_\ell$ -stückweisen Polynome** vom Grad $p \in \mathbb{N}$ bezüglich der Bogenlänge, dann gilt, dass alle Funktionen $f_\ell \in \mathcal{P}^p(\mathcal{E}_\ell)$ auf den Elementen $E_j \in \mathcal{E}_\ell$ mit $f_\ell \circ \gamma_j : [-1, 1] \rightarrow \mathbb{R}$

$$f_\ell \circ \gamma_j \in \mathcal{P}^p[-1, 1]$$

erfüllen. Im Allgemeinen weisen die Funktionen $f_\ell \in \mathcal{P}^p(\mathcal{E}_\ell)$ Sprünge an den Knoten von $\mathcal{E}_\ell$ auf und sind daher nicht stetig.

Insbesondere ist $\mathcal{P}^0(\mathcal{E}_\ell)$ der Raum aller $\mathcal{E}_\ell$ -stückweise konstanten Funktionen mit der Basis $\{\chi_1, \dots, \chi_N\}$, wobei

$$\chi_j(x) = \begin{cases} 1, & \text{falls } x \in E_j \\ 0, & \text{falls } x \notin E_j \end{cases}$$

die Indikatorfunktion auf E_j darstellt.

Weiters definieren wir $\mathcal{E}_\ell$ -stückweise die **lokale Netzweite** $h_\ell \in \mathcal{P}^0(\mathcal{E}_\ell)$ mit

$$h_\ell|_{E_j} := \text{length}(E_j), \quad \text{für alle } E_j \in \mathcal{E}_\ell,$$

wobei $\text{length}(E_j) = |b_j - a_j|$ den euklidischen Abstand darstellt.

Abschließend bezeichnet $\mathcal{S}^1(\mathcal{E}_\ell) := \mathcal{P}^1(\mathcal{E}_\ell) \cap \mathcal{C}(\Gamma)$ den **Raum der global stetigen und $\mathcal{E}_\ell$ -stückweise affinen Funktionen** mit der Basis $\{\zeta_1, \dots, \zeta_N\}$, wobei $\zeta_j \in \mathcal{S}^1(\mathcal{E}_\ell)$ die zum Knoten $z_j \in \mathcal{K}_\ell$ gehörige Hutfunktion ist.

2.3 Randintegrale

Sei $I \subseteq \mathbb{R}$ ein kompaktes Intervall in $\mathbb{R}$. Für $E_j \in \mathcal{E}_\ell$ sei $\pi_j : I \rightarrow E_j$ eine stetig differenzierbare und bijektive Abbildung. Für jede Funktion $f : E_j \rightarrow \mathbb{R}$ ist folglich das Randintegral definiert

als

$$\int_{E_j} f d\Gamma = \int_{E_j} f(x) d\Gamma(x) := \int_I (f \circ \pi_j)(t) |\pi_j'(t)| dt.$$

Diese Definition ist unabhängig von der gewählten Parametrisierung π_j . Für die Referenzparametrisierung γ_j aus (2.1) gilt

$$\int_{E_j} f d\Gamma = \frac{\text{length}(E_j)}{2} \int_{-1}^1 f \circ \gamma_j dt$$

und für die Parametrisierung bezüglich der Bogenlänge

$$\beta_j : [0, \text{length}(E_j)] \rightarrow E_j, \quad \beta_j(t) := a_j + \frac{t}{\text{length}(E_j)} (b_j - a_j) \quad (2.2)$$

gilt

$$\int_{E_j} f d\Gamma = \int_0^{\text{length}(E_j)} f \circ \beta_j dt.$$

2.4 Bogenlängenableitung

Bezeichne $I \subseteq \mathbb{R}$ wieder ein kompaktes Intervall in $\mathbb{R}$. Weiters sei $\pi_j : I \rightarrow E_j$ für $E_j \in \mathcal{E}_\ell$ eine stetig differenzierbare und bijektive Abbildung mit $|\pi_j'(t)| > 0$ für alle $t \in I$. Dann ist für jede Funktion $f : E_j \rightarrow \mathbb{R}$ die Bogenlängenableitung definiert als

$$(f' \circ \pi_j)(t) = \frac{1}{|\pi_j'(t)|} (f \circ \pi_j)'(t) \quad \text{für alle } x = \pi_j(t) \in E_j.$$

Da diese Definition unabhängig von der gewählten Parametrisierung ist, gilt für unsere Referenzparametrisierung γ_j aus (2.1)

$$f' \circ \gamma_j = \frac{2}{\text{length}(E_j)} (f \circ \gamma_j)' \quad \text{für alle } x = \gamma_j(t) \in E_j.$$

Für die Bogenlängen-Parametrisierung β_j aus (2.2) gilt hingegen

$$f' \circ \beta_j = (f \circ \beta_j)'.$$

3 Symm'sche Integralgleichung

In diesem Kapitel befassen wir uns mit der Symm'schen Integralgleichung als Folgerung des Dirichlet-Randwertproblems (1.6). Durch Diskretisierung einer äquivalenten Variationsformulierung können wir die, unter bestimmten Voraussetzungen, eindeutige Lösung $\phi \in H^{-1/2}(\Gamma)$ durch eine Galerkin-Lösung $\Phi_\ell \in \mathcal{P}^0(\mathcal{E}_\ell)$ approximieren. Mit Satz 3.6 erhalten wir für diese Approximation auf uniformen Netzen und hinreichend glatten Daten eine Konvergenzrate $\|\phi - \Phi_\ell\|_V = \mathcal{O}(h^{3/2})$. Weiters führen wir in diesem Kapitel den wichtigen Begriff der Saturationsannahme ein, dem wir im Laufe dieser Arbeit noch öfter begegnen werden.

3.1 Variationsformulierung und eindeutige Lösbarkeit

Bevor wir uns mit dem Dirichlet-Randwertproblem (1.6) beschäftigen, definieren wir den für die (schwache) Lösbarkeit elliptischer Differentialgleichungen benötigten Begriff der **Sobolev-Räume**. Dabei definieren wir den Raum $H^1(\Gamma)$ gemäß [18, Abschnitt 2.5]. Weiters übernehmen wir aus [6, Abschnitt 2] die Definition des Sobolev-Raumes $H^{1/2}(\Gamma)$: Mit dem zum beschränkten Gebiet $\Omega \subseteq \mathbb{R}^2$ gehörigen Lipschitz-Rand Γ besteht der Sobolev-Raum $H^{1/2}(\Gamma)$ aus allen Funktionen $u : \Gamma \rightarrow \mathbb{R}$, welche Spur einer H^1 -Funktion sind, d.h.

$$H^{1/2}(\Gamma) := \{u|_\Gamma : u \in H^1(\mathbb{R}^2)\}.$$

Als die zugehörige Norm definieren wir die **Sobolev-Slobodeckij-Norm**

$$\|u\|_{H^{1/2}(\Gamma)} := \left(\|u\|_{L^2(\Gamma)}^2 + |u|_{H^{1/2}(\Gamma)}^2 \right)^{1/2} \quad (3.1)$$

mit der Halbnorm

$$|u|_{H^{1/2}(\Gamma)}^2 := \int_\Gamma \int_\Gamma \frac{|u(x) - u(y)|^2}{|x - y|^2} d\Gamma(x) d\Gamma(y). \quad (3.2)$$

und als zugehörigen Dualraum setzen wir

$$H^{-1/2}(\Gamma) := H^{1/2}(\Gamma)'. \quad (3.3)$$

Aufgrund der Dichtheit von $L^2(\Gamma)$ in $H^{-1/2}(\Gamma)$, vgl. [12, §3 - Sobolev Spaces], gibt es für jedes $\psi \in H^{-1/2}(\Gamma)$ eine Folge $\psi_n \in L^2(\Gamma)$ mit $\psi_n \rightarrow \psi$ in $H^{-1/2}(\Gamma)$. Da nun alle Elemente $\psi \in H^{-1/2}(\Gamma)$ lineare und stetige Funktionale sind, definieren wir das **erweiterte L^2 -Skalarprodukt** durch

$$\langle \nu, \psi \rangle_\Gamma := \langle \nu, \lim_{n \rightarrow \infty} \psi_n \rangle_\Gamma = \lim_{n \rightarrow \infty} \langle \nu, \psi_n \rangle_\Gamma$$

für $\psi \in H^{-1/2}(\Gamma)$ und $\nu \in H^{1/2}(\Gamma)$. Demnach gilt für $\nu \in H^{1/2}(\Gamma) \subseteq L^2(\Gamma)$ und $v \in L^2(\Gamma)$ das gewöhnliche L^2 -Skalarprodukt

$$\langle \nu, v \rangle_\Gamma = \int_\Gamma \nu(x) v(x) d\Gamma(x).$$

In weiterer Folge können wir mit dem erweiterten $L^2(\Gamma)$ -Skalarprodukt $\langle \cdot, \cdot \rangle_\Gamma$ auf $H^{-1/2}(\Gamma)$ die Norm

$$\|\psi\|_{H^{-1/2}(\Gamma)} := \sup_{\substack{\nu \neq 0 \\ \nu \in H^{1/2}(\Gamma)}} \frac{|\langle \nu, \psi \rangle_\Gamma|}{\|\nu\|_{H^{1/2}(\Gamma)}} \quad (3.3)$$

definieren. Aus der Funktionalanalysis wissen wir, dass der Hilbertraum $H^{1/2}(\Gamma)$ reflexiv ist, siehe beispielsweise [11, Theorem 4.6-6]. Folglich gilt für den Dualraum von $H^{-1/2}(\Gamma)$

$$H^{-1/2}(\Gamma)' = H^{1/2}(\Gamma)'' = H^{1/2}(\Gamma).$$

Zusammenfassend gilt mit $H^0(\Gamma) = L^2(\Gamma)$ die allgemeine Beziehung

$$H^1(\Gamma) \subseteq H^{1/2}(\Gamma) \subseteq L^2(\Gamma) \subseteq H^{-1/2}(\Gamma).$$

Mit diesen Begriffen ausgestattet, gehen wir nun auf das Dirichlet-Randwertproblem (1.6)

$$\begin{aligned} -\Delta u &= 0 & \text{in } \Omega \\ u &= g & \text{auf } \Gamma \end{aligned}$$

ein und orientieren uns bei der weiteren Vorgangsweise an der Dokumentation von [1, Kapitel 5]. Die erste Gleichung in (1.5) vereinfacht sich nun zur **Symm'schen Integralgleichung**

$$V\phi = (K + 1/2)g \quad \text{auf } \Gamma, \quad (3.4)$$

mit den beschränkten, linearen Operatoren $V : H^{-1/2}(\Gamma) \rightarrow H^{1/2}(\Gamma)$

$$(V\phi)(x) := \int_\Gamma U(x, y) \phi(y) d\Gamma(y) = -\frac{1}{2\pi} \int_\Gamma \log|x - y| \phi(y) d\Gamma(y) \quad (3.5)$$

und $K : H^{1/2}(\Gamma) \rightarrow H^{1/2}(\Gamma)$

$$(Kg)(x) := \int_\Gamma \frac{\partial}{\partial n_y} U(x, y) g(y) d\Gamma(y) = -\frac{1}{2\pi} \int_\Gamma \frac{(y - x) \cdot n_y}{|y - x|^2} g(y) d\Gamma(y) \quad (3.6)$$

für alle $x \in \Gamma$, siehe etwa [18, Abschnitt 6.2 und 6.4]. Dabei bezeichnet V das **Einfachschichtpotential** und K das **Doppelschichtpotential**. Auf einem Lipschitz-Gebiet gilt sogar mehr (vgl. [18, Satz 6.11]):

Lemma 3.1 *Sei Γ der Rand eines Lipschitz-Gebietes Ω . Dann sind die linearen Randintegraloperatoren*

$$\begin{aligned} V &: H^{-1/2+s}(\Gamma) \rightarrow H^{1/2+s}(\Gamma) \\ K &: H^{1/2+s}(\Gamma) \rightarrow H^{1/2+s}(\Gamma) \end{aligned}$$

für alle $s \in [-\frac{1}{2}, \frac{1}{2}]$ beschränkt. □

Weiters vereinfacht sich auch die Darstellungsformel (1.2), sodass sich die Lösung u bei bekannten Cauchy-Daten $g \in H^{1/2}(\Gamma)$ und $\phi = \frac{\partial}{\partial n} u \in H^{-1/2}(\Gamma)$ auf dem Gebiet Ω durch

$$u(\tilde{x}) = \tilde{V}\phi(\tilde{x}) - \tilde{K}g(\tilde{x}) \quad \text{für alle } \tilde{x} \in \Omega \quad (3.7)$$

berechnen lässt. Die Operatoren $\tilde{V}$ und $\tilde{K}$ sind dabei jene aus (1.3) bzw. (1.4).

Wir suchen nun bei gegebenen Randdaten $g \in H^{1/2}(\Gamma)$ die fehlenden Neumann-Daten $\phi \in H^{-1/2}(\Gamma)$, um die Lösung u von (1.6) mittels (3.7) berechnen zu können. Dazu schreiben wir die Symm'sche Integralgleichung (3.4) als äquivalente **Variationsformulierung**

$$\langle V\phi, \psi \rangle_\Gamma = \langle (K + 1/2)g, \psi \rangle_\Gamma \quad \text{für alle } \psi \in H^{-1/2}(\Gamma) \quad (3.8)$$

an. Zum Einen ist jede Lösung $\phi \in H^{-1/2}(\Gamma)$ von (3.4) auch Lösung von (3.8). Umgekehrt gilt für jede Lösung $\phi \in H^{-1/2}(\Gamma)$ von (3.8) wegen $H^{-1/2}(\Gamma)' = H^{1/2}(\Gamma)$ und (3.3) nach Hahn-Banach

$$\|V\phi - (K + 1/2)g\|_{H^{1/2}(\Gamma)} = \sup_{\substack{\psi \neq 0 \\ \psi \in H^{-1/2}(\Gamma)}} \frac{|\langle V\phi - (K + 1/2)g, \psi \rangle_\Gamma|}{\|\psi\|_{H^{-1/2}(\Gamma)}} = 0,$$

was äquivalent zu $0 = V\phi - (K + 1/2)g$ ist, siehe auch Beweisführung in [18, Abschnitt 3.1]. Mit der zusätzlichen Voraussetzung $\text{diam}(\Omega) < 1$ an das Gebiet $\Omega \subseteq \mathbb{R}^2$ folgt, dass $V : H^{-1/2}(\Gamma) \rightarrow H^{1/2}(\Gamma)$ elliptisch ist, d.h. es gilt

$$\langle V\psi, \psi \rangle_\Gamma \geq C \|\psi\|_{H^{-1/2}(\Gamma)}^2 \quad (3.9)$$

mit einer Konstanten $C > 0$, wobei wir für den Beweis auf [18, Satz 6.6] verweisen. Da $V : H^{-1/2}(\Gamma) \rightarrow H^{1/2}(\Gamma)$ beschränkt ist, folgt mit dem Lemma von Lax-Milgram, dass (3.8) eine **eindeutige Lösung** besitzt, die stetig von den Anfangsdaten abhängt, d.h. $\|\phi\|_{H^{-1/2}(\Gamma)} \leq C \|g\|_{H^{1/2}(\Gamma)}$ mit einer Konstanten $C > 0$. Damit ist V von $H^{-1/2}(\Gamma)$ nach $H^{1/2}(\Gamma)$ ein stetiger, linearer und bijektiver Operator.

Auf $H^{-1/2}(\Gamma)$ erhalten wir dann ein weiteres Skalarprodukt durch

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

Die Bilinearität folgt unmittelbar aus der Linearität des Einfachschichtpotentials V und der des Integrals. Wegen der Elliptizität (3.9) des Einfachschichtpotentials gilt

$$\langle\langle \phi, \phi \rangle\rangle_V = \langle V\phi, \phi \rangle_\Gamma \geq C \|\phi\|_{H^{-1/2}(\Gamma)}^2 \quad \text{für alle } \phi \in H^{-1/2}(\Gamma),$$

daher ist $\langle\langle \cdot, \cdot \rangle\rangle_V$ positiv definit. Es bleibt noch die Symmetrie zu zeigen: Da $L^2(\Gamma) \subseteq H^{-1/2}(\Gamma)$ dicht liegt, gibt es zu jedem $\psi \in H^{-1/2}(\Gamma)$ eine Folge $(\psi_n)_{n \in \mathbb{N}}$ in $L^2(\Gamma)$ mit $\psi_n \rightarrow \psi$ in $H^{-1/2}(\Gamma)$. Analog bezeichne $(\phi_m)_{m \in \mathbb{N}}$ eine Folge aus $L^2(\Gamma)$ mit $\phi_m \rightarrow \phi$ in $H^{-1/2}(\Gamma)$. Dann gilt für alle $\psi_n \in L^2(\Gamma)$ mit Cauchy-Schwarz und wegen $V\phi_m \in H^{1/2}(\Gamma) \subseteq L^2(\Gamma)$

$$\int_\Gamma |V\phi_m(x) \psi_n(x)| d\Gamma(x) \leq \|V\phi_m\|_{L^2(\Gamma)} \|\psi_n\|_{L^2(\Gamma)} < \infty.$$

Also ist $V\phi_m \psi_n$ integrierbar und wir erhalten mit dem Satz von Fubini

$$\begin{aligned} \langle\langle \phi_m, \psi_n \rangle\rangle_V &= \langle V\phi_m, \psi_n \rangle_\Gamma = \int_\Gamma \left(\int_\Gamma U(x, y) \phi_m(y) d\Gamma(y) \right) \psi_n(x) d\Gamma(x) \\ &= \int_\Gamma \left(\int_\Gamma U(x, y) \psi_n(x) d\Gamma(x) \right) \phi_m(y) d\Gamma(y) = \langle V\psi_n, \phi_m \rangle_\Gamma = \langle\langle \psi_n, \phi_m \rangle\rangle_V \end{aligned}$$

für alle $m, n \in \mathbb{N}$. Aufgrund der Stetigkeit von V und der des Skalarproduktes $\langle \cdot, \cdot \rangle_\Gamma$ folgt dann

$$\langle\langle \phi, \psi \rangle\rangle_V = \lim_{m \rightarrow \infty} \lim_{n \rightarrow \infty} \langle\langle \phi_m, \psi_n \rangle\rangle_V = \lim_{m \rightarrow \infty} \lim_{n \rightarrow \infty} \langle\langle \psi_n, \phi_m \rangle\rangle_V = \langle\langle \psi, \phi \rangle\rangle_V$$

für alle $\phi, \psi \in H^{-1/2}(\Gamma)$.

Die durch $\langle\langle \cdot, \cdot \rangle\rangle_V$ induzierte Norm $\|\phi\|_V := \langle\langle \phi, \phi \rangle\rangle_V^{1/2}$ bezeichnen wir als die **Energienorm**. Sie ist eine äquivalente Norm auf $H^{-1/2}(\Gamma)$, denn einerseits gilt für alle $\psi \in H^{-1/2}$ wegen (3.9),

$$\|\psi\|_V \geq C \|\psi\|_{H^{-1/2}(\Gamma)}$$

und andererseits folgt aus der Definition (3.3)

$$\|\psi\|_{H^{-1/2}(\Gamma)} = \sup_{\substack{\nu \neq 0 \\ \nu \in H^{1/2}(\Gamma)}} \frac{|\langle \nu, \psi \rangle_\Gamma|}{\|\nu\|_{H^{1/2}(\Gamma)}} \geq \frac{|\langle V\psi, \psi \rangle_\Gamma|}{\|V\psi\|_{H^{1/2}(\Gamma)}} = \frac{\|\psi\|_V^2}{\|V\psi\|_{H^{1/2}(\Gamma)}}.$$

Durch Umformulierung impliziert die Beschränktheit von V mit $C_V := \|V\|$

$$\|\psi\|_V^2 \leq \|V\psi\|_{H^{1/2}(\Gamma)} \|\psi\|_{H^{-1/2}(\Gamma)} \leq C_V \|\psi\|_{H^{-1/2}(\Gamma)}^2$$

und folglich die Äquivalenz $\|\cdot\|_{H^{-1/2}(\Gamma)} \sim \|\cdot\|_V$.

3.2 Galerkin-Diskretisierung

Wir übernehmen wieder aus der Dokumentation von [1, Kapitel 5, S. 32 f.] und nehmen an, dass die Dirichlet-Daten $g \in H^1(\Gamma) \subseteq H^{1/2}(\Gamma)$ erfüllen. Da $\Gamma \subseteq \mathbb{R}^2$ eine eindimensionale Mannigfaltigkeit ist, ist g aufgrund des Einbettungssatzes von Sobolev insbesondere stetig. Nach den Vorüberlegungen in Abschnitt 2.2, können wir nun (3.8) diskretisieren, indem wir die stetigen Dirichlet-Daten $g \in H^1(\Gamma)$ durch den nodalen Interpolanten

$$G_\ell := \sum_{j=1}^N g(z_j) \zeta_j$$

approximieren, wobei $G_\ell \in \mathcal{S}^1(\mathcal{E}_\ell) \subseteq H^1(\Gamma)$ gilt. Weiters wird der Raum $H^{-1/2}(\Gamma)$ durch den endlich dimensionalen Raum $\mathcal{P}^0(\mathcal{E}_\ell)$ ersetzt. Da $\mathcal{P}^0(\mathcal{E}_\ell) \subseteq H^{-1/2}(\Gamma)$ ein Unterraum ist, ist $\langle\langle \cdot, \cdot \rangle\rangle_V$ auch ein Skalarprodukt auf $\mathcal{P}^0(\mathcal{E}_\ell)$. Somit existiert auch eine eindeutige Galerkin-Lösung $\Phi_\ell \in \mathcal{P}^0(\mathcal{E}_\ell)$ von

$$\langle V\Phi_\ell, \Psi_\ell \rangle_\Gamma = \langle (K + 1/2) G_\ell, \Psi_\ell \rangle_\Gamma \quad \text{für alle } \Psi_\ell \in \mathcal{P}^0(\mathcal{E}_\ell). \quad (3.10)$$

Aus der Linearen Algebra ist bekannt, dass die lineare Gleichung (3.10) genau dann für alle $\Psi_\ell \in \mathcal{P}^0(\mathcal{E}_\ell)$ gilt, wenn sie für alle Basisfunktionen $\chi_j \in \{\chi_1, \dots, \chi_N\}$ von $\mathcal{P}^0(\mathcal{E}_\ell)$ gilt. Mit dem Koeffizientenvektor $\mathbf{x} = (x_1, \dots, x_N) \in \mathbb{R}^N$ setzen wir

$$\Phi_\ell = \sum_{j=1}^N x_j \chi_j.$$

Weiters sei der Vektor $\mathbf{g} = (g_1, \dots, g_N) \in \mathbb{R}^N$ definiert durch $g_j := g(z_j)$ für alle $z_j \in \mathcal{K}_\ell$. Damit ist (3.10) äquivalent zu

$$\sum_{j=1}^N x_j \langle V \chi_j, \chi_k \rangle_\Gamma = \langle V \Phi_\ell, \chi_k \rangle_\Gamma = \langle (K + 1/2) G_\ell, \chi_k \rangle_\Gamma = \sum_{j=1}^N g_j \langle (K + 1/2) \zeta_j, \chi_k \rangle_\Gamma$$

für alle $k \in \{1, \dots, N\}$. Mit den Definitionen

$$\mathbf{V}_{kj} = \langle V \chi_j, \chi_k \rangle_\Gamma \quad (3.11)$$

$$\mathbf{K}_{kj} = \langle K \zeta_j, \chi_k \rangle_\Gamma \quad (3.12)$$

$$\mathbf{M}_{kj} = \langle \zeta_j, \chi_k \rangle_\Gamma \quad (3.13)$$

für alle $j, k \in \{1, \dots, N\}$ erhalten wir schlussendlich ein zu (3.10) äquivalentes lineares Gleichungssystem

$$\mathbf{V} \mathbf{x} = \mathbf{K} \mathbf{g} + \frac{1}{2} \mathbf{M} \mathbf{g} =: \mathbf{b}. \quad (3.14)$$

Da die Matrix $\mathbf{V} \in \mathbb{R}^{N \times N}$ von einem Skalarprodukt stammt, ist sie symmetrisch und positiv definit. Somit hat (3.14) eine eindeutige Lösung $\mathbf{x} \in \mathbb{R}^N$.

3.3 Dirichlet-Datenoszillationen

Da wir im Zuge der Diskretisierung die Randdaten g durch seinen nodalen Interpolanten G_ℓ approximieren, lösen wir anstelle von (3.8) die gestörte Variationsformulierung

$$\langle V \phi_\ell, \psi \rangle_\Gamma = \langle (K + 1/2) G_\ell, \psi \rangle_\Gamma \quad \text{für alle } \psi \in H^{-1/2}(\Gamma). \quad (3.15)$$

Der Fehler zwischen der exakten Lösung ϕ von (3.8) und der exakten Lösung ϕ_ℓ von (3.15) ist beschränkt durch die **Dirichlet-Datenoszillationen**¹ (siehe [1, Formel (5.11)])

$$\|\phi - \phi_\ell\|_V \lesssim \|h_\ell^{1/2} (g - G_\ell)'\|_{L^2(\Gamma)} =: \text{osc}_{D,\ell}, \quad (3.16)$$

wobei $(\cdot)'$ die Bogenlängenableitung aus Abschnitt 2.4 bezeichnet. Es gilt dabei

$$\text{osc}_{D,\ell}^2 = \sum_{j=1}^N \|h_\ell^{1/2} (g - G_\ell)'\|_{L^2(E_j)}^2 =: \sum_{j=1}^N \text{osc}_{D,\ell}(E_j)^2. \quad (3.17)$$

¹Mit der Notation $\lesssim$ unterdrücken wir generische Konstanten, d.h. für $a, b \in \mathbb{R}$ gilt $a \lesssim b$ genau dann, wenn $a \leq C b$ mit einer nur von der K-Netzkonstante $\kappa(\mathcal{E}_\ell)$ abhängigen Konstante $C > 0$ gilt, vgl. Kapitel 4.

Unser Ziel ist es nun, die lokalen Beiträge $\text{osc}_{D,\ell}(E_j)$ zu approximieren, damit wir den Fehler $\|\phi - \phi_\ell\|_V$ quantifizieren können.

Hierzu werden wir $g \circ \gamma_j$ auf $E_j = [a_j, b_j] \in \mathcal{E}_\ell$ durch ein Polynom aus $\mathcal{P}^2[-1, 1]$ approximieren, wobei $\gamma_j : [-1, 1] \rightarrow E_j$ die Referenzparametrisierung (2.1) bezeichnet.

Wir gehen wie in [1, Abschnitt 5.1] vor: Mit $h := \text{length}(E_j)$ gilt für eine Funktion $v \in H^1(\Gamma)$ und den Vorüberlegungen aus den Abschnitten 2.3 und 2.4

$$\|v'\|_{L^2(E_j)}^2 = \int_{E_j} (v')^2 d\Gamma = \frac{h}{2} \int_{-1}^1 ((v' \circ \gamma_j)(s))^2 ds = \frac{2}{h} \int_{-1}^1 ((v \circ \gamma_j)'(s))^2 ds.$$

Wir approximieren nun $p_j \approx v \circ \gamma_j : [-1, 1] \rightarrow [a_j, b_j] \rightarrow \mathbb{R}$ durch $p_j \in \mathcal{P}^2[-1, 1]$ mit

$$p_j(-1) = v(a_j), \quad p_j(0) = v(m_j) \quad p_j(1) = v(b_j),$$

wobei $m_j = (a_j + b_j)/2$ der Mittelpunkt von $E_j \in \mathcal{E}_\ell$ ist. Damit können wir nun $\|v'\|_{L^2(E_j)}$ durch

$$\|v'\|_{L^2(E_j)}^2 = \frac{2}{h} \int_{-1}^1 ((v \circ \gamma_j)'(s))^2 ds \approx \frac{2}{h} \int_{-1}^1 (p_j')^2 ds = \frac{2}{h} \text{quad}_2((p_j')^2),$$

approximieren, wobei $\text{quad}_2(\cdot)$ eine Quadraturregel auf $[-1, 1]$ bezeichnet, welche auf $\mathcal{P}^2[-1, 1]$ exakt ist. Da $p_j \in \mathcal{P}^2[-1, 1]$ ist, gilt $p_j' \in \mathcal{P}^1[-1, 1]$ bzw. auch $(p_j')^2 \in \mathcal{P}^2[-1, 1]$ und folglich die rechte Gleichheit in obiger Formel. Für die Quadratur verwenden wir eine 3-Punkt-Newton-Côtes Formel mit den Knoten $s_1 = -1$, $s_2 = 0$ und $s_3 = 1$, welche aufgrund der geraden Ordnung von p_j sogar auf $\mathcal{P}^3[-1, 1]$ exakt ist, und schreiben

$$\begin{aligned} p_j &= v(a_j)L_1 + v(m_j)L_2 + v(b_j)L_3, \\ p_j' &= v(a_j)L_1' + v(m_j)L_2' + v(b_j)L_3', \end{aligned} \tag{3.18}$$

wobei L_i die Lagrange-Polynome zu den Knoten s_i bezeichnen. Es gilt

$$L_1(s) = \frac{s(s-1)}{2}, \quad L_2(s) = 1 - s^2, \quad L_3(s) = \frac{s(s+1)}{2}$$

mit ihren Ableitungen

$$L_1'(s) = \frac{2s-1}{2}, \quad L_2'(s) = -2s, \quad L_3'(s) = \frac{2s+1}{2}.$$

Wir können nun (3.18) als Matrix-Vektor-Multiplikation anschreiben und erhalten an den Knoten s_i

$$\begin{pmatrix} p_j'(-1) \\ p_j'(0) \\ p_j'(1) \end{pmatrix} = \begin{pmatrix} L_1'(-1) & L_2'(-1) & L_3'(-1) \\ L_1'(0) & L_2'(0) & L_3'(0) \\ L_1'(1) & L_2'(1) & L_3'(1) \end{pmatrix} \begin{pmatrix} v(a_j) \\ v(m_j) \\ v(b_j) \end{pmatrix} = \begin{pmatrix} -3/2 & +2 & -1/2 \\ -1/2 & 0 & +1/2 \\ +1/2 & -2 & +3/2 \end{pmatrix} \begin{pmatrix} v(a_j) \\ v(m_j) \\ v(b_j) \end{pmatrix}.$$

Setzen wir nun $v := g - G_\ell$, so folgt $g(a_j) = G_\ell(a_j)$ bzw. $g(b_j) = G_\ell(b_j)$ aufgrund der linearen Interpolation von G_ℓ . Damit gilt nun $v(a_j) = v(b_j) = 0$ und wir erhalten mit $G_\ell(m_j) = (g(a_j) +$

$g(b_j))/2$

$$\begin{pmatrix} p'_j(-1) \\ p'_j(0) \\ p'_j(+1) \end{pmatrix} = \begin{pmatrix} 2v(m_j) \\ 0 \\ -2v(m_j) \end{pmatrix} = (g(m_j) - G(m_j)) \begin{pmatrix} 2 \\ 0 \\ -2 \end{pmatrix} = (g(a_j) + g(b_j) - 2g(m_j)) \begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix}. \quad (3.19)$$

Für die lokalen Beiträge $\text{osc}_{D,\ell}(E_j)^2$ der Datenoszillationen erhalten wir mit obigen Überlegungen

$$\begin{aligned} \text{osc}_{D,\ell}(E_j)^2 &= h \|(g - G_\ell)'\|_{L^2(E_j)}^2 = h \int_{E_j} ((g - G_\ell)')^2 d\Gamma = \frac{h^2}{2} \int_{-1}^1 ((g - G_\ell)' \circ \gamma_j)^2(s) ds \\ &= 2 \int_{-1}^1 \left(((g - G_\ell) \circ \gamma_j)' \right)^2(s) ds \approx 2 \text{quad}_2((p'_j)^2) =: \widetilde{\text{osc}}_{D,\ell}(E_j)^2 \end{aligned}$$

mit den approximierten lokalen Dirichlet-Datenoszillationen $\widetilde{\text{osc}}_{D,\ell}(E_j)$. Da $p_j \in \mathcal{P}^2[-1, 1]$ und $G_\ell \in \mathcal{S}^1(\mathcal{E}_\ell)$, existiert elementweise ein Polynom $\tilde{p}_j \in \mathcal{P}^2(E_j)$ mit $(\tilde{p}_j - G_\ell) \circ \gamma_j = p_j$. Nun folgt analog obiger Rechnung

$$\begin{aligned} \widetilde{\text{osc}}_{D,\ell}(E_j)^2 &= 2 \text{quad}_2((p'_j)^2) = 2 \int_{-1}^1 \left(((\tilde{p}_j - G_\ell) \circ \gamma_j)' \right)^2(s) ds \\ &= \frac{h^2}{2} \int_{-1}^1 ((\tilde{p}_j - G_\ell)' \circ \gamma_j)^2(s) ds = h \|(\tilde{p}_j - G_\ell)'\|_{L^2(E_j)}^2. \end{aligned}$$

Insgesamt approximieren wir

$$\text{osc}_{D,\ell} \approx \widetilde{\text{osc}}_{D,\ell} := \left(\sum_{j=1}^N \widetilde{\text{osc}}_{D,\ell}(E_j)^2 \right)^{1/2} = \|h_\ell^{1/2} (\tilde{p} - G_\ell)'\|_{L^2(\Gamma)}, \quad (3.20)$$

wobei $\tilde{p} \in \mathcal{P}^2(\mathcal{E}_\ell)$ das zusammengesetzte Polynom mit $\tilde{p}|_{E_j} \circ \gamma_j = \tilde{p}_j \circ \gamma_j$ ist. Mit den Gewichten $\omega_i = \int_{-1}^1 L_i(s) ds$ der Newton-Côtes Formel

$$\omega_1 = \frac{1}{3}, \quad \omega_2 = \frac{4}{3} \quad \text{und} \quad \omega_3 = \frac{1}{3}$$

gilt schlussendlich mit (3.19)

$$\begin{aligned} \text{osc}_{D,\ell}(E_j)^2 &\approx \widetilde{\text{osc}}_{D,\ell}(E_j)^2 = 2 \text{quad}_2((p'_j)^2) = 2 \sum_{i=1}^3 \omega_i (p'_j(s_i))^2 \\ &= \frac{4}{3} (g(a_j) + g(b_j) - 2g(m_j))^2. \end{aligned} \quad (3.21)$$

Um den Quadraturfehler analysieren zu können, der aufgrund der Approximation $\widetilde{\text{osc}}_{D,\ell}(E_j) \approx \text{osc}_{D,\ell}(E_j)$ entsteht, brauchen wir folgendes wichtiges Lemma über die Integration der Ableitung, welches wir noch öfter benötigen werden. Wir verweisen für den Beweis auf [14, Satz 15].

Lemma 3.2 Seien $m \in \mathbb{N}_0$ und $f \in \mathcal{C}^{m+1}$. Weiters sei $p \in \mathcal{P}^m$ das Interpolationspolynom zu beliebigen Knoten $a \leq t_0 < \dots < t_m \leq b$. Dann gilt

$$\|f' - p'\|_{L^\infty[a,b]} \leq C \frac{\|f^{(m+1)}\|_{L^\infty[a,b]}}{m!} (b-a)^m,$$

wobei $C > 0$ nur von m abhängt. □

Nun kommen wir zu folgendem

Satz 3.3 (Quadraturfehler-Analyse). Für glatte Dirichlet-Daten g und uniforme Netze mit Netzweite h gilt:

- (a) $\text{osc}_{D,\ell} \geq C h^{3/2} \|g''\|_{L^2(\Gamma)}$ für $g \in \mathcal{P}^m(\Gamma) \setminus \mathcal{P}^1(\Gamma)$, $m \geq 2$ beliebig und C abhängig von m .
- (b) $\text{osc}_{D,\ell} = \mathcal{O}(h^{3/2})$ für $g \in \mathcal{C}^2(\Gamma)$.
- (c) $|\text{osc}_{D,\ell} - \widetilde{\text{osc}}_{D,\ell}| = \mathcal{O}(h^{5/2})$ für $g \in \mathcal{C}^3(\Gamma)$ gilt.

Der Quadraturfehler hat somit für $g \in \mathcal{P}^m(\Gamma) \setminus \mathcal{P}^1(\Gamma)$ und $m \geq 2$ eine höhere Ordnung verglichen mit der Diskretisierungsordnung.

Für den Beweis von (a) benötigen wir noch ein Lemma aus der Funktionalanalysis:

Lemma 3.4 Seien $|\cdot|_p$ und $|\cdot|_q$ zwei Seminormen auf dem Vektorraum X mit $\dim X = n < \infty$. Seien N_p der Kern von $|\cdot|_p$, d.h. $N_p := \ker |\cdot|_p = \{\mathbf{x} \in X : |\mathbf{x}|_p = 0\}$ und analog N_q der Kern von $|\cdot|_q$.

Sei $N_p \subseteq N_q$, dann existiert eine Konstante $C > 0$, sodass

$$|\mathbf{x}|_q \leq C |\mathbf{x}|_p$$

für alle $\mathbf{x} \in X$ gilt.

BEWEIS: Wir unterteilen den Beweis in mehrere Schritte:

- (i) N_p ist ein Unterraum von X :

Für alle $\mathbf{x}, \mathbf{y} \in N_p$ liegt auch $\mathbf{x} + c\mathbf{y} \in N_p$ mit $c \in \mathbb{R}$, denn

$$0 \leq |\mathbf{x} + c\mathbf{y}|_p \leq |\mathbf{x}|_p + |c| |\mathbf{y}|_p = 0.$$

- (ii) Definiere $n_p := \dim N_p$ und $n_q := \dim N_q$, wobei wegen $N_p \subseteq N_q$ die Beziehung $n_p \leq n_q \leq n$ gilt. Sei nun $\{\mathbf{b}_1, \dots, \mathbf{b}_n\}$ eine Basis von X , sodass

$$\begin{aligned} N_p &= \text{span}\{\mathbf{b}_1, \dots, \mathbf{b}_{n_p}\} \\ N_q &= \text{span}\{\mathbf{b}_1, \dots, \mathbf{b}_{n_p}, \mathbf{b}_{n_p+1}, \dots, \mathbf{b}_{n_q}\} \\ X &= \text{span}\{\mathbf{b}_1, \dots, \mathbf{b}_n\} \\ R_p &:= \text{span}\{\mathbf{b}_{n_p+1}, \dots, \mathbf{b}_n\}. \end{aligned}$$

Der Raum R_p stellt dabei den Bildbereich von $|\cdot|_p$ dar.

(iii) Auf R_p ist $|\cdot|_p$ eine Norm:

Da R_p auch ein Unterraum von X und $|\cdot|_p$ eine Seminorm auf X ist, genügt es $|\mathbf{x}|_p = \mathbf{0} \Leftrightarrow \mathbf{x} = \mathbf{0}$ nachzuweisen. Aus $\mathbf{x} \in R_p$ und $|\mathbf{x}|_p = 0$ folgt $\mathbf{x} \in R_p \cap N_p$. Damit gilt $\mathbf{x} = \mathbf{0}$. Umgekehrt folgt natürlich aus $\mathbf{x} = \mathbf{0}$ auch $|\mathbf{x}|_p = 0$.

(iv) Da $|\cdot|_p$ eine Norm auf R_p ist, existiert ein $c > 0$, sodass mit der Darstellung $\mathbf{x} = \sum_{i=n_p+1}^n x_i \mathbf{b}_i$

$$c \sum_{i=n_p+1}^n |x_i| \leq |\mathbf{x}|_p = \left| \sum_{i=n_p+1}^n x_i \mathbf{b}_i \right|_p$$

für alle $\mathbf{x} \in R_p$ gilt, siehe zum Beispiel [11, Lemma 2.4-1].

(v) Sei $\mathbf{x} = \sum_{i=1}^n x_i \mathbf{b}_i \in X$, dann gilt wegen $|\mathbf{b}_i|_q = 0$ für $i \in \{1, \dots, n_q\}$

$$|\mathbf{x}|_q \leq \sum_{i=1}^n |x_i| |\mathbf{b}_i|_q = \sum_{i=n_q+1}^n |x_i| |\mathbf{b}_i|_q \leq \sum_{i=n_q+1}^{n_q} |x_i| 1 + \sum_{i=n_q+1}^n |x_i| |\mathbf{b}_i|_q \leq K \sum_{i=n_p+1}^n |x_i|,$$

wobei $K := \max\{1, |\mathbf{b}_{n_q+1}|_q, \dots, |\mathbf{b}_n|_q\}$. Mit (iv) erhalten wir

$$\frac{c}{K} |\mathbf{x}|_q \leq c \sum_{i=n_p+1}^n |x_i| \leq \left| \sum_{i=n_p+1}^n x_i \mathbf{b}_i \right|_p.$$

Da $\left| \sum_{i=1}^{n_p} (-x_i) \mathbf{b}_i \right|_p = 0$, gilt mit umgekehrter Dreiecksungleichung

$$\frac{c}{K} |\mathbf{x}|_q \leq \left| \sum_{i=n_p+1}^n x_i \mathbf{b}_i \right|_p - \left| \sum_{i=1}^{n_p} (-x_i) \mathbf{b}_i \right|_p \leq \left| \sum_{i=n_p+1}^n x_i \mathbf{b}_i - \sum_{i=1}^{n_p} (-x_i) \mathbf{b}_i \right|_p = |\mathbf{x}|_p.$$

Mit der Konstanten $C := \frac{K}{c}$ folgt die Behauptung. $\square$

BEWEIS VON SATZ 3.3: Sei $E_j \in \mathcal{E}_\ell$ und $f^{(k)} \in L^2(E_j)$, wobei $(\cdot)^{(k)}$ die k -te Bogenlängenableitung darstellt. Dann gilt mit $|\gamma'_j| = h/2$

$$\|f^{(k)}\|_{L^2(E_j)}^2 = \int_{E_j} (f^{(k)})^2 d\Gamma = \frac{h}{2} \int_{-1}^1 (f^{(k)} \circ \gamma_j(s))^2 ds = \frac{h}{2} \left(\frac{2}{h}\right)^{2k} \int_{-1}^1 \left((f \circ \gamma_j(s))^{(k)}\right)^2 ds$$

und infolgedessen

$$\|f^{(k)}\|_{L^2(E_j)}^2 = \left(\frac{2}{h}\right)^{2k-1} \|(f \circ \gamma_j)^{(k)}\|_{L^2(-1,1)}^2. \quad (3.22)$$

Da $\|(\cdot)'\|_{L^2(-1,1)}$ und $\|(\cdot)''\|_{L^2(-1,1)}$ zwei Halbnormen auf $\mathcal{P}^m(-1,1)$ mit $\ker \|(\cdot)'\|_{L^2(-1,1)} \subseteq \ker \|(\cdot)''\|_{L^2(-1,1)}$ sind, folgt mit Lemma 3.4

$$\|p''\|_{L^2(-1,1)} \leq C_m^{-1} \|p'\|_{L^2(-1,1)} \quad \text{für alle } p \in \mathcal{P}^m(-1,1) \quad (3.23)$$

mit einer Konstanten $C_m > 0$ abhängig vom Polynomgrad m . Sei nun $\tilde{p} \in \mathcal{P}^m(\Gamma) \setminus \mathcal{P}^1(\Gamma)$, $m \geq 2$ ein zusammengesetztes Polynom mit $\tilde{p}|_{E_j} \circ \gamma_j = \tilde{p}_j \circ \gamma_j = p_j$. Dann folgt aus (3.22) und (3.23) für die lokalen Anteile

$$\begin{aligned} h \|(\tilde{p}_j - G_\ell)'\|_{L^2(E_j)}^2 &= 2 \|(\tilde{p}_j \circ \gamma_j - G_\ell \circ \gamma_j)'\|_{L^2(-1,1)}^2 \geq 2 C_m \|(\tilde{p}_j \circ \gamma_j - G_\ell \circ \gamma_j)''\|_{L^2(-1,1)}^2 \\ &= \frac{h^3}{4} C_m \|(\tilde{p}_j - G_\ell)''\|_{L^2(E_j)}^2. \end{aligned}$$

Wegen $G_\ell \in \mathcal{S}^1(\mathcal{E}_\ell)$ gilt $G_\ell'' = 0$ und daher auch

$$h \|(\tilde{p}_j - G_\ell)'\|_{L^2(E_j)}^2 \geq h^3 \frac{C_m}{4} \|\tilde{p}_j''\|_{L^2(E_j)}^2. \quad (3.24)$$

Dies impliziert nun weiters

$$\begin{aligned} h^{1/2} \|(\tilde{p} - G_\ell)'\|_{L^2(\Gamma)} &= \left(\sum_{j=1}^N h \|(\tilde{p}_j - G_\ell)'\|_{L^2(E_j)}^2 \right)^{1/2} \geq \left(\sum_{j=1}^N h^3 \frac{C_m}{4} \|\tilde{p}_j''\|_{L^2(E_j)}^2 \right)^{1/2} \\ &= \frac{C_m^{1/2}}{2} \left(\sum_{j=1}^N \|h^{3/2} \tilde{p}_j''\|_{L^2(E_j)}^2 \right)^{1/2} = \frac{C_m^{1/2}}{2} \|h^{3/2} \tilde{p}''\|_{L^2(\Gamma)}. \end{aligned}$$

Da das Polynom $\tilde{p}'' \in \mathcal{P}^{m-2}(\Gamma) \subseteq L^2(\Gamma)$ für $m \geq 2$ beliebig ist, folgt mit $g \in \mathcal{P}^m(\Gamma) \setminus \mathcal{P}^1(\Gamma)$ und $C := C_m^{1/2}/2$

$$\text{osc}_{D,\ell} = h^{1/2} \|(g - G_\ell)'\|_{L^2(\Gamma)} \geq C h^{3/2} \|g''\|_{L^2(\Gamma)}$$

und damit (a).

Für die Behauptung (b) gilt zunächst

$$\text{osc}_{D,\ell}(E_j)^2 = 2 \int_{-1}^1 \left(((g - G_\ell) \circ \gamma_j)'(s) \right)^2 ds \leq 2 \cdot 2 \cdot \|((g - G_\ell) \circ \gamma_j)'\|_{L^\infty(-1,1)}^2.$$

Unter der Voraussetzung $g \in \mathcal{C}^2(\Gamma)$ können wir wegen $G_\ell \circ \gamma_j \in \mathcal{P}^1[-1,1]$ Lemma 3.2 mit $m = 1$ anwenden. Mit $|\gamma_j'| = h/2$ und Berücksichtigung der Kettenregel führt dies zu

$$\text{osc}_{D,\ell}(E_j)^2 \leq 4 \left(C \| (g \circ \gamma_j)'' \|_{L^\infty(-1,1)} 2 \right)^2 = C^2 h^4 \|g'' \circ \gamma_j\|_{L^\infty(-1,1)}^2 = C^2 h^4 \|g''\|_{L^\infty(E_j)}^2.$$

Damit gilt weiters

$$\begin{aligned} \text{osc}_{D,\ell}^2 &= h \|(g - G_\ell)'\|_{L^2(\Gamma)}^2 = \sum_{j=1}^N h \|(g - G_\ell)'\|_{L^2(E_j)}^2 = \sum_{j=1}^N \text{osc}_{D,\ell}(E_j)^2 \\ &\leq C^2 h^4 \sum_{j=1}^N \|g''\|_{L^\infty(E_j)}^2 \leq N C^2 h^4 \|g''\|_{L^\infty(\Gamma)}^2. \end{aligned}$$

3. Symm'sche Integralgleichung

Da wir von uniformer Netzverfeinerung ausgehen, gilt $N = 1/h$. Damit folgt nun mit $C' := C^2 \|g''\|_{L^\infty(\Gamma)}^2$

$$\text{osc}_{D,\ell}^2 \leq C' h^3.$$

Nach Wurzelziehen folgt die zweite Behauptung.

Für die Behauptung (c) gilt mit (3.20) und der umgekehrten Dreiecksungleichung

$$\begin{aligned} |\text{osc}_{D,\ell} - \widetilde{\text{osc}}_{D,\ell}|^2 &= \left| \|h^{1/2}(g - G_\ell)'\|_{L^2(\Gamma)} - \|h^{1/2}(\tilde{p} - G_\ell)'\|_{L^2(\Gamma)} \right|^2 \leq \|h^{1/2}(g - \tilde{p})'\|_{L^2(\Gamma)}^2 \\ &= \sum_{j=1}^N h \| (g - \tilde{p})' \|_{L^2(E_j)}^2. \end{aligned}$$

Analog obigen Überlegungen gilt mit Lemma 3.2 und $m = 2$ unter der Voraussetzung $g \in \mathcal{C}^3(\Gamma)$

$$\begin{aligned} h \| (g - \tilde{p})' \|_{L^2(E_j)}^2 &= 2 \int_{-1}^1 \left(((g - \tilde{p}) \circ \gamma_j)' \right)^2(s) ds \leq 4 \| ((g - \tilde{p}) \circ \gamma_j)' \|_{L^\infty(-1,1)}^2 \\ &\leq C^2 h^6 \|g'''\|_{L^\infty(E_j)}^2. \end{aligned}$$

Oben eingesetzt und mit $N = 1/h$ für uniforme Netze erhalten wir

$$|\text{osc}_{D,\ell} - \widetilde{\text{osc}}_{D,\ell}| \leq \mathcal{O}(h^{5/2})$$

und damit die zweite Behauptung. □

Implementierung in MATLAB

Listing 1: computeOscDirichlet

```

1 function [osc,uDh] = computeOscDirichlet(coordinates,elements,uD,varargin)
2 %*** compute midpoints of all elements
3 midpoints = 0.5*( coordinates(elements(:,1),:) + coordinates(elements(:,2),:) );
4
5 %*** evaluate Dirichlet data at element midpoints
6 uD_midpoints = uD(midpoints,varargin{:});
7
8 %*** determine size of problem
9 dim = size(uD_midpoints,2);
10
11 %*** evaluate Dirichlet data at all Dirichlet nodes
12 uDh = zeros(size(coordinates,1),dim);
13 idx = unique(elements);
14 uDh(idx,:) = uD(coordinates(idx,:),varargin{:});

```

Listing 1 (vgl. [1, Listing 5]) zeigt die Berechnung der Dirichlet-Datenoszillationen in MATLAB:

- Die Funktion übernimmt das Netz $\mathcal{E}_\ell$ in Form von **coordinates** und **elements** und die Funktion **uD** für die Dirichlet-Daten g . In unserem Fall wird daher kein Parameter über **varargin** eingelesen. Die Rückgabewerte der Funktion sind die approximierten Datenoszillationen $\widetilde{\text{osc}}_{D,\ell}$ als **osc** und der Spaltenvektor $\mathbf{g} = \mathbf{uDh}$ als Auswertung der Dirichlet-Daten g an den Knotenpunkten $z \in \mathcal{K}_\ell$ (Zeile 1).
- Zuerst werden alle Mittelpunkte (Zeile 3) berechnet. Anschließend werden die Dirichlet-Daten an den Mittelpunkten (Zeile 6) und an allen Knoten $z \in \mathcal{K}_\ell$ ausgewertet (Zeilen 9–11).
- Zum Schluss wird die Formel (3.21) für alle Elemente $\mathcal{E}_j \in \mathcal{E}_\ell$ angewendet (Zeile 14).
- Die Funktion liefert also einen Spaltenvektor von elementweisen Dirichlet-Datenoszillationen

$$\mathbf{v} := (\widetilde{\text{osc}}_{D,\ell}(E_1)^2, \dots, \widetilde{\text{osc}}_{D,\ell}(E_N)^2) \in \mathbb{R}^N$$

zurück, sodass $\text{osc}_{D,\ell} \approx (\sum_{j=1}^N v_j)^{1/2}$ gilt.

3.4 Konvergenzrate und zuverlässige Schranke des Fehlers $\|\phi - \Phi_\ell\|_V$

In den vorigen Abschnitten haben wir gesehen, dass der Fehler $\|\phi - \phi_\ell\|_V$ der exakten Lösung ϕ der Variationsformulierung (3.8) durch Ersetzen der exakten Lösung ϕ_ℓ der gestörten Variationsformulierung (3.15) durch $\text{osc}_{D,\ell}$ beschränkt ist und beliebig klein wird für kleines h , vgl. (3.16) und Satz 3.3. Für die Praxis ist es jedoch nicht möglich die exakte Lösung ϕ_ℓ der gestörten Variationsformulierung zu berechnen. Vielmehr interessieren wir uns also für den Fehler $\|\phi - \Phi_\ell\|_V$, wobei Φ_ℓ die (berechenbare) Galerkin-Lösung von (3.10) darstellt.

Um nun eine Schranke für $\|\phi - \Phi_\ell\|_V$ zu finden, betrachten wir die nicht-gestörte Variationsformulierung (3.8) mit den exakten Dirichlet-Daten $g \in H^1(\Gamma)$. Die zugehörige Galerkin-Lösung bzgl. der Triangulierung $\mathcal{E}_\ell$ bezeichnen wir mit $\Phi_\ell^* \in \mathcal{P}^0(\mathcal{E}_\ell)$. Analoges vereinbaren wir für die Galerkin-Lösung $\widehat{\Phi}_\ell^* \in \mathcal{P}^0(\widehat{\mathcal{E}}_\ell)$ bzgl. der uniformen Verfeinerung $\widehat{\mathcal{E}}_\ell$ von $\mathcal{E}_\ell$, wobei unter einer uniformen Verfeinerung die Bisektion aller Elemente $E \in \mathcal{E}_\ell$ in zwei Elemente der Länge $\text{length}(E)/2$ verstanden wird. Es gilt also

$$\langle V\Phi_\ell^*, \Psi_\ell \rangle_\Gamma = \langle (K + 1/2)g, \Psi_\ell \rangle_\Gamma \quad \text{für alle } \Psi_\ell \in \mathcal{P}^0(\mathcal{E}_\ell) \quad (3.25)$$

und

$$\langle V\widehat{\Phi}_\ell^*, \widehat{\Psi}_\ell \rangle_\Gamma = \langle (K + 1/2)g, \widehat{\Psi}_\ell \rangle_\Gamma \quad \text{für alle } \widehat{\Psi}_\ell \in \mathcal{P}^0(\widehat{\mathcal{E}}_\ell). \quad (3.26)$$

Dann gilt wegen $(K + 1/2)g = V\phi$ die **Galerkin-Orthogonalität**

$$\langle \phi - \Phi_\ell^*, \Psi_\ell \rangle_V = 0 \quad \text{für alle } \Psi_\ell \in \mathcal{P}^0(\mathcal{E}_\ell) \quad (3.27)$$

als Grundlage für die Fehleranalyse. Mit $\Psi_\ell \in \mathcal{P}^0(\mathcal{E}_\ell)$ beliebig, gilt

$$\|\phi - \Phi_\ell^*\|_V^2 = \langle \phi - \Phi_\ell^*, \phi - \Phi_\ell^* \rangle_V = \langle \phi - \Phi_\ell^*, \phi - \Psi_\ell \rangle_V + \langle \phi - \Phi_\ell^*, \Psi_\ell - \Phi_\ell^* \rangle_V.$$

Da $\Psi_\ell - \Phi_\ell^* \in \mathcal{P}^0(\mathcal{E}_\ell)$ folgt mit (3.27) und Cauchy-Schwarz

$$\|\phi - \Phi_\ell^*\|_V^2 = \langle \phi - \Phi_\ell^*, \phi - \Psi_\ell \rangle_V \leq \|\phi - \Phi_\ell^*\|_V \|\phi - \Psi_\ell\|_V.$$

Da die Abschätzung unabhängig vom gewählten $\Psi_\ell \in \mathcal{P}^0(\mathcal{E}_\ell)$ ist, erhalten wir nach Kürzen die **Bestapproximation** von $\phi \in H^{-1/2}(\Gamma)$ in $\mathcal{P}^0(\mathcal{E}_\ell)$ durch $\Phi_\ell^* \in \mathcal{P}^0(\mathcal{E}_\ell)$ **bzgl. der Energienorm** $\|\cdot\|_V$, d.h. es gilt

$$\inf_{\Psi_\ell \in \mathcal{P}^0(\mathcal{E}_\ell)} \|\phi - \Psi_\ell\|_V = \|\phi - \Phi_\ell^*\|_V. \quad (3.28)$$

Wir führen weiters mit Π_ℓ die **L^2 -Orthogonalprojektion** auf den Raum $\mathcal{P}^0(\mathcal{E}_\ell)$ aller $\mathcal{E}_\ell$ -stückweise konstanten Funktionen, definiert durch

$$(\Pi_\ell v)|_{E_i} := \frac{\langle v, \chi_i \rangle_{L^2(\Gamma)}}{\|\chi_i\|_{L^2(\Gamma)}^2} = \frac{1}{\text{length}(E_i)} \int_{E_i} v \, d\Gamma \quad \text{für alle } E_i \in \mathcal{E}_\ell, \quad (3.29)$$

für $v \in L^1(\Gamma)$ ein. Dann gilt

$$(\Pi_\ell v)(x) = \sum_{i=1}^N (\Pi_\ell v)|_{E_i} \chi_i(x) \in \mathcal{P}^0(\mathcal{E}_\ell)$$

für $x \in \Gamma$.

Im Folgenden nehmen wir stets an, dass die exakte Lösung $\phi \in H^{-1/2}(\Gamma)$ zusätzliche Regularität $\phi \in L^2(\Gamma) \subseteq H^{-1/2}(\Gamma)$ besitzt. Dann gilt die Orthogonalität auch im L^2 -Sinn: Sei $c \in \mathbb{R}$ beliebig und $E_i \in \mathcal{E}_\ell$. Dann folgt mit (3.29) die elementweise Orthogonalität

$$\begin{aligned} \langle \phi - \Pi_\ell \phi, c \rangle_{L^2(E_i)} &= \langle \phi, c \rangle_{L^2(E_i)} - \langle \Pi_\ell \phi, c \rangle_{L^2(E_i)} = c \langle \phi, \chi_i \rangle_{L^2(E_i)} - \frac{\langle \phi, \chi_i \rangle_{L^2(\Gamma)}}{\|\chi_i\|_{L^2(\Gamma)}^2} \int_{E_i} \chi_i c \, d\Gamma \\ &= c \langle \phi, \chi_i \rangle_{L^2(\Gamma)} - c \frac{\langle \phi, \chi_i \rangle_{L^2(\Gamma)}}{\|\chi_i\|_{L^2(\Gamma)}^2} \int_{E_i} \chi_i^2 \, d\Gamma = 0 \end{aligned}$$

und damit, analog obiger Berechnung, für alle $E \in \mathcal{E}_\ell$

$$\inf_{c \in \mathbb{R}} \|\phi - c\|_{L^2(E)} = \|\phi - \Pi_\ell \phi\|_{L^2(E)}.$$

Insgesamt bedeutet dies die **Bestapproximation** von $\phi \in L^2(\Gamma) \subseteq H^{-1/2}(\Gamma)$ in $\mathcal{P}^0(\mathcal{E}_\ell)$ durch $\Pi_\ell \phi \in \mathcal{P}^0(\mathcal{E}_\ell) \subseteq L^2(\Gamma)$ **bzgl. der L^2 -Norm** $\|\cdot\|_{L^2(\Gamma)}$, d.h. es gilt

$$\inf_{\Psi_\ell \in \mathcal{P}^0(\mathcal{E}_\ell)} \|\phi - \Psi_\ell\|_{L^2(\Gamma)} = \|\phi - \Pi_\ell \phi\|_{L^2(\Gamma)}. \quad (3.30)$$

Das folgende Lemma liefert uns einige nützliche Abschätzungen hinsichtlich der L^2 -Orthogonalprojektion, entnommen aus [9, Lemma 3.1 (ii)].

Lemma 3.5 Für die L^2 -Orthogonalprojektion Π_ℓ auf $\mathcal{P}^0(\mathcal{E}_\ell)$ und $v \in L^2(\Gamma)$ gilt

$$\|v - \Pi_\ell v\|_V \leq C \|h_\ell^{1/2} (v - \Pi_\ell v)\|_{L^2(\Gamma)} \leq C \|h_\ell^{1/2} v\|_{L^2(\Gamma)}.$$

Die Konstante $C > 0$ hängt nur vom Rand Γ , und nicht von der Triangulierung $\mathcal{E}_\ell$, ab. $\square$

Damit erhalten wir folgenden

Satz 3.6 Bei hinreichend glatten Neumann-Daten $\phi \in H^{-1/2}(\Gamma)$ und Dirichlet-Daten $g \in H^{1/2}(\Gamma)$ gilt bei uniformer Netzverfeinerung

$$\|\phi - \Phi_\ell\|_V = \mathcal{O}(h^{3/2}).$$

BEWEIS: Da $\Pi_\ell \phi \in \mathcal{P}^0(\mathcal{E}_\ell)$, gilt mit $\Phi_\ell^* \in \mathcal{P}^0(\mathcal{E}_\ell)$ als Bestapproximation von $\phi \in H^{-1/2}(\Gamma)$ bzgl. $\|\cdot\|_V$

$$\|\phi - \Phi_\ell\|_V \leq \|\phi - \Phi_\ell^*\|_V + \|\Phi_\ell^* - \Phi_\ell\|_V \leq \|\phi - \Pi_\ell \phi\|_V + \|\Phi_\ell^* - \Phi_\ell\|_V. \quad (3.31)$$

Weiters gilt mit $\Phi_\ell^* - \Phi_\ell$ als Bestapproximation von $\phi - \phi_\ell$ in $\mathcal{P}^0(\mathcal{E}_\ell)$ bzgl. der Energienorm

$$\begin{aligned} \|\Phi_\ell^* - \Phi_\ell\|_V &\leq \|\phi - \phi_\ell - (\Phi_\ell^* - \Phi_\ell)\|_V - \|\phi - \phi_\ell\|_V \leq \|\phi - \phi_\ell - 0\|_V - \|\phi - \phi_\ell\|_V \\ &\leq 2 \|\phi - \phi_\ell\|_V, \end{aligned}$$

was mit (3.16) zu

$$\|\Phi_\ell^* - \Phi_\ell\|_V \lesssim \text{osc}_{D,\ell} \quad (3.32)$$

führt. Wegen $\phi \in L^2(\Gamma)$, folgt mit Lemma 3.5 und (3.32) aus (3.31)

$$\begin{aligned} \|\phi - \Phi_\ell\|_V &\leq \|\phi - \Pi_\ell \phi\|_V + \text{osc}_{D,\ell} \lesssim \|h^{1/2}(\phi - \Pi_\ell \phi)\|_{L^2(\Gamma)} + \text{osc}_{D,\ell} \\ &= \left(\sum_{j=1}^N \|h^{1/2}(\phi - \Pi_\ell \phi)\|_{L^2(E_j)}^2 \right)^{1/2} + \text{osc}_{D,\ell}. \end{aligned} \quad (3.33)$$

Dann gilt mit der Referenzparametrisierung γ_j und der Definition der L^2 -Orthogonalprojek-

tion (3.29)

$$\begin{aligned}
\|h^{1/2}(\phi - \Pi_\ell \phi)\|_{L^2(E_j)}^2 &= h \int_{-1}^1 (\phi - \Pi_\ell \phi)^2 d\Gamma = \frac{h^2}{2} \int_{-1}^1 \left((\phi - \Pi_\ell \phi) \circ \gamma_j(s) \right)^2 ds \\
&= \frac{h^2}{2} \int_{-1}^1 \left(\phi \circ \gamma_j(s) - \frac{1}{h} \int_{E_j} \phi d\Gamma \right)^2 ds \\
&= \frac{h^2}{2} \int_{-1}^1 \left(\phi \circ \gamma_j(s) - \frac{1}{2} \int_{-1}^1 \phi \circ \gamma_j(t) dt \right)^2 ds \\
&= \frac{h^2}{2} \int_{-1}^1 \left(\frac{1}{2} \int_{-1}^1 \phi \circ \gamma_j(s) dt - \frac{1}{2} \int_{-1}^1 \phi \circ \gamma_j(t) dt \right)^2 ds \\
&= \frac{h^2}{8} \int_{-1}^1 \left(\int_{-1}^1 \phi \circ \gamma_j(s) - \phi \circ \gamma_j(t) dt \right)^2 ds.
\end{aligned}$$

Mit $\phi \circ \gamma_j \in \mathcal{C}^1(-1, 1)$, folgt für alle festen $s, t \in (-1, 1)$ aus dem Mittelwertsatz der Differentialrechnung mit einer Zwischenstelle $\xi \in (-1, 1)$

$$\phi \circ \gamma_j(s) - \phi \circ \gamma_j(t) = (\phi \circ \gamma_j)'(\xi) \cdot (s - t) \leq \|(\phi \circ \gamma_j)'\|_{L^\infty(-1, 1)} \cdot (s - t).$$

Oben eingesetzt folgt mit $|\gamma_j'| = h/2$

$$\begin{aligned}
\|h^{1/2}(\phi - \Pi_\ell \phi)\|_{L^2(E_j)}^2 &\leq \frac{h^2}{8} \|(\phi \circ \gamma_j)'\|_{L^\infty(-1, 1)}^2 \int_{-1}^1 \left(\int_{-1}^1 (s - t) dt \right)^2 ds \\
&= \frac{h^2}{8} \|(\phi \circ \gamma_j)'\|_{L^\infty(-1, 1)}^2 \int_{-1}^1 (2s)^2 ds = \frac{h^2}{3} \|(\phi \circ \gamma_j)'\|_{L^\infty(-1, 1)}^2 \\
&= \frac{h^4}{12} \|\phi'\|_{L^\infty(E_j)}^2.
\end{aligned}$$

Wegen uniformer Netzverfeinerung gilt $N = 1/h$ und infolgedessen

$$\|h^{1/2}(\phi - \Pi_\ell \phi)\|_{L^2(\Gamma)} = \left(\sum_{j=1}^N \|h^{1/2}(\phi - \Pi_\ell \phi)\|_{L^2(E_j)}^2 \right)^{1/2} \lesssim \left(h^4 N \|\phi'\|_{L^\infty(\Gamma)}^2 \right)^{1/2} = \mathcal{O}(h^{3/2}).$$

Mit Satz 3.3 folgt dies auch für den zweiten Term in (3.33) und damit die Behauptung. $\square$

Der Vollständigkeit halber wollen wir an dieser Stelle zusätzlich zur Konvergenzrate auch eine zuverlässige (berechenbare) Schranke des Fehlers $\|\phi - \Phi_\ell\|_V$ als Vorbereitung für das nächste Kapitel angeben. Sie bildet die Voraussetzung für eine effiziente Verfeinerung von Netzelementen im Zuge der adaptiven BEM. Seien dazu wieder Φ_ℓ^* die Galerkin-Lösung bzgl. der nicht-gestörten Variationsformulierung (3.25), d.h.

$$\langle V \Phi_\ell^*, \Psi_\ell \rangle_\Gamma = \langle (K + 1/2) g, \Psi_\ell \rangle_\Gamma \quad \text{für alle } \Psi_\ell \in \mathcal{P}^0(\mathcal{E}_\ell)$$

und $\widehat{\Phi}_\ell^*$ jene bzgl. des uniform verfeinerten Netzes $\widehat{\mathcal{E}}_\ell$ von $\mathcal{E}_\ell$. Weiters sei Φ_ℓ die Galerkin-Lösung der gestörten Variationsformulierung (3.15) d.h. es gilt

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

und $\widehat{\Phi}_\ell^*$ jene bzgl. des uniform verfeinerten Netzes $\widehat{\mathcal{E}}_\ell$ von $\mathcal{E}_\ell$. Wir definieren nun einen für die weiteren Überlegungen unerlässlichen Begriff:

Definition 3.7 (Saturationsannahme). *Unter der Saturationsannahme verstehen wir die Annahme der Gültigkeit von*

$$\|\phi - \widehat{\Phi}_\ell^*\|_V \leq C_{\text{sat}} \|\phi - \Phi_\ell^*\|_V \quad (3.34)$$

mit einer $\mathcal{E}_\ell$ -unabhängigen Konstante $C_{\text{sat}} \in (0, 1)$.

Unter dieser Annahme stellt dann die zur uniformen Netzverfeinerung gehörige Lösung $\widehat{\Phi}_\ell^*$ eine lineare Verbesserung des Approximationsfehlers gegenüber der Lösung Φ_ℓ^* dar. Weiters kann bei n -maliger Verfeinerung, dargestellt durch $\widehat{\cdot}^{(n)}$, auch das Konvergenzverhalten bzgl. der „ursprünglichen“ Lösung Φ_ℓ^* beschrieben werden, d.h. es gilt

$$\|\phi - \widehat{\Phi}_\ell^*\|_V \leq C_{\text{sat}} \|\phi - \widehat{\Phi}_\ell^{(n-1)*}\|_V \leq C_{\text{sat}}^n \|\phi - \Phi_\ell^*\|_V$$

mit $C_{\text{sat}}^n \ll 1$ für großes $n \in \mathbb{N}$. Diese Annahme der Saturation ist sehr intuitiv und kann auch für die Finite Element Methode bewiesen werden, vgl. [1, Remark 5.4]. Für die Randelementmethode ist die Gültigkeit von (3.34) noch offen.

Wir nehmen nun Satz 5.7 aus Kapitel 5 vorweg und erhalten mit dem Fehlerschätzer

$$\tilde{\mu}_\ell = \|h_\ell^{1/2} (\widehat{\Phi}_\ell - \Pi_\ell \widehat{\Phi}_\ell)\|_{L^2(\Gamma)}$$

unter der Saturationsannahme (3.34) als berechenbare Schranke von $\|\phi - \Phi_\ell\|_V$ die Abschätzung

$$\|\phi - \Phi_\ell\|_V \lesssim \tilde{\mu}_\ell + \text{osc}_{D,\ell}. \quad (3.35)$$

4 Adaptive BEM

Um eine effiziente Berechnung hinsichtlich Speicheraufwand und Rechenzeit zu gewährleisten, ist es sinnvoll, nur Bereiche des Randes mit einem dichteren Netz auszustatten, wo die (unbekannte) Lösung singular ist. Jene Elemente $E \in \mathcal{E}_\ell$, auf denen die Galerkin-Lösung Φ_ℓ nicht „hinreichend genau“ ist und daher einen großen Beitrag zum Gesamtfehler $\|\phi - \Phi_\ell\|_V$ leisten, werden mit Hilfe von Indikatoren ermittelt und anschließend in feinere Elemente zerlegt. In diesem Abschnitt stellen wir eine Verfeinerungsstrategie von [2] vor, die in einem gewissen Sinne optimal ist, vgl. Satz 4.2. Die Berechnung dieser Verfeinerungsindikatoren erfolgt dabei auf Grundlage des am Ende des letzten Kapitels vorgestellten Fehlerschätzers $\tilde{\mu}_\ell$. Prinzipiell eignet sich dazu aber jeder a-posteriori Fehlerschätzer, der Auskunft über den lokalen Fehler auf einem Element $E \in \mathcal{E}_\ell$ zulässt. Wir werden im Kapitel 5 noch weitere in Betracht kommende Fehlerschätzer sehen.

Weiters stellen wir mit Algorithmus 4.3 auch einen adaptiven Algorithmus vor, der unter der Saturationsannahme (3.34) eine konvergente Folge Φ_ℓ mit $\|\phi - \Phi_\ell\|_V \rightarrow 0$ generiert, vgl. Satz 4.4.

Wir beginnen mit der **Dörfler Markierung** für lokale Netzverfeinerung, siehe [7].

4.1 Dörfler Markierung für lokale Netzverfeinerung

Wir folgen der Dokumentation von [1, Abschnitt 4.1] und definieren auf allen $E \in \mathcal{E}_\ell$ einen Indikator $\iota_\ell(E)$. Ziel ist es nun für gegebenes $\theta \in (0, 1)$ die minimale Menge $\mathcal{M}_\ell \subseteq \mathcal{E}_\ell$ zu finden, sodass

$$\theta \sum_{E \in \mathcal{E}_\ell} \iota_\ell(E)^2 \leq \sum_{E \in \mathcal{M}_\ell} \iota_\ell(E)^2 \quad (4.1)$$

gilt. Im Zuge der adaptiven BEM wird dann jedes Element $E_i \in \mathcal{M}_\ell$ verfeinert, d.h. es erfolgt die Bisektion des Elements $E_i \in \mathcal{E}_\ell$ in zwei Elemente $e_j, e_k \in \mathcal{E}_{\ell+1}$ mit $E_i = e_j \cup e_k$ halber Länge. Es gilt also $\frac{1}{2}h_\ell|_{E_i} = h_{\ell+1}|_{e_j} = h_{\ell+1}|_{e_k}$.

Klarerweise führen kleine Mengen $\mathcal{M}_\ell$ bei wiederholten Schleifenvorgängen im Zuge des adaptiven Algorithmus zu einer hohen Laufzeit. Mit einem zusätzlichen Parameter $\rho \in (0, 1)$ können wir die minimale Anzahl an Elementen der Menge $\mathcal{M}_\ell$ beeinflussen und suchen nun eine Obermenge $\mathcal{M}_\ell \subseteq \overline{\mathcal{M}}_\ell \subseteq \mathcal{E}_\ell$ mit den Eigenschaften

- (a) $\frac{\#\overline{\mathcal{M}}_\ell}{\#\mathcal{E}_\ell} \geq \rho$
- (b) $\iota_\ell(E) \geq \iota_\ell(E')$ für alle $E \in \overline{\mathcal{M}}_\ell$ und $E' \in \mathcal{E}_\ell \setminus \overline{\mathcal{M}}_\ell$

Die Eigenschaft (a) führt zur Verfeinerung eines fixen Prozentsatzes von Elementen aus $\mathcal{E}_\ell$ und (b) garantiert, dass jene Elemente mit den größten Indikatoren verfeinert werden.

Wir definieren mit

$$\tilde{\mu}_\ell := \|h_\ell^{1/2} (\hat{\Phi}_\ell - \Pi_\ell \hat{\Phi}_\ell)\|_{L^2(\Gamma)} = \left(\sum_{E \in \mathcal{E}_\ell} \|h_\ell^{1/2} (\hat{\Phi}_\ell - \Pi_\ell \hat{\Phi}_\ell)\|_{L^2(E)}^2 \right)^{1/2}$$

und den lokalen Dirichlet-Datenoszillationen $\text{osc}_{D,\ell}(E)$ aus (3.17) den Verfeinerungsindikator $\iota_\ell(E)$ durch

$$\iota_\ell(E)^2 := \|h_\ell^{1/2} (\widehat{\Phi}_\ell - \Pi_\ell \widehat{\Phi}_\ell)\|_{L^2(E)}^2 + \text{osc}_{D,\ell}(E)^2 \quad \text{für alle } E \in \mathcal{E}_\ell.$$

Dann erhalten wir für den Fehlerschätzer ι_ℓ

$$\iota_\ell^2 := \sum_{E \in \mathcal{E}_\ell} \iota_\ell(E)^2 = \tilde{\mu}_\ell^2 + \text{osc}_{D,\ell}^2. \quad (4.2)$$

Beschränktheit der K-Netzkonstante

Lokale Eigenschaften von Netzen spielen in vielen Abschätzungen der numerischen Analysis eine wichtige Rolle. Ein wichtiges Beispiel dafür ist die Beschränktheit der K-Netzkonstante. Sie ist definiert durch

$$\kappa(\mathcal{E}_\ell) := \sup \left\{ \frac{\text{length}(E_j)}{\text{length}(E_k)} : E_j, E_k \in \mathcal{E}_\ell \text{ mit } E_j \cap E_k \neq \emptyset \right\} \geq 1 \quad (4.3)$$

und beschreibt das maximale Verhältnis der Elementgröße zweier benachbarter Elemente. Liefert nun ein Algorithmus mittels obiger Dörfler Markierung die zu verfeinernden Elemente $\mathcal{M}_\ell \subseteq \mathcal{E}_\ell$ zurück, so müssen auch Elemente aus $\mathcal{E}_\ell \setminus \mathcal{M}_\ell$ verfeinert werden, damit $\kappa(\mathcal{E}_\ell)$ für $\ell \rightarrow \infty$ beschränkt bleibt.

Eine Möglichkeit der Realisierung ist der folgende Algorithmus in Anlehnung an [2, Algorithm 2.4]:

Algorithmus 4.1

INPUT: Ausgangsnetz $\mathcal{E}_0$, Triangulierung $\mathcal{E}_\ell$, markierte Elemente $\mathcal{M}_\ell^{(0)} := \mathcal{M}_\ell$, Zähler $i := 0$.

- (1) Definiere $\mathcal{U}^{(i)} := \bigcup_{E \in \mathcal{M}_\ell^{(i)}} \left\{ E' \in \mathcal{E}_\ell \setminus \mathcal{M}_\ell^{(i)} \text{ Nachbar von } E : h_{\ell|E'} > \kappa(\mathcal{E}_0) h_{\ell|E} \right\}$
- (2) Falls $\mathcal{U}^{(i)} \neq \emptyset$, definiere $\mathcal{M}_\ell^{(i+1)} := \mathcal{M}_\ell^{(i)} \cup \mathcal{U}^{(i)}$, erhöhe Zähler $i \mapsto i + 1$ und gehe zu (1).
- (3) Ansonsten verfeinere alle markierten Elemente $E \in \mathcal{M}_\ell^{(i)}$ durch Bisektion um $\mathcal{E}_{\ell+1}$ zu erhalten.

OUTPUT: Verfeinerte Triangulierung $\mathcal{E}_{\ell+1}$.

Dieser Algorithmus ist in einem bestimmten Sinne sogar optimal, vgl. [2, Theorem 2.5]:

Satz 4.2 Sei $\mathcal{E}_0$ ein gegebenes Ausgangsnetz und $\mathcal{E}_\ell$ die durch Algorithmus 4.1 generierten Triangulierungen, wobei die Menge der für die Verfeinerung markierten Elemente $\mathcal{M}_\ell \subseteq \mathcal{E}_\ell$

beliebig ist. Dann gilt Optimalität im Sinne von

$$\#\mathcal{E}_\ell - \#\mathcal{E}_0 \leq C_{\mathcal{E}_0} \sum_{j=0}^{\ell-1} \#\mathcal{M}_j$$

mit einer nur von dem Ausgangsnetz $\mathcal{E}_0$ abhängigen Konstante $C_{\mathcal{E}_0} > 0$. Weiters gilt

$$\kappa(\mathcal{E}_\ell) \leq 2\kappa(\mathcal{E}_0) \tag{4.4}$$

für alle $\ell \in \mathbb{N}$. □

Wegen der Beschränktheit (4.4) der K-Netzkonstante $\kappa(\mathcal{E}_\ell)$ sprechen wir auch von **K-Netzen**.

4.2 Adaptiver Netzverfeinerungs-Algorithmus

Im vorigen Abschnitt haben wir den Verfeinerungsindikator durch

$$\iota_\ell^2 = \tilde{\mu}_\ell^2 + \text{osc}_{D,\ell}^2 = \sum_{j=1}^N \tilde{\mu}_\ell(E_j)^2 + \sum_{j=1}^N \text{osc}_{D,\ell}(E_j)^2$$

mit den lokalen Beiträgen

$$\begin{aligned} \iota_\ell(E_j)^2 &= \tilde{\mu}_\ell(E_j)^2 + \text{osc}_{D,\ell}(E_j)^2 \\ &= \|h_\ell^{1/2}(\hat{\Phi}_\ell - \Pi_\ell \hat{\Phi}_\ell)\|_{L^2(E_j)}^2 + \|h_\ell^{1/2}(g - G_\ell)'\|_{L^2(E_j)}^2 \quad \text{für alle } E_j \in \mathcal{E}_\ell \end{aligned}$$

definiert. Wir kommen nun zu einem adaptiven Algorithmus, der eine Folge von Galerkin-Lösungen Φ_ℓ inklusive den zugehörige Triangulierungen zurück gibt und schreiben in Anlehnung an [2, Algorithm 4.1] den

Algorithmus 4.3

INPUT: Ausgangsnetz $\mathcal{E}_0$, Parameter $\theta \in (0, 1)$, Dirichlet-Daten g , maximale Anzahl an Elementen $N_{\max} \in \mathbb{N}$ und Zähler $\ell := 0$

- (1) Führe uniforme Netzverfeinerung $\hat{\mathcal{E}}_\ell$ von $\mathcal{E}_\ell$ durch.
- (2) Berechne die Galerkin-Lösung $\hat{\Phi}_\ell \in \mathcal{P}^0(\hat{\mathcal{E}}_\ell)$.
- (3) Berechne die Verfeinerungsindikatoren $\iota_\ell(E)$ für alle $E \in \mathcal{E}_\ell$.
- (4) Bestimme (minimale) Menge $\mathcal{M}_\ell \subseteq \mathcal{E}_\ell$, sodass

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

gilt.

- (5) Verfeinere alle Elemente $\mathcal{M}_\ell$ mittels Algorithmus 4.1 zu Triangulierung $\mathcal{E}_{\ell+1}$ mit $\kappa(\mathcal{E}_{\ell+1}) \leq 2\kappa(\mathcal{E}_0)$.

(6) Falls $\#\mathcal{E}_{\ell+1} \leq N_{max}$ erhöhe $\ell \mapsto \ell + 1$ und gehe zu (1).

OUTPUT: Folge von adaptiv generierten Triangulierungen $\hat{\mathcal{E}}_\ell$ mit zugehörigen Galerkin-Lösungen $\hat{\Phi}_\ell \in \mathcal{P}^0(\hat{\mathcal{E}}_\ell)$.

Für den folgenden Satz erinnern wir an die Gültigkeit von (3.35), d.h.

$$\|\phi - \Phi_\ell\|_V \lesssim \tilde{\mu}_\ell + \text{osc}_{D,\ell} \quad \text{mit} \quad \tilde{\mu}_\ell = \|h_\ell^{1/2} (\hat{\Phi}_\ell - \Pi_\ell \hat{\Phi}_\ell)\|_{L^2(\Gamma)}$$

unter der Saturationsannahme (3.34) und an die Definition (4.2)

$$\iota_\ell^2 = \sum_{E \in \mathcal{E}_\ell} \iota_\ell(E)^2 = \tilde{\mu}_\ell^2 + \text{osc}_{D,\ell}^2.$$

Wir zitieren nun Theorem 4.2 aus [2]:

Satz 4.4 Der Algorithmus 4.3 garantiert die Konvergenz des Fehlerschätzers ι_ℓ aus (4.2)

$$\lim_{\ell \rightarrow \infty} \iota_\ell = 0.$$

Weiters folgt unter der Saturationsannahme (3.34) die Konvergenz

$$\lim_{\ell \rightarrow \infty} \|\phi - \Phi_\ell\|_V = 0 = \lim_{\ell \rightarrow \infty} \|\phi - \hat{\Phi}_\ell\|_V$$

und insbesondere die Konvergenz der adaptiven BEM. □

5 Fehlerschätzer

In diesem Kapitel beschäftigen wir uns anhand des Modellproblems (1.6) mit folgenden a-posteriori Fehlerschätzern:

- $(h - h/2)$ -basierte Fehlerschätzer (inkl. Two-Level-Fehlerschätzer)
- gewichteter Residualschätzer
- Faermann-Residualschätzer
- Steinbach-Schätzer

Mit obigen Fehlerschätzern ist es möglich den Fehler $\|\phi - \Phi_\ell\|_V$ zwischen der (unbekannten) exakten Lösung ϕ und der approximierten Lösung Φ_ℓ zu schätzen. Allgemein lassen sich Fehlerschätzer in die Gruppen der **globalen und lokalen Fehlerschätzer** teilen. Die globalen Fehlerschätzer geben dabei nur Auskunft über den Fehler $\|\phi - \Phi_\ell\|_V$ auf dem gesamten Netz $\mathcal{E}_\ell$, während die lokalen Fehlerschätzer (zumindest heuristisch) auch Information über die Ungenauigkeit der Approximation auf einem Element $E \in \mathcal{E}_\ell$ liefern und daher als Verfeinerungsindikatoren für einen adaptiven Netzverfeinerungsalgorithmus infrage kommen. Bei den $(h - h/2)$ -basierten Fehlerschätzern finden wir Fehlerschätzer beider Gruppen zugehörig, während der Two-Level-Fehlerschätzer, der gewichtete Residualschätzer, der Faermann-Schätzer und der Steinbach-Schätzer zu den lokalen Fehlerschätzern zählen.

5.1 Notation

Wir fassen zuerst die in diesem Kapitel häufig verwendeten Notationen zusammen:

- Sei $\mathcal{E}_\ell$ eine Triangulierung mit zugehörigen Elementen $E_i = [a_i, b_i]$, d.h. $\mathcal{E}_\ell = \{E_1, \dots, E_N\}$.
- Die Menge der Knoten zur Triangulierung $\mathcal{E}_\ell$ bezeichnen wir mit $\mathcal{K}_\ell = \{z_1, \dots, z_N\}$.
- Die uniforme Verfeinerung von $\mathcal{E}_\ell$ sei durch $\widehat{\mathcal{E}}_\ell$ gekennzeichnet mit $E_i = e_j \cup e_k$, d.h. es gilt $\widehat{\mathcal{E}}_\ell = \{e_1, \dots, e_{2N}\}$.
- Für die Basis des Raumes $\mathcal{P}^0(\mathcal{E}_\ell)$ der $\mathcal{E}_\ell$ -stückweise konstanten Funktionen schreiben wir $(\chi_j)_{j=1}^N$. Analog bezeichne $(\widehat{\chi}_j)_{j=1}^{2N}$ die Basis von $\mathcal{P}^0(\widehat{\mathcal{E}}_\ell)$.
- Die Basis des Raumes $\mathcal{S}^1(\mathcal{E}_\ell)$ der global stetigen und $\mathcal{E}_\ell$ -stückweise affinen Funktionen sei durch $(\zeta_j)_{j=1}^N$ und die Basis des Raumes $\mathcal{S}^1(\widehat{\mathcal{E}}_\ell)$ auf dem verfeinerten Netz $\widehat{\mathcal{E}}_\ell$ durch $(\widehat{\zeta}_j)_{j=1}^{2N}$ gegeben.
- Die Galerkin-Lösungen von (3.10) in Bezug auf $\mathcal{E}_\ell$ bzw. $\widehat{\mathcal{E}}_\ell$ mit den Koeffizientenvektoren $\mathbf{x} = (x_1, \dots, x_N) \in \mathbb{R}^N$ bzw. $\widehat{\mathbf{x}} = (\widehat{x}_1, \dots, \widehat{x}_{2N}) \in \mathbb{R}^{2N}$ bzgl. der Basis $(\chi_j)_{j=1}^N$ bzw. $(\widehat{\chi}_j)_{j=1}^{2N}$ seien durch $\Phi_\ell \in \mathcal{P}^0(\mathcal{E}_\ell)$ bzw. $\widehat{\Phi}_\ell \in \mathcal{P}^0(\widehat{\mathcal{E}}_\ell)$ definiert, d.h. es gilt

$$\Phi_\ell = \sum_{j=1}^N x_j \chi_j \quad \text{und} \quad \widehat{\Phi}_\ell = \sum_{j=1}^{2N} \widehat{x}_j \widehat{\chi}_j.$$

- Die nodalen Interpolanten von $g \in H^{1/2}(\Gamma)$ in Bezug auf $\mathcal{E}_\ell$ bzw. $\widehat{\mathcal{E}}_\ell$ seien durch $G_\ell \in \mathcal{S}^1(\mathcal{E}_\ell)$ bzw. $\widehat{G}_\ell \in \mathcal{S}^1(\widehat{\mathcal{E}}_\ell)$ bezeichnet. Wir stellen sie bzgl. der Basis $(\zeta_j)_{j=1}^N$ bzw. $(\widehat{\zeta}_j)_{j=1}^{2N}$ mittels

$$G_\ell = \sum_{j=1}^N g_j \zeta_j \quad \text{und} \quad \widehat{G}_\ell = \sum_{j=1}^{2N} \widehat{g}_j \widehat{\zeta}_j$$

dar, wobei g_j und $\widehat{g}_j$ die Auswertungen von den Randdaten $g \in H^{1/2}(\Gamma)$ im jeweiligen Knoten des Netzes $\mathcal{E}_\ell$ bzw. $\widehat{\mathcal{E}}_\ell$ bezeichnen.

5.2 Kanonischer $(h - h/2)$ -Fehlerschätzer η_ℓ bzw. η_ℓ^*

Bevor wir uns konkret mit den $(h - h/2)$ -Fehlerschätzern beschäftigen können, führen wir noch einige Eigenschaften von Fehlerschätzern ein und definieren die wichtigen Begriffe der **Effizienz** und **Zuverlässigkeit** eines Schätzers bzgl. einer Norm $\|\cdot\|$ laut [8, S. 2714]:

Definition 5.1 (Effizienz). Sei ϕ die exakte Lösung des Dirichlet-Randwertproblems (1.6) und $\Phi_\ell \in \mathcal{P}^0(\mathcal{E}_\ell)$ die zur Triangulierung $\mathcal{E}_\ell$ gehörige Galerkin-Lösung. Dann heißt ein Fehlerschätzer η_ℓ effizient, falls es eine Konstante $C_{\text{eff}} > 0$ gibt, sodass

$$\eta_\ell \leq C_{\text{eff}} \|\phi - \Phi_\ell\| \tag{5.1}$$

gilt. Die Konstante C_{eff} darf dabei nicht von ϕ oder Φ_ℓ , sondern lediglich von der K -Netzkonstante $\kappa(\mathcal{E}_\ell)$, siehe (4.3), abhängen.

Die Effizienz eines Fehlerschätzers garantiert somit bei Konvergenz von $\|\phi - \Phi_\ell\| \rightarrow 0$ auch die Konvergenz des Fehlerschätzers gegen Null. Damit bildet ein lokaler a-posteriori Fehlerschätzer auch eine untere lokale Schranke des Fehlers $\|\phi - \Phi_\ell\|$ (bis auf eine Konstante). Umgekehrt ist auch die Konvergenz des Approximationsfehlers $\|\phi - \Phi_\ell\| \rightarrow 0$ bei Konvergenz des Fehlerschätzers wünschenswert. Diese Eigenschaft bezeichnen wir als die Zuverlässigkeit eines Schätzers.

Definition 5.2 (Zuverlässigkeit). Bezeichne ϕ die exakte Lösung des Dirichlet-Randwertproblems (1.6) und $\Phi_\ell \in \mathcal{P}^0(\mathcal{E}_\ell)$ die zur Triangulierung $\mathcal{E}_\ell$ gehörige Galerkin-Lösung. Dann heißt ein Fehlerschätzer η_ℓ zuverlässig, falls es eine Konstante $C_{\text{rel}} > 0$ gibt, sodass

$$\|\phi - \Phi_\ell\| \leq C_{\text{rel}} \eta_\ell \tag{5.2}$$

gilt. Die Konstante C_{rel} darf dabei nicht von ϕ oder Φ_ℓ , sondern lediglich von der K -Netzkonstante $\kappa(\mathcal{E}_\ell)$ abhängen.

Ein „guter“ Fehlerschätzer erfüllt somit die Ungleichungskette

$$C_{\text{eff}}^{-1} \eta_\ell \leq \|\phi - \Phi_\ell\| \leq C_{\text{rel}} \eta_\ell.$$

Mit den Galerkin-Lösungen Φ_ℓ^* und $\widehat{\Phi}_\ell^*$ der Variationsformulierung (3.25) bzw. (3.26) mit exakten Randdaten g , können wir den Fehlerschätzer η_ℓ^* durch

$$\eta_\ell^* := \|\widehat{\Phi}_\ell^* - \Phi_\ell^*\|_V \quad (5.3)$$

definieren.

Wir zeigen zuerst, dass der Fehlerschätzer (5.3) effizient und, unter der Saturationsannahme (3.34), auch zuverlässig ist. Mit dieser Kenntnis wird es uns möglich sein, die Effizienz und Zuverlässigkeit des Fehlerschätzers

$$\eta_\ell := \|\widehat{\Phi}_\ell - \Phi_\ell\|_V$$

bis auf Dirichlet-Datenoszillationen $\text{osc}_{D,\ell}$ für die Galerkin-Lösungen des gestörten Variationsproblems (3.10) zu zeigen.

Mit Hilfe der Galerkin-Orthogonalität (3.27) bzgl. der Energienorm gilt immer

$$\|\phi - \Phi_\ell^*\|_V^2 = \|\phi - \widehat{\Phi}_\ell^*\|_V^2 + \|\widehat{\Phi}_\ell^* - \Phi_\ell^*\|_V^2 = \|\phi - \widehat{\Phi}_\ell^*\|_V^2 + \eta_\ell^{*2} \quad (5.4)$$

Damit haben wir zum einen gezeigt, dass $\eta_\ell^* \leq \|\phi - \Phi_\ell^*\|_V$ und damit effizient mit $C_{\text{eff}} = 1$ ist und zum anderen auch

$$\|\phi - \widehat{\Phi}_\ell^*\|_V \leq \|\phi - \Phi_\ell^*\|_V. \quad (5.5)$$

Damit stellt die Saturationsannahme (3.34) eine Verschärfung von (5.5) dar. Wir kommen zu folgendem Satz, vgl. [8, Proposition 3.1].

Satz 5.3 Für den $(h - h/2)$ -Fehlerschätzer $\eta_\ell^* = \|\widehat{\Phi}_\ell^* - \Phi_\ell^*\|_V$ gilt:

- (a) Der $(h - h/2)$ -Fehlerschätzer η_ℓ^* ist immer effizient mit der Konstanten $C_{\text{eff}} = 1$.
- (b) Die Saturationsannahme (3.34) mit der $\mathcal{E}_\ell$ -unabhängigen Konstante $C_{\text{sat}} \in (0, 1)$ ist äquivalent zur Zuverlässigkeit von η_ℓ^* mit der Konstanten $C_{\text{rel}} = (1 - C_{\text{sat}}^2)^{-1/2}$.

BEWEIS: Teil (a) haben wir bereits gezeigt. Mit (5.4) folgt

$$\eta_\ell^{*2} = \|\phi - \Phi_\ell^*\|_V^2 - \|\phi - \widehat{\Phi}_\ell^*\|_V^2$$

und deshalb unter der Annahme der Zuverlässigkeit mit $C_{\text{rel}} = (1 - C_{\text{sat}}^2)^{-1/2}$

$$\begin{aligned} \|\phi - \Phi_\ell^*\|_V^2 &\leq (1 - C_{\text{sat}}^2)^{-1} \eta_\ell^{*2} = (1 - C_{\text{sat}}^2)^{-1} (\|\phi - \Phi_\ell^*\|_V^2 - \|\phi - \widehat{\Phi}_\ell^*\|_V^2) \\ \Leftrightarrow (1 - C_{\text{sat}}^2) \|\phi - \Phi_\ell^*\|_V^2 &\leq \|\phi - \Phi_\ell^*\|_V^2 - \|\phi - \widehat{\Phi}_\ell^*\|_V^2 \\ \Leftrightarrow \|\phi - \widehat{\Phi}_\ell^*\|_V^2 &\leq C_{\text{sat}}^2 \|\phi - \Phi_\ell^*\|_V^2 \end{aligned}$$

und damit die Äquivalenz in (b). □

Wir führen nun eine Erweiterung der Begriffe „Effizienz“ und „Zuverlässigkeit“ laut [2, S. 2] ein und werden nun zeigen, dass der Fehlerschätzer η_ℓ effizient im Sinne von

$$C_{\text{eff}}^{-1} \eta_\ell \leq \|\phi - \Phi_\ell\|_V + \text{osc}_{D,\ell} \quad (5.6)$$

und unter der Saturationsannahme (3.34) zuverlässig im Sinne von

$$C_{\text{rel}}^{-1} \|\phi - \Phi_\ell\|_V \leq \eta_\ell + \text{osc}_{D,\ell} \quad (5.7)$$

ist. D.h. wir zeigen die **Effizienz und Zuverlässigkeit** des Fehlerschätzers η_ℓ **bis auf Datenoszillationen**. Erfüllt nun ein Fehlerschätzer diese Eigenschaften, so folgt für dessen Konvergenz aus Satz 3.6 sofort $\mathcal{O}(h^{3/2})$, falls die Lösung ϕ und die Daten g hinreichend glatt sind.

Satz 5.4 *Der $(h - h/2)$ -Fehlerschätzer $\eta_\ell = \|\hat{\Phi}_\ell - \Phi_\ell\|_V$ ist effizient im Sinne von (5.6) und unter der Saturationsannahme (3.34) zuverlässig im Sinne von (5.7).*

BEWEIS: Für den Beweis gehen wir wie in [2, Theorem 3.4] vor. Mit Ungleichung (3.32) gilt

$$\|\Phi_\ell^* - \Phi_\ell\|_V \lesssim \|\phi - \phi_\ell\|_V \lesssim \text{osc}_{D,\ell}.$$

Analog beweist man auch $\|\hat{\Phi}_\ell^* - \hat{\Phi}_\ell\|_V \lesssim \|\phi - \phi_\ell\|_V \lesssim \text{osc}_{D,\ell}$. Wir zeigen zuerst die Effizienz von η_ℓ im Sinne von (5.6): Mit Hilfe der Dreiecksungleichung, der Definition (5.3) und Ungleichung (3.32) gilt

$$\eta_\ell = \|\hat{\Phi}_\ell - \Phi_\ell\|_V \leq \|\hat{\Phi}_\ell^* - \hat{\Phi}_\ell\|_V + \|\hat{\Phi}_\ell^* - \Phi_\ell^*\|_V + \|\Phi_\ell^* - \Phi_\ell\|_V \lesssim \eta_\ell^* + \text{osc}_{D,\ell}. \quad (5.8)$$

Da η_ℓ^* effizient mit $C_{\text{eff}} = 1$ ist, d.h. $\eta_\ell^* \leq \|\phi - \Phi_\ell^*\|_V$, folgt mit der Bestapproximation bzgl. der Energienorm (3.28)

$$\eta_\ell \lesssim \|\phi - \Phi_\ell^*\|_V + \text{osc}_{D,\ell} \leq \|\phi - \Phi_\ell\|_V + \text{osc}_{D,\ell}$$

und damit die Effizienz im Sinne von (5.6).

Für den Beweis der Zuverlässigkeit von η_ℓ verwenden wir die Zuverlässigkeit von η_ℓ^* und erhalten mit Gleichung (3.32)

$$\|\phi - \Phi_\ell\|_V \leq \|\phi - \Phi_\ell^*\|_V + \|\Phi_\ell^* - \Phi_\ell\|_V \lesssim \eta_\ell^* + \|\Phi_\ell^* - \Phi_\ell\|_V \lesssim \eta_\ell^* + \text{osc}_{D,\ell}$$

Nun brauchen wir die Ungleichung

$$\eta_\ell^* \lesssim \eta_\ell + \text{osc}_{D,\ell},$$

welche man auf gleichem Wege wie (5.8) erhält. Damit gelangen wir schlussendlich zu

$$\|\phi - \Phi_\ell\|_V \lesssim \eta_\ell + \text{osc}_{D,\ell},$$

was den Beweis beschließt. □

5.3 Weitere $(h - h/2)$ -Fehlerschätzer $\tilde{\eta}_\ell$, μ_ℓ , $\tilde{\mu}_\ell$

Im vorigen Abschnitt haben wir die Effizienz und unter der Saturationsannahme die Zuverlässigkeit des Fehlerschätzers η_ℓ gezeigt. Wir werden sehen, dass daraus die Effizienz und Zuverlässigkeit der Schätzer

$$\begin{aligned}\tilde{\eta}_\ell &:= \|\hat{\Phi}_\ell - \Pi_\ell \hat{\Phi}_\ell\|_V \\ \mu_\ell &:= \|h_\ell^{1/2}(\hat{\Phi}_\ell - \Phi_\ell)\|_{L^2(\Gamma)} \\ \tilde{\mu}_\ell &:= \|h_\ell^{1/2}(\hat{\Phi}_\ell - \Pi_\ell \hat{\Phi}_\ell)\|_{L^2(\Gamma)}\end{aligned}$$

folgt, wobei Π_ℓ die L^2 -Orthogonalprojektion aus (3.29) bezeichnet. Bevor wir auf die **Äquivalenz der Fehlerschätzer** eingehen, motivieren wir noch die oben angeführten Schätzer und formulieren in Anlehnung an [1, Abschnitt 5.6]:

Die Definition des Fehlerschätzers $\eta_\ell = \|\hat{\Phi}_\ell - \Phi_\ell\|_V$ birgt zwei Nachteile. Bei bekannter (genauerer) Galerkin-Lösung $\hat{\Phi}_\ell$ ist die Galerkin-Lösung Φ_ℓ wegen (5.5) uninteressant und deren Berechnung inklusive Speicherung würde nur zu unnötigem Aufwenden von Ressourcen führen. Man versucht nun, die für die Berechnung des Fehlerschätzers nötige Lösung Φ_ℓ durch eine möglichst einfache Rechnung zu approximieren. Aufgrund der Bestapproximation von $\hat{\Phi}_\ell \in \mathcal{P}^0(\hat{\mathcal{E}}_\ell) \subseteq H^{-1/2}(\Gamma)$ durch Φ_ℓ in $\mathcal{P}^0(\mathcal{E}_\ell)$ bzgl. der Energienorm, gilt mit Hilfe der L^2 -Orthogonalprojektion (3.29) $\Pi_\ell \hat{\Phi}_\ell \in \mathcal{P}^0(\mathcal{E}_\ell)$ und weiters

$$\eta_\ell = \|\hat{\Phi}_\ell - \Phi_\ell\|_V \leq \|\hat{\Phi}_\ell - \Pi_\ell \hat{\Phi}_\ell\|_V = \tilde{\eta}_\ell.$$

Für die Berechnung des Fehlerschätzers $\tilde{\eta}_\ell$ braucht dann nur mehr die Galerkin-Lösung $\hat{\Phi}_\ell$ gespeichert werden. Ein weiterer Nachteil von η_ℓ liegt darin, dass er keine Information über den lokalen Approximationsfehler auf einem Element $E \in \mathcal{E}_\ell$ liefert. Dies ist jedoch essentiell für einen adaptiven BEM-Algorithmus. Wenn wir die Energienorm durch die mit h_ℓ gewichtete L^2 -Norm ersetzen, gelangen wir so zum Fehlerschätzer μ_ℓ und weiters, mit der oben argumentierten L^2 -Orthogonalprojektion, zur Definition des Schätzers $\tilde{\mu}_\ell$. Diesen Schätzer haben wir auch für den Verfeinerungsindikator ι_ℓ des adaptiven Algorithmus in Kapitel 4 verwendet, vgl. Formel (4.2).

Insgesamt erhalten wir die **globalen Fehlerschätzer** η_ℓ , $\tilde{\eta}_\ell$ und die **lokalen Fehlerschätzer** μ_ℓ , $\tilde{\mu}_\ell$ mit den lokalen Fehleranteilen

$$\begin{aligned}\mu_\ell(E_j) &:= \text{length}(E_j)^{1/2} \|\hat{\Phi}_\ell - \Phi_\ell\|_{L^2(E_j)} \quad \text{bzw.} \\ \tilde{\mu}_\ell(E_j) &:= \text{length}(E_j)^{1/2} \|\hat{\Phi}_\ell - \Pi_\ell \hat{\Phi}_\ell\|_{L^2(E_j)} \quad \text{für alle } E_j \in \mathcal{E}_\ell.\end{aligned}\tag{5.9}$$

Mit der Definition (5.9) gilt dann

$$\mu_\ell = \|h_\ell^{1/2}(\hat{\Phi}_\ell - \Phi_\ell)\|_{L^2(\Gamma)} = \left(\sum_{j=1}^N \text{length}(E_j) \|\hat{\Phi}_\ell - \Phi_\ell\|_{L^2(E_j)}^2 \right)^{1/2} = \left(\sum_{j=1}^N \mu_\ell(E_j)^2 \right)^{1/2}$$

und analog

$$\tilde{\mu}_\ell = \left(\sum_{j=1}^N \tilde{\mu}_\ell(E_j)^2 \right)^{1/2}.$$

Wir kommen nun zur Äquivalenz von Fehlerschätzern:

Definition 5.5 Zwei Fehlerschätzer η und μ heißen äquivalent, wenn es zwei Konstanten $C_1, C_2 > 0$ gibt, sodass

$$C_1^{-1} \mu \leq \eta \leq C_2 \mu$$

gilt. Die Konstanten C_1 und C_2 dürfen dabei nur von der K -Netzkonstante $\kappa(\mathcal{E}_\ell)$, siehe (4.3), abhängen.

Wir entnehmen nun Theorem 3.6 aus [10] und schreiben folgendes

Lemma 5.6 Für jede Funktion $v \in \mathcal{P}^m(\mathcal{E}_\ell)$, $m \in \mathbb{N}_0$, gilt

$$\|h_\ell^{1/2} v\|_{L^2(\Gamma)} \leq C \|v\|_V,$$

wobei die Konstante $C > 0$ nur vom Rand Γ und der K -Netzkonstante $\kappa(\mathcal{E}_\ell)$ abhängt. □

Mit diesem Lemma können wir folgenden Satz beweisen, vgl. [9, Theorem 3.2 und 3.4].

Satz 5.7 Für die Fehlerschätzer η_ℓ , $\tilde{\eta}_\ell$, μ_ℓ und $\tilde{\mu}_\ell$ gilt

$$\eta_\ell \leq \tilde{\eta}_\ell \leq C_1 \tilde{\mu}_\ell \leq \mu_\ell \leq C_2 \eta_\ell$$

mit den Konstanten $C_1 > 0$ aus Lemma 3.5 und $C_2 > 0$ aus Lemma 5.6. Somit sind alle Fehlerschätzer äquivalent. Insbesondere sind daher alle Fehlerschätzer effizient bis auf Datenoszillationen und unter der Saturationsannahme (3.34) zuverlässig bis auf Datenoszillationen.

BEWEIS: Es gilt aufgrund der Bestapproximation von $\hat{\Phi}_\ell \in H^{-1/2}(\Gamma)$ durch Φ_ℓ in $\mathcal{P}^0(\mathcal{E}_\ell)$ bzgl. der Energienorm

$$\eta_\ell = \|\hat{\Phi}_\ell - \Phi_\ell\|_V \leq \|\hat{\Phi}_\ell - \Pi_\ell \Phi_\ell\|_V = \tilde{\eta}_\ell.$$

Weiters gilt wegen $\hat{\Phi}_\ell \in \mathcal{P}^0(\hat{\mathcal{E}}_\ell) \subseteq L^2(\Gamma)$ mit Hilfe von Lemma 3.5

$$\tilde{\eta}_\ell = \|\hat{\Phi}_\ell - \Pi_\ell \Phi_\ell\|_V \leq C_1 \|h_\ell^{1/2}(\hat{\Phi}_\ell - \Pi_\ell \hat{\Phi}_\ell)\|_{L^2(\Gamma)} = \tilde{\mu}_\ell.$$

Da nun $\Pi_\ell \hat{\Phi}_\ell$ die $\mathcal{E}_\ell$ -elementweise Bestapproximation von $\hat{\Phi}_\ell$ bzgl. der L^2 -Norm ist, folgt

$$\tilde{\mu}_\ell = \|h_\ell^{1/2}(\hat{\Phi}_\ell - \Pi_\ell \hat{\Phi}_\ell)\|_{L^2(\Gamma)} \leq \|h_\ell^{1/2}(\hat{\Phi}_\ell - \Phi_\ell)\|_{L^2(\Gamma)} = \mu_\ell.$$

Schlussendlich gilt wegen $\hat{\Phi}_\ell - \Phi_\ell \in \mathcal{P}^0(\hat{\mathcal{E}}_\ell)$ und Lemma 5.6

$$\mu_\ell = \|h_\ell^{1/2}(\hat{\Phi}_\ell - \Phi_\ell)\|_{L^2(\Gamma)} \leq C_2 \|\hat{\Phi}_\ell - \Phi_\ell\|_V = \tilde{\eta}_\ell.$$

Da nun der Fehlerschätzer η_ℓ wegen Satz 5.4 effizient und zuverlässig bis auf Datenoszillationen ist, folgen diese Eigenschaften wegen der ebenen bewiesenen Äquivalenz auch für alle anderen Fehlerschätzer. $\square$

Mit Satz 5.7 folgt nun unter der Saturationsannahme (3.34) unmittelbar

$$\|\phi - \Phi_\ell\|_V \lesssim \eta_\ell + \text{osc}_{D,\ell} \lesssim \tilde{\mu}_\ell + \text{osc}_{D,\ell},$$

womit wir nun auch die Gleichung (3.35) am Ende des Kapitels 3 bewiesen haben.

5.4 Implementierung der globalen $(h - h/2)$ -Fehlerschätzer $\eta_\ell, \tilde{\eta}_\ell$ in MATLAB

5.4.1 Fehlerschätzer η_ℓ

Listing 2: computeEstSlpEta

```

1 function est = computeEstSlpEta(father2son,V_fine,x_fine,x_coarse)
2 *** compute coefficient vector of (phi_fine - phi_coarse) w.r.t. to fine mesh
3 x_fine(father2son(:,1),:) = x_fine(father2son(:,1),:) - x_coarse;
4 x_fine(father2son(:,2),:) = x_fine(father2son(:,2),:) - x_coarse;
5
6 *** compute energy ||| phi_fine - phi_coarse |||^2
7 est = x_fine(:)'*(V_fine*x_fine(:));

```

In diesem Abschnitt berechnen wir den Fehlerschätzer

$$\eta_\ell = \|\hat{\Phi}_\ell - \Phi_\ell\|_V.$$

Wir gehen von den Galerkin-Lösungen

$$\Phi_\ell = \sum_{j=1}^N x_j \chi_j \in \mathcal{P}^0(\mathcal{E}_\ell) \quad \text{und} \quad \hat{\Phi}_\ell = \sum_{j=1}^{2N} \hat{x}_j \hat{\chi}_j \in \mathcal{P}^0(\hat{\mathcal{E}}_\ell)$$

aus. Da $\mathcal{P}^0(\mathcal{E}_\ell) \subseteq \mathcal{P}^0(\hat{\mathcal{E}}_\ell)$ ein Unterraum ist, existiert ein eindeutiger Vektor $\hat{\mathbf{y}} \in \mathbb{R}^{2N}$, sodass

$$\Phi_\ell = \sum_{j=1}^{2N} \hat{y}_j \hat{\chi}_j$$

gilt. Mit $\hat{\mathbf{x}}, \hat{\mathbf{y}} \in \mathbb{R}^{2N}$ und $\hat{\mathbf{V}}$ als Einfachschichtpotential-Matrix (3.11) bezogen auf $\hat{\mathcal{E}}_\ell$ gilt

$$\begin{aligned} \eta_\ell^2 &= \|\hat{\Phi}_\ell - \Phi_\ell\|_V^2 = \langle \hat{\Phi}_\ell - \Phi_\ell, \hat{\Phi}_\ell - \Phi_\ell \rangle_V = \sum_{j,k=1}^{2N} (\hat{x}_j - \hat{y}_j)(\hat{x}_k - \hat{y}_k) \langle \hat{\chi}_j, \hat{\chi}_k \rangle_V \\ &= (\hat{\mathbf{x}} - \hat{\mathbf{y}})^T \hat{\mathbf{V}} (\hat{\mathbf{x}} - \hat{\mathbf{y}}). \end{aligned}$$

Listing 2 (vgl. [1, Listing 9]) zeigt die Implementierung des Schätzers:

- Die Funktion übernimmt die Koeffizienten $\mathbf{x} \in \mathbb{R}^N$ bzw. $\hat{\mathbf{x}} \in \mathbb{R}^{2N}$ von den Galerkin-Lösungen Φ_ℓ und $\hat{\Phi}_\ell$ in Form von $\mathbf{x}$ bzw. $\mathbf{x_fine}$. Weiters wird auch die Einfachschichtpotential-Matrix $\hat{\mathbf{V}}$ bzgl. dem uniform verfeinerten Netz $\hat{\mathcal{E}}_\ell$ als $\mathbf{v_fine}$ und das $(N \times 2)$ -Array **father2son** übergeben. Das Array **father2son** verlinkt dabei 2 Elemente $e_j, e_k \in \hat{\mathcal{E}}_\ell$ mit $E_i = e_j \cup e_k \in \mathcal{E}_\ell$ mittels **father2son**($i, :$) = [j, k]. Demnach gilt auch $\hat{y}_j = \hat{y}_k = x_i$. Als Output erhalten wir den quadrierten globalen Fehlerschätzer η_ℓ^2 (Zeile 1).
- Anschließend wird der Vektor $\hat{\mathbf{x}}$ durch den Koeffizientenvektor $\hat{\mathbf{x}} - \hat{\mathbf{y}}$ von $\hat{\Phi}_\ell - \Phi_\ell$ ersetzt (Zeile 3 f.).
- Zum Schluss wird die Energienorm $\eta_\ell^2 = \|\hat{\Phi}_\ell - \Phi_\ell\|_V^2$ berechnet und zurückgegeben (Zeile 7).

5.4.2 Fehlerschätzer $\tilde{\eta}_\ell$

Listing 3: computeEstSlpEtaTilde

```

1 function est = computeEstSlpEtaTilde(father2son,v_fine,x_fine)
2 %*** compute L2-projection Pi_coarse*phi_fine onto coarse mesh
3 pi_x_fine = 0.5*( x_fine(father2son(:,1),:) + x_fine(father2son(:,2),:) );
4
5 %*** compute coefficient vector of (1-Pi_coarse)*phi_fine
6 x_fine(father2son(:,1),:) = x_fine(father2son(:,1),:) - pi_x_fine;
7 x_fine(father2son(:,2),:) = x_fine(father2son(:,2),:) - pi_x_fine;
8
9 %*** compute energy || (1-Pi_coarse)*phi_fine |||^2
10 est = x_fine(:)'*(v_fine*x_fine(:));

```

Für die Berechnung des Fehlerschätzers

$$\tilde{\eta}_\ell = \|\hat{\Phi}_\ell - \Pi_\ell \hat{\Phi}_\ell\|_V$$

benötigen wir die L^2 -Orthogonalprojektion $\Pi_\ell \hat{\Phi}_\ell$ aus (3.29). Mit

$$\hat{\Phi}_\ell = \sum_{j=1}^{2N} \hat{x}_j \hat{\chi}_j$$

und den Elementen $e_j, e_k \in \hat{\mathcal{E}}_\ell$ als den Söhnen von dem Element $E_i \in \mathcal{E}_\ell$, d.h. $E_i = e_j \cup e_k$ und $\text{length}(e_j) = \text{length}(e_k) = \text{length}(E_i)/2$, erhalten wir

$$\begin{aligned}
(\Pi_\ell \hat{\Phi}_\ell)|_{E_i} &= \frac{1}{\text{length}(E_i)} \int_{E_i} \hat{\Phi}_\ell d\Gamma = \frac{1}{\text{length}(E_i)} \left(\int_{e_j} \hat{\Phi}_\ell d\Gamma + \int_{e_k} \hat{\Phi}_\ell d\Gamma \right) \\
&= \frac{1}{\text{length}(E_i)} (\text{length}(e_j) \hat{x}_j + \text{length}(e_k) \hat{x}_k) = \frac{\hat{x}_j + \hat{x}_k}{2}
\end{aligned} \tag{5.10}$$

für alle $E_i = e_j \cup e_k \in \mathcal{E}_\ell$ mit $e_j, e_k \in \hat{\mathcal{E}}_\ell$. Da $\Pi_\ell \hat{\Phi}_\ell \in \mathcal{P}^0(\mathcal{E}_\ell)$, erhalten wir bzgl. dem feineren Netz $\hat{\mathcal{E}}_\ell$

$$\Pi_\ell \hat{\Phi}_\ell = \sum_{n=1}^{2N} \hat{y}_n \hat{\chi}_n \quad \text{mit } \hat{y}_j = \hat{y}_k = \frac{\hat{x}_j + \hat{x}_k}{2},$$

wobei $e_j, e_k \in \hat{\mathcal{E}}_\ell$ wieder die Söhne von $E_i \in \mathcal{E}_\ell$ darstellen. Analog der Berechnung des Fehlerschätzers η_ℓ erhalten wir schlussendlich

$$\tilde{\eta}_\ell^2 = \|\hat{\Phi}_\ell - \Pi_\ell \hat{\Phi}_\ell\|_V^2 = (\hat{\mathbf{x}} - \hat{\mathbf{y}})^T \hat{\mathbf{V}} (\hat{\mathbf{x}} - \hat{\mathbf{y}}).$$

Wir kommen zur Dokumentation von Listing 3 (vgl. [1, Listing 10]):

- Die Funktion übernimmt, jeweils bzgl. dem Netz $\hat{\mathcal{E}}_\ell$, die Einfachschichtpotential-Matrix $\hat{\mathbf{V}}$ in Form der Matrix `v_fine` und den Koeffizientenvektor $\hat{\mathbf{x}} \in \mathbb{R}^{2N}$ von $\hat{\Phi}_\ell$ als Spaltenvektor `x_fine`. Wieder stellt die Funktion `father2son` die Beziehung zwischen $E_i \in \mathcal{E}_\ell$ und den Söhnen $e_j, e_k \in \hat{\mathcal{E}}_\ell$ her. Als Output erhalten wir die quadrierten globalen Fehlerschätzer $\tilde{\eta}_\ell^2$ (Zeile 1).
- Zuerst wird die L^2 -Orthogonalprojektion $\Pi_\ell \hat{\Phi}_\ell$ bzgl. dem gröberen Netz $\mathcal{E}_\ell$ berechnet, vgl. Formel (5.10) (Zeile 3).
- Anschließend wird $\hat{\mathbf{x}}$ mit dem Koeffizientenvektor $\hat{\mathbf{x}} - \hat{\mathbf{y}} \in \mathbb{R}^{2N}$ von $\hat{\Phi}_\ell - \Pi_\ell \hat{\Phi}_\ell$ überschrieben (Zeile 6 f.).
- Zum Schluss gibt die Funktion $\tilde{\eta}_\ell^2 = \|\hat{\Phi}_\ell - \Pi_\ell \hat{\Phi}_\ell\|_V^2$ zurück (Zeile 10).

5.5 Implementierung der lokalen $(h - h/2)$ -Fehlerschätzer $\mu_\ell, \tilde{\mu}_\ell$ in MATLAB

5.5.1 Fehlerschätzer μ_ℓ

Listing 4: `computeEstSlpMu`

```
1 function ind = computeEstSlpMu(coordinates,elements,father2son,x_fine,x_coarse)
2 %*** compute (squared) local mesh-size
3 h = sum( (coordinates(elements(:,1),:)-coordinates(elements(:,2),:)).^2 ,2);
4
5 %*** compute coefficient vector of (phi_fine - phi_coarse) w.r.t. to fine mesh
6 x_fine(father2son(:,1),:) = x_fine(father2son(:,1),:) - x_coarse;
7 x_fine(father2son(:,2),:) = x_fine(father2son(:,2),:) - x_coarse;
8
9 %*** compute ind(j) = diam(Ej)*|| phi_fine - phi_coarse ||_{L2(Ej)}^2
10 ind = 0.5*h.*sum((x_fine(father2son(:,1),:).^2 + x_fine(father2son(:,2),:).^2),2);
```

In diesem Abschnitt berechnen wir den Fehlerschätzer

$$\mu_\ell = \left(\sum_{i=1}^N \mu_\ell(E_i)^2 \right)^{1/2}$$

mit $\mu_\ell(E_i)^2 = \text{length}(E_i) \|\hat{\Phi}_\ell - \Phi_\ell\|_{L^2(E_i)}^2$. Für festes $E_i \in \mathcal{E}_\ell$ und dessen Söhne $e_j, e_k \in \hat{\mathcal{E}}_\ell$ erhalten wir mit den Bezeichnungen $\hat{\mathbf{x}}, \hat{\mathbf{y}} \in \mathbb{R}^{2N}$ und

$$\hat{\Phi}_\ell = \sum_{j=1}^{2N} \hat{x}_j \hat{\chi}_j \quad \text{bzw.} \quad \Phi_\ell = \sum_{j=1}^{2N} \hat{y}_j \hat{\chi}_j,$$

vgl. Abschnitt 5.4.2, die Beziehung

$$\|\hat{\Phi}_\ell - \Phi_\ell\|_{L^2(E_i)}^2 = \int_{e_j} (\hat{\Phi}_\ell - \Phi_\ell)^2 d\Gamma + \int_{e_k} (\hat{\Phi}_\ell - \Phi_\ell)^2 d\Gamma = \frac{\text{length}(E_i)}{2} ((\hat{x}_j - \hat{y}_j)^2 + (\hat{x}_k - \hat{y}_k)^2).$$

Somit folgt für die Berechnung der lokalen Fehleranteile

$$\mu_\ell(E_i)^2 = \frac{\text{length}(E_i)^2}{2} ((\hat{x}_j - \hat{y}_j)^2 + (\hat{x}_k - \hat{y}_k)^2). \quad (5.11)$$

Listing 4 (vgl. [1, Listing 11]) zeigt die Implementierung in MATLAB:

- Die Funktion übernimmt das Netz $\mathcal{E}_\ell$ in Form von **coordinates** und **elements** und weiters die Verknüpfung zwischen dem Netz $\mathcal{E}_\ell$ und dessen uniformer Verfeinerung $\hat{\mathcal{E}}_\ell$ in Form von **father2son**. Darüber hinaus werden auch noch die Koeffizientenvektoren $\mathbf{x} \in \mathbb{R}^N$ und $\hat{\mathbf{x}} \in \mathbb{R}^{2N}$ von den Galerkin-Lösungen Φ_ℓ bzw. $\hat{\Phi}_\ell$ als Spaltenvektoren **x** bzw. **x_fine** übergeben. Als Output erhalten wir die quadrierten lokalen Fehleranteile $\mu_\ell(E)^2$ für jedes Element $E \in \mathcal{E}_\ell$ als Spaltenvektor (Zeile 1).
- Anschließend berechnen wir den Vektor von allen quadrierten Netzweiten (Zeile 3)
- Danach wird $\hat{\mathbf{x}}$ mit dem Koeffizientenvektor $\hat{\mathbf{x}} - \hat{\mathbf{y}} \in \mathbb{R}^{2N}$ von $\hat{\Phi}_\ell - \Phi_\ell$ überschrieben (Zeile 6 f.).
- Zum Schluss wird Formel (5.11) realisiert und der Vektor

$$\mathbf{v} := (\mu_\ell(E_1)^2, \dots, \mu_\ell(E_1)^2) \in \mathbb{R}^N$$

zurückgegeben, sodass $\mu_\ell = (\sum_{j=1}^N v_j)^{1/2}$ gilt.

5.5.2 Fehlerschätzer $\tilde{\mu}_\ell$ Listing 5: computeEstSlpMuTilde

```

1 function ind = computeEstSlpMuTilde(coordinates,elements,father2son,x_fine)
2 %*** compute (squared) local mesh-size
3 h = sum( (coordinates(elements(:,1),:)-coordinates(elements(:,2),:)).^2, 2);
4
5 %*** compute L2-projection Pi_coarse*phi_fine onto coarse mesh
6 pi_x_fine = 0.5*( x_fine(father2son(:,1),:)+ x_fine(father2son(:,2),:));
7
8 %*** compute coefficient vector of (1-Pi_coarse)*phi_fine
9 x_fine(father2son(:,1),:) = x_fine(father2son(:,1),:) - pi_x_fine;
10 x_fine(father2son(:,2),:) = x_fine(father2son(:,2),:) - pi_x_fine;
11
12 %*** compute ind(j) = diam(Ej)*|| (1-Pi_coarse)*phi_fine ||_{L2(Ej)}^2
13 ind = 0.5*h.*sum(( x_fine(father2son(:,1),:).^2 + x_fine(father2son(:,2),:).^2 ), 2);

```

Für die Berechnung des Fehlerschätzers

$$\tilde{\mu}_\ell = \left(\sum_{i=1}^N \tilde{\mu}_\ell(E_i)^2 \right)^{1/2}$$

mit $\mu_\ell(E_i)^2 = \text{length}(E_i)^{1/2} \|\hat{\Phi}_\ell - \Phi_\ell\|_{L^2(E_i)}$ gehen wir analog der Berechnung von $\tilde{\eta}_\ell$ in Abschnitt 5.4.2 vor. Mit $\mathbf{x}, \mathbf{y} \in \mathbb{R}^{2N}$ und

$$\hat{\Phi}_\ell = \sum_{i=1}^{2N} \hat{x}_i \hat{\chi}_i \quad \text{und} \quad \Pi_\ell \hat{\Phi}_\ell = \sum_{i=1}^{2N} \hat{y}_i \hat{\chi}_i,$$

wobei $\hat{y}_j = \hat{y}_k = \frac{\hat{x}_j + \hat{x}_k}{2}$ und $e_j, e_k \in \hat{\mathcal{E}}_\ell$ wieder die Söhne von $E_i \in \mathcal{E}_\ell$ darstellen. Die lokalen Fehleranteile können dann wieder mit Formel (5.11), wie für den Schätzer μ_ℓ , berechnet werden. D.h.

$$\tilde{\mu}_\ell(E_i)^2 = \frac{\text{length}(E_i)^2}{2} ((\hat{x}_j - \hat{y}_j)^2 + (\hat{x}_k - \hat{y}_k)^2).$$

Listing 5 (vgl. [1, Listing 12]) zeigt dann die Implementierung:

- Die Funktion übernimmt das Netz $\mathcal{E}_\ell$ in Form von **coordinates** und **elements**, die Verbindung **father2son** zwischen $\mathcal{E}_\ell$ und dessen Verfeinerung $\hat{\mathcal{E}}_\ell$ und den Koeffizientenvektor $\hat{\mathbf{x}} \in \mathbb{R}^{2N}$ der Galerkin-Lösung $\hat{\Phi}_\ell$ als Spaltenvektor **x_fine**. Als Output erhalten wir die quadrierten lokalen Fehleranteile $\tilde{\mu}_\ell(E)^2$ für jedes Element $E \in \mathcal{E}_\ell$ als Spaltenvektor (Zeile 1).
- Es wird der Vektor der quadrierten Netzweite (Zeile 3) und die Formel (5.10), also die Projektion $(\Pi_\ell \hat{\Phi})|_{E_i}$ bzgl. dem größeren Netz $\mathcal{E}_\ell$ für alle Elemente realisiert (Zeile 6).

- Anschließend wird $\widehat{\mathbf{x}}$ mit dem Koeffizientenvektor $\widehat{\mathbf{x}} - \widehat{\mathbf{y}} \in \mathbb{R}^{2N}$ von $\widehat{\Phi}_\ell - \Pi_\ell \widehat{\Phi}_\ell$ überschrieben (Zeile 9 f.).
- Zum Schluss wird Formel (5.11) realisiert und der Vektor

$$\mathbf{v} := (\tilde{\mu}_\ell(E_1)^2, \dots, \tilde{\mu}_\ell(E_N)^2) \in \mathbb{R}^N$$

zurückgegeben, sodass $\tilde{\mu}_\ell = (\sum_{j=1}^N v_j)^{1/2}$ gilt.

5.6 Two-Level-Fehlerschätzer τ_ℓ

Für die Berechnung der bisherigen Fehlerschätzer wurde immer die Lösung $\widehat{\Phi}_\ell$ bzgl. dem verfeinerten Netz $\widehat{\mathcal{E}}_\ell$ benötigt. Wir untersuchen nun einen Fehlerschätzer, dessen besondere Eigenschaft es ist, für die Berechnung nur die Lösung Φ_ℓ bzgl. dem gröberen Netz $\mathcal{E}_\ell$ zu verwenden. Wir definieren zuerst die Galerkin-Projektion:

Seien e_j, e_k die Söhne von $E_i \in \mathcal{E}_\ell$ mit den zugehörigen Indikatorfunktionen $\widehat{\chi}_j, \widehat{\chi}_k$. Dann ist die **Galerkin-Projektion** auf den eindimensionalen Unterraum $\text{span}\{\psi_{E_i}\}$ mit

$$\psi_{E_i} := -\widehat{\chi}_j + \widehat{\chi}_k$$

durch

$$\mathbb{G}_{E_i} \psi := \frac{\langle \psi, \psi_{E_i} \rangle_V}{\|\psi_{E_i}\|_V^2} \psi_{E_i} \quad \text{für alle } \psi \in H^{-1/2}(\Gamma)$$

definiert. Damit ist der Two-Level-Fehlerschätzer

$$\tau_\ell := \left(\sum_{j=1}^N \tau_\ell(E_j)^2 \right)^{1/2}$$

mit den lokalen Fehleranteilen

$$\tau_\ell(E_i) := \|\mathbb{G}_{E_i}(\widehat{\Phi}_\ell - \Phi_\ell)\|_V \quad (5.12)$$

definitionsgemäß der Klasse der lokalen Fehlerschätzer zugehörig. Da ψ_{E_i} und $\widehat{\Phi}_\ell - \Phi_\ell$ Elemente aus $\mathcal{P}^0(\widehat{\mathcal{E}}_\ell)$ sind, können wir die Galerkin-Variationsformulierung (3.10) für $\mathcal{P}^0(\widehat{\mathcal{E}}_\ell)$ anwenden und erhalten

$$\mathbb{G}_{E_i}(\widehat{\Phi}_\ell - \Phi_\ell) = \frac{\langle \widehat{\Phi}_\ell - \Phi_\ell, \psi_{E_i} \rangle_V}{\|\psi_{E_i}\|_V^2} \psi_{E_i} = \frac{\langle (K + 1/2) \widehat{G}_\ell, \psi_{E_i} \rangle_\Gamma - \langle \Phi_\ell, \psi_{E_i} \rangle_V}{\|\psi_{E_i}\|_V^2} \psi_{E_i}, \quad (5.13)$$

wobei $\widehat{G}_\ell \in \mathcal{S}^1(\widehat{\mathcal{E}}_\ell)$ mit

$$\widehat{G}_\ell = \sum_{i=1}^{2N} \widehat{g}_i \widehat{\zeta}_i \quad \text{mit } \widehat{g}_i = g(z_i)$$

der nodale Interpolant von den Randdaten g ist. Wir zitieren nun Theorem 4.6 aus [8] als

Satz 5.8 Der lokale Fehlerschätzer $\tau_\ell = \left(\sum_{i=1}^N \|\mathbb{G}_{E_i}(\widehat{\Phi}_\ell - \Phi_\ell)\|_V^2 \right)^{1/2}$ ist äquivalent zum Fehlerschätzer $\eta_\ell = \|\widehat{\Phi}_\ell - \Phi_\ell\|_V$. Insbesondere ist τ_ℓ immer effizient und unter der Saturationsannahme (3.34) auch zuverlässig bis auf Datenoszillationen. $\square$

Mit Satz 5.7 ist τ_ℓ auch zu allen anderen bisher eingeführten Fehlerschätzern äquivalent.

5.7 Implementierung des lokalen Fehlerschätzers τ_ℓ in MATLAB

Listing 6: computeEstSlpTau

```

1 function ind = computeEstSlpTau(father2son,V_fine,b_fine,x_coarse)
2 nE = size(x_coarse,1);
3
4 %*** build index vector son2father to link fine mesh with coarse mesh
5 son2father = zeros(2*nE,1);
6 son2father(father2son) = repmat((1:nE)',1,2);
7
8 %*** compute energy ||| psi_Ej |||^2 of two-level basis function
9 energy = 2*( V_fine(father2son(:,1) + 2*nE*(father2son(:,1) - 1)) ...
10             - V_fine(father2son(:,1) + 2*nE*(father2son(:,2) - 1)) );
11
12 %*** compute residual of x_coarse w.r.t. fine mesh
13 residual = b_fine - V_fine*x_coarse(son2father);
14
15 %*** compute vector of (squared) indicators w.r.t. coarse mesh
16 ind = ( residual(father2son(:,1)) - residual(father2son(:,2)) ).^2./energy;
```

Wir definieren wieder mit den Koeffizientenvektoren $\mathbf{x} \in \mathbb{R}^N$ und $\widehat{\mathbf{y}} \in \mathbb{R}^{2N}$ von Φ_ℓ bzgl. der Basis von $\mathcal{P}^0(\mathcal{E}_\ell)$ bzw. $\mathcal{P}^0(\widehat{\mathcal{E}}_\ell)$

$$\Phi_\ell = \sum_{j=1}^N x_j \chi_j = \sum_{j=1}^{2N} \widehat{y}_j \widehat{\chi}_j,$$

d.h. für ein Element $E_i \in \mathcal{E}_\ell$ und dessen Söhne $e_j, e_k \in \widehat{\mathcal{E}}_\ell$ gilt $x_i = \widehat{y}_j = \widehat{y}_k$. Weiters definieren wir das algebraische Residuum

$$\widehat{\mathbf{r}} := \widehat{\mathbf{b}} - \widehat{\mathbf{V}}\widehat{\mathbf{y}} \in \mathbb{R}^{2N}$$

mit der Einfachschichtpotential-Matrix $\widehat{\mathbf{V}} \in \mathbb{R}^{2N \times 2N}$ und $\widehat{\mathbf{b}} \in \mathbb{R}^{2N}$ aus (3.14) die Galerkin-Daten bzgl. $\widehat{\mathcal{E}}_\ell$. Mit der Definition (5.12) und Gleichung (5.13) erhalten wir

$$\tau_\ell(E_i)^2 = \|\mathbb{G}_{E_i}(\widehat{\Phi}_\ell - \Phi_\ell)\|_V^2 = \frac{|\langle (K+1/2) \widehat{G}_\ell, \psi_{E_i} \rangle_V - \langle \Phi_\ell, \psi_{E_i} \rangle_V|^2}{\|\psi_{E_i}\|_V^2}.$$

Mit $\widehat{G}_\ell \in \mathcal{S}^1(\widehat{\mathcal{E}}_\ell)$ als nodalen Interpolanten von g bzgl. der Basis $(\widehat{\zeta}_i)_{i=1}^{2N}$ gilt dann mit den Definitionen der Matrizen $\widehat{\mathbf{V}}, \widehat{\mathbf{K}}$ und $\widehat{\mathbf{M}}$, vgl. (3.11)-(3.13),

$$\langle\langle \Phi_\ell, \widehat{\chi}_j \rangle\rangle_V = \langle V\Phi_\ell, \widehat{\chi}_j \rangle_\Gamma = \sum_{i=1}^{2N} \widehat{y}_i \langle V\widehat{\chi}_i, \widehat{\chi}_j \rangle_\Gamma = \sum_{i=1}^{2N} \widehat{y}_i \widehat{\mathbf{V}}_{ji} = (\widehat{\mathbf{V}}\widehat{\mathbf{y}})_j$$

und

$$\begin{aligned} \langle (K+1/2)\widehat{G}_\ell, \widehat{\chi}_j \rangle_\Gamma &= \sum_{i=1}^{2N} \widehat{g}_i \langle (K+1/2)\widehat{\zeta}_i, \widehat{\chi}_j \rangle_\Gamma = \sum_{i=1}^{2N} \widehat{g}_i (\widehat{\mathbf{K}}_{ji} + \frac{1}{2}\widehat{\mathbf{M}}_{ji}) = (\widehat{\mathbf{K}}\widehat{\mathbf{g}} + \frac{1}{2}\widehat{\mathbf{M}}\widehat{\mathbf{g}})_j \\ &= (\widehat{\mathbf{b}})_j = \widehat{b}_j. \end{aligned}$$

Damit erhalten wir, aufgrund von $\psi_{E_i} = -\widehat{\chi}_j + \widehat{\chi}_k$ und $\|\widehat{\chi}_j\|_V = \|\widehat{\chi}_k\|_V$, schlussendlich

$$\begin{aligned} \tau_\ell(E_i)^2 &= \frac{|-\widehat{b}_j + \widehat{b}_k - (-\widehat{\mathbf{V}}\widehat{\mathbf{y}})_j + (\widehat{\mathbf{V}}\widehat{\mathbf{y}})_k|^2}{\|-\widehat{\chi}_j + \widehat{\chi}_k\|_V^2} = \frac{|\widehat{r}_k - \widehat{r}_j|^2}{\|\widehat{\chi}_j\|_V^2 - 2\langle\langle \widehat{\chi}_j, \widehat{\chi}_k \rangle\rangle_V + \|\widehat{\chi}_k\|_V^2} \\ &= \frac{|\widehat{r}_k - \widehat{r}_j|^2}{2(\|\widehat{\chi}_j\|_V^2 - \langle\langle \widehat{\chi}_j, \widehat{\chi}_k \rangle\rangle_V)}. \end{aligned} \quad (5.14)$$

Wir kommen zur Dokumentation von Listing 6:

- Die Funktion übernimmt den Koeffizientenvektor $\mathbf{x} \in \mathbb{R}^N$ der Galerkin-Lösung Φ_ℓ bzgl. dem Netz $\mathcal{E}_\ell$ als Spaltenvektor `x_coarse`. Weiters werden auch die Galerkin-Daten $\widehat{\mathbf{b}} \in \mathbb{R}^{2N}$, $\widehat{\mathbf{V}} \in \mathbb{R}^{2N}$ bzgl. dem uniform verfeinerten Netz $\widehat{\mathcal{E}}_\ell$ in Form von `b_fine`, `v_fine` und der Zusammenhang der beiden Netze mittels `father2son` übergeben. Dabei gilt für ein Element $E_i = e_j \cup e_k \in \mathcal{E}_\ell$ mit $e_j, e_k \in \widehat{\mathcal{E}}_\ell$, dass `father2son(i) = [j, k]`. Als Output erhalten wir die quadrierten lokalen Fehleranteile $\tau_\ell(E)^2$ für jedes Element $E \in \mathcal{E}_\ell$ als Spaltenvektor (Zeile 1).
- Es wird die Anzahl der Elemente (Zeile 2) und die Inverse zu `father2son` berechnet, sodass für ein Element $E_i = e_j \cup e_k \in \mathcal{E}_\ell$ mit $e_j, e_k \in \widehat{\mathcal{E}}_\ell$ `son2father(j) = son2father(k) = i` gilt (Zeile 5 f.).
- Anschließend wird die Energienorm

$$\|\psi_{E_i}\|_V^2 = 2(\|\widehat{\chi}_j\|_V^2 - \langle\langle \widehat{\chi}_j, \widehat{\chi}_k \rangle\rangle_V) = 2(\widehat{\mathbf{V}}_{jj} - \widehat{\mathbf{V}}_{kj})$$

für alle $i = 1, \dots, N$ berechnet. Dabei wird auf die zugrunde liegende spaltenweise Speicherung von Matrizen in FORTRAN eingegangen, d.h. auf die Einträge wird in der Form

$$\widehat{\mathbf{V}}_{jk} = \widehat{\mathbf{V}}_{kj} = \widehat{\mathbf{V}}_{k+2N(j-1)}$$

zugegriffen (Zeile 9 f.).

- Es wird das Residuum $\widehat{\mathbf{r}}$ berechnet, wobei das Indexfeld `son2father` die Söhne dem richtigen Index vom Koeffizientenvektor $\mathbf{x}$ bzgl. dem größeren Netz $\mathcal{E}_\ell$ zuordnet (Zeile 13).

- Zum Schluss wird Formel (5.14) ausgewertet und der Vektor

$$\mathbf{v} := (\tau_\ell(E_1)^2, \dots, \tau_\ell(E_N)^2) \in \mathbb{R}^N$$

zurückgegeben, sodass $\tau_\ell = (\sum_{j=1}^N v_j)^{1/2}$ gilt.

5.8 Gewichteter Residualschätzer ϱ_ℓ

Im Kapitel 3 haben wir gesehen, dass das Dirichlet-Randwertproblem (1.6) äquivalent zu Formel (3.4) ist, d.h.

$$V\phi = (K + 1/2)g \quad \text{auf } \Gamma.$$

Approximieren wir nun $G_\ell \approx g$ so gilt

$$V\phi_\ell = (K + 1/2)G_\ell \quad \text{auf } \Gamma$$

und mit der Galerkin-Lösung $\Phi_\ell \in \mathcal{P}(\mathcal{E}_\ell)$ der gestörten Variationsformulierung (3.10)

$$\langle V\Phi_\ell, \chi_j \rangle_\Gamma = \langle (K + 1/2)G_\ell, \chi_j \rangle_\Gamma \quad \text{für alle } j \in \{1, \dots, N\}. \quad (5.15)$$

Eine weitere Möglichkeit den Approximationsfehler $\|\phi - \Phi_\ell\|_V$ zu schätzen, führt nun über das Residuum R_ℓ , definiert durch

$$R_\ell := V\Phi_\ell - (K + 1/2)G_\ell = V(\Phi_\ell - \phi_\ell) \in H^{1/2}(\Gamma).$$

Wir motivieren mit Hilfe von [6, Abschnitt 2]: Mit gegebener Galerkin-Approximation $\Phi_\ell \in \mathcal{P}^0(\mathcal{E}_\ell) \subseteq H^{-1/2}(\Gamma)$ von ϕ gilt mit $V : H^{-1/2}(\Gamma) \rightarrow H^{1/2}(\Gamma)$ als beschränkten linearen Isomorphismus

$$C_1^{-1} \|R_\ell\|_{H^{1/2}(\Gamma)} \leq \|\phi_\ell - \Phi_\ell\|_{H^{-1/2}(\Gamma)} \leq C_2 \|R_\ell\|_{H^{1/2}(\Gamma)}, \quad (5.16)$$

wobei $C_1 := \|V\|$ und $C_2 := \|V^{-1}\|$ bezeichnen. Diese Abschätzungen bilden zum Einen die Grundlage für den Faermann-Residualschätzer, vgl. Abschnitt 5.10, und führen zum Anderen zur Definition des **gewichteten Residualschätzers**

$$\varrho_\ell := \|h_\ell^{1/2} R'_\ell\|_{L^2(\Gamma)} = \left(\sum_{j=1}^N \varrho_\ell(E_j)^2 \right)^{1/2} \quad (5.17)$$

mit den lokalen Fehleranteilen

$$\varrho_\ell(E_j) := \|h_\ell^{1/2} R'_\ell\|_{L^2(E_j)}.$$

Es gilt der folgende Satz, wobei wir die Effizienz bzgl. des Fehlers $\|\phi - \Phi_\ell\|_V$ für quasi-uniforme Netze² aus [3, Theorem 2] zitieren.

²Für ein quasi-uniformes Netz gilt $\max_{E \in \mathcal{E}_\ell} h_\ell|_E \leq C \min_{E \in \mathcal{E}_\ell} h_\ell|_E$ für alle $n \in \mathbb{N}$ mit einer $\mathcal{E}_\ell$ -unabhängigen Konstante $C > 0$.

Satz 5.9 Der gewichtete Residualschätzer ϱ_ℓ ist zuverlässig bis auf Datenoszillationen, d.h.

$$C_{\text{rel}}^{-1} \|\phi - \Phi_\ell\|_V \leq \varrho_\ell + \text{osc}_{D,\ell}.$$

Weiters ist ϱ_ℓ für quasi-uniforme Netze auch effizient bis auf Datenoszillationen, d.h.

$$C_{\text{eff}}^{-1} \varrho_\ell \leq \|\phi - \Phi_\ell\|_V + \text{osc}_{D,\ell},$$

wobei die Konstanten $C_{\text{eff}}, C_{\text{rel}} > 0$ jeweils $\mathcal{E}_\ell$ -unabhängig sind.

Für den Beweis brauchen wir noch das folgende Lemma (vgl. [4, Theorem 1]):

Lemma 5.10 Hat $f \in H^1(\Gamma)$ zumindest eine Nullstelle auf jedem Element E_j der Triangulierung $\mathcal{E}_\ell$, so gilt

$$\|f\|_{H^{1/2}(\Gamma)} \leq C_\ell \left(\sum_{j=1}^N \|h_\ell^{1/2} f'\|_{L^2(E_j)}^2 \right)^{1/2} = C_\ell \|h_\ell^{1/2} f'\|_{L^2(\Gamma)}$$

mit der Konstanten

$$C_\ell := C \left(\log(1 + \kappa(\mathcal{E}_\ell)) \right)^{(1/2)},$$

wobei $C > 0$, $\kappa(\mathcal{E}_\ell)$ die K -Netzkonstante aus (4.3) und $(\cdot)'$ die Bogenlängenableitung darstellt. $\square$

BEWEIS VON SATZ 5.9 (nur Zuverlässigkeit): Wir wollen Lemma 5.10 auf das Residuum

$$R_\ell = V\Phi_\ell - (K + 1/2) G_\ell$$

anwenden und zeigen nun, dass R_ℓ auf allen Elementen E_j zumindest eine Nullstelle hat.

Für die Operatoren V, K gilt $V : H^{-1/2+s}(\Gamma) \rightarrow H^{1/2+s}(\Gamma)$ und $K : H^{1/2+s}(\Gamma) \rightarrow H^{1/2+s}(\Gamma)$ mit $s \in [-1/2, 1/2]$, vgl. Lemma 3.1. Da $\Phi_\ell \in \mathcal{P}^0(\mathcal{E}_\ell)$, gilt auch $\Phi_\ell \in L^2(\Gamma)$ und mit $s = 1/2$ ist $V\Phi_\ell \in H^1(\Gamma)$. Weil Γ eine eindimensionale Mannigfaltigkeit ist, ist $V\Phi_\ell$ laut dem Einbettungssatz von Sobolev insbesondere stetig. Analog ist mit $G_\ell \in \mathcal{S}^1(\mathcal{E}_\ell) \subseteq H^1(\Gamma)$ auch $(K + 1/2) G_\ell \in H^1(\Gamma)$ stetig, womit insgesamt $R_\ell \in H^1(\Gamma)$ stetig ist. Mit (5.15) gilt

$$0 = \langle R_\ell, \chi_j \rangle_\Gamma = \int_{E_j} R_\ell d\Gamma \quad \text{für alle } E_j \in \mathcal{E}_\ell. \quad (5.18)$$

Die Stetigkeit von R_ℓ impliziert nun auf jedem Element $E_j \in \mathcal{E}_\ell$ zumindest eine Nullstelle. Die Kombination von (5.16) und Lemma 5.10 ergibt

$$\|\phi_\ell - \Phi_\ell\|_{H^{-1/2}(\Gamma)} \leq C_2 \|R_\ell\|_{H^{1/2}(\Gamma)} \leq C_2 C_\ell \varrho_\ell =: C_{\text{rel}} \varrho_\ell,$$

womit der gewichtete Residualschätzer ϱ_ℓ einen zuverlässigen Fehlerschätzer für den Fehler $\|\phi_\ell - \Phi_\ell\|_{H^{-1/2}(\Gamma)}$ darstellt. Da $\|\cdot\|_{H^{-1/2}(\Gamma)}$ eine zu $\|\cdot\|_V$ äquivalente Norm ist, gilt mit Dreiecksungleichung und zweimaligem Anwenden von (3.16)

$$\begin{aligned} \|\phi_\ell - \Phi_\ell\|_{H^{-1/2}(\Gamma)} &\lesssim \|\phi_\ell - \Phi_\ell\|_V \leq \|\phi - \Phi_\ell\|_V + \|\phi - \phi_\ell\|_V \\ &\lesssim \|\phi - \Phi_\ell\|_V + \text{osc}_{D,\ell} \lesssim \|\phi_\ell - \Phi_\ell\|_V + \text{osc}_{D,\ell}. \end{aligned} \quad (5.19)$$

Damit sind die Fehler $\|\phi - \Phi_\ell\|_V$ und $\|\phi_\ell - \Phi_\ell\|_V$ bis auf Datenoszillationen äquivalent. Daher ist der Fehlerschätzer ϱ_ℓ , vgl. Definition (5.7), zuverlässig bis auf Datenoszillationen. $\square$

Für die Implementierung gehen wir mit den selben Überlegungen wie bei den Dirichlet-Datenoszillationen aus Abschnitt 3.3 vor und approximieren das Residuum R_ℓ auf einem Element E_j durch ein Polynom $p_j \in \mathcal{P}^2[-1, 1]$, d.h. $p_j \approx R_\ell \circ \gamma_j : [-1, 1] \rightarrow \mathbb{R}$. Eine klassische Gaußquadratur auf $[-1, 1]$ mit Exaktheitsgrad 3 mit den zwei Knoten

$$x_1 = -\frac{\sqrt{3}}{3} \quad \text{und} \quad x_2 = \frac{\sqrt{3}}{3}$$

und den Gewichten $\omega_1 = \omega_2 = 1$ ersetzt dabei die Integration. Als Interpolationspolynom dienen uns wieder die Lagrange-Polynome mit den Stützstellen $(s_1, s_2, s_3) = (x_1, 0, x_2)$, d.h.

$$L_1(s) = \frac{1}{2}s(3s - \sqrt{3}), \quad L_2(s) = 1 - 3s^2, \quad L_3(s) = \frac{1}{2}s(3s + \sqrt{3}) \quad (5.20)$$

mit den zugehörigen Ableitungen

$$L'_1(s) = 3s - \frac{\sqrt{3}}{2}, \quad L'_2(s) = -6s, \quad L'_3(s) = 3s + \frac{\sqrt{3}}{2}.$$

Dann gilt für unser Interpolationspolynom bzw. für dessen Ableitung

$$p_j = \sum_{i=1}^3 R_\ell \circ \gamma_j(s_i) L_i \quad \text{bzw.} \quad p'_j = \sum_{i=1}^3 R_\ell \circ \gamma_j(s_i) L'_i.$$

Für die Quadratur müssen wir die abgeleiteten Lagrange-Polynome noch an den Knoten x_1, x_2 auswerten. Dies liefert

$$\begin{aligned} L'_1(x_1) &= -3\sqrt{\frac{3}{4}}, & L'_2(x_1) &= 2\sqrt{3}, & L'_3(x_1) &= -\sqrt{\frac{3}{4}}, \\ L'_1(x_2) &= \sqrt{\frac{3}{4}}, & L'_2(x_2) &= -2\sqrt{3}, & L'_3(x_2) &= 3\sqrt{\frac{3}{4}}. \end{aligned} \quad (5.21)$$

Mit Hilfe von Abschnitt 3.3 und $h := \text{length}(E_j)$ gilt

$$\begin{aligned} \varrho_\ell(E_j)^2 &= \|h^{1/2} R'_\ell\|_{L^2(E_j)}^2 = h \|R'_\ell\|_{L^2(E_j)}^2 = 2 \int_{-1}^1 ((R_\ell \circ \gamma_j)'(s))^2 ds \\ &\approx 2 \int_{-1}^1 (p'_j(s))^2 ds = 2 \text{quad}_3((p'_j)^2) = 2 \sum_{k=1}^2 \omega_k (p'_j(x_k))^2 \\ &= 2 \sum_{k=1}^2 \left(\sum_{i=1}^3 R_\ell \circ \gamma_j(s_i) L'_i(x_k) \right)^2, \end{aligned}$$

wobei $\text{quad}_3(\cdot)$ obige Gaußquadratur mit Exaktheitsgrad 3 bezeichnet, welche aufgrund von $(p'_j)^2 \in \mathcal{P}^2[-1, 1]$ exakt ist. Das bedeutet, wir approximieren

$$\varrho_\ell(E_j)^2 \approx 2 \sum_{k=1}^2 \left(\sum_{i=1}^3 R_\ell \circ \gamma_j(s_i) L'_i(x_k) \right)^2 =: \tilde{\varrho}_\ell(E_j)^2 \quad (5.22)$$

mit den approximierten lokalen Fehleranteilen $\tilde{\varrho}_\ell(E_j)$. Analog zu Abschnitt 3.3 existiert zu $p_j \in \mathcal{P}^2[-1, 1]$ ein Polynom $\tilde{p}_j \in \mathcal{P}^2(E_j)$ mit $\tilde{p}_j \circ \gamma_j = p_j$. Führen wir obige Schritte in umgekehrter Reihenfolge durch, erhalten wir

$$\tilde{\varrho}_\ell(E_j)^2 = 2 \text{quad}_3((p'_j)^2) = 2 \int_{-1}^1 (p'_j(s))^2 ds = h \|\tilde{p}'_j\|_{L^2(E_j)}^2$$

und insgesamt

$$\varrho_\ell \approx \tilde{\varrho}_\ell := \left(\sum_{j=1}^N \tilde{\varrho}_\ell(E_j)^2 \right)^{1/2} = \left(\sum_{j=1}^N \|h_\ell^{1/2} (\tilde{p}_j)'\|_{L^2(E_j)}^2 \right)^{1/2} = \|h_\ell^{1/2} \tilde{p}'\|_{L^2(\Gamma)}, \quad (5.23)$$

wobei $\tilde{p} \in \mathcal{P}^2(\mathcal{E}_\ell)$ das zusammengesetzte Polynom mit $\tilde{p}|_{E_j} = \tilde{p}_j$ ist.

Es gilt der folgende

Satz 5.11 (Quadraturfehler-Analyse). Für ein für alle Netze $\mathcal{E}_\ell$ nach oben gleichmäßig beschränktes und hinreichend glattes Residuum R_ℓ , gilt bei uniformer Netzverfeinerung mit Netzweite h :

- (a) $\varrho_\ell \geq C h^{3/2}$ für $\phi \in \mathcal{P}^1(\Gamma) \setminus \mathcal{P}^0(\Gamma)$ und $g \in \mathcal{P}^m(\Gamma) \setminus \mathcal{P}^1(\Gamma)$ mit einer Konstante $C > 0$ abhängig vom Rand Γ .
- (b) $\varrho_\ell = \|h_\ell^{1/2} R'_\ell\|_{L^2(\Gamma)} = \mathcal{O}(h^{3/2})$ für glatte Dirichlet-Daten $g \in H^{1/2}(\Gamma)$.
- (c) $|\varrho_\ell - \tilde{\varrho}_\ell| = \mathcal{O}(h^{5/2})$ für $R_\ell \in \mathcal{C}^3(\Gamma)$.

Insbesondere hat der Quadraturfehler für $\phi \in \mathcal{P}^1(\Gamma) \setminus \mathcal{P}^0(\Gamma)$ eine höhere Ordnung verglichen mit der Diskretisierungsordnung.

BEWEIS: Wegen der Zuverlässigkeit von ϱ_ℓ (Satz 5.9) gilt

$$\text{osc}_{D,\ell} + \varrho_\ell \geq C_{\text{rel}}^{-1} \|\phi - \Phi_\ell\|_V,$$

wobei die Datenoszillationen für $g \in \mathcal{P}^m(\Gamma) \setminus \mathcal{P}^1(\Gamma)$ und $m \geq 2$ laut Satz 3.3 (a) $\text{osc}_{D,\ell} \geq C' h^{3/2}$ erfüllen. Da laut Voraussetzung $\phi \in \mathcal{P}^1(\Gamma) \setminus \mathcal{P}^0(\Gamma)$ ist, gilt weiters mit der Konstante C_1 aus Lemma 5.6 und der Bestapproximation bzgl. der L^2 -Norm (3.30) mit $C_2 := C_{\text{rel}}^{-1} C_1^{-1}$

$$\text{osc}_{D,\ell} + \varrho_\ell \geq C_{\text{rel}}^{-1} \|\phi - \Phi_\ell\|_V \geq C_{\text{rel}}^{-1} C_1^{-1} \|h^{1/2}(\phi - \Phi_\ell)\|_{L^2(\Gamma)} \geq C_2 \|h^{1/2}(\phi - \Pi_\ell \phi)\|_{L^2(\Gamma)}.$$

Wir betrachten nun die lokalen Anteile und erhalten wie im Beweis von Satz 3.6

$$\|h^{1/2}(\phi - \Pi_\ell \phi)\|_{L^2(E_j)}^2 = \frac{h^2}{8} \int_{-1}^1 \left(\int_{-1}^1 \phi \circ \gamma_j(s) - \phi \circ \gamma_j(t) dt \right)^2 ds.$$

Wegen $\phi \in \mathcal{P}^1(\Gamma) \setminus \mathcal{P}^0(\Gamma) \subseteq \mathcal{C}^1(\Gamma)$ ist $(\phi \circ \gamma_j)'(\tilde{\xi})$ für alle $\tilde{\xi} \in (-1, 1)$ konstant. Nun folgt aus dem Mittelwertsatz der Differentialrechnung für alle festen $s, t \in (-1, 1)$ mit einer Zwischenstelle $\xi \in (-1, 1)$

$$\phi \circ \gamma_j(s) - \phi \circ \gamma_j(t) = (\phi \circ \gamma_j)'(\xi) \cdot (s - t) = \|(\phi \circ \gamma_j)'\|_{L^\infty(-1,1)} \cdot (s - t).$$

Analog dem Beweis von Satz 3.6 gilt weiters

$$\|h^{1/2}(\phi - \Pi_\ell \phi)\|_{L^2(E_j)}^2 = \frac{h^4}{12} \|\phi'\|_{L^\infty(E_j)}^2$$

und mit $N = 1/h$

$$\begin{aligned} \text{osc}_{D,\ell} + \varrho_\ell &\geq C_2 \|h^{1/2}(\phi - \Pi_\ell \phi)\|_{L^2(\Gamma)} = C_2 \left(\sum_{j=1}^N \|h^{1/2}(\phi - \Pi_\ell \phi)\|_{L^2(E_j)}^2 \right)^{1/2} \\ &= C_2 \left(\frac{h^4}{12} N \|\phi'\|_{L^\infty(\Gamma)}^2 \right)^{1/2} = C_2 \frac{\sqrt{3}}{6} h^{3/2} \|\phi'\|_{L^\infty(\Gamma)}, \end{aligned}$$

womit (a) bewiesen ist.

Laut Satz 5.9 ist ϱ_ℓ effizient bis auf Datenoszillationen bzgl. des Fehlers $\|\phi - \Phi_\ell\|_V$, d.h. es gilt

$$\varrho_\ell \lesssim \|\phi - \Phi_\ell\|_V + \text{osc}_{D,\ell}.$$

Daraus folgt wegen Satz 3.3 und Satz 3.6 unmittelbar die Behauptung (b).

Für die Behauptung (c) folgt mit (5.23) aus der umgekehrten Dreiecksungleichung

$$|\varrho_\ell - \tilde{\varrho}_\ell|^2 \leq \|h^{1/2}(R_\ell - \tilde{p})'\|_{L^2(\Gamma)}^2 = \sum_{j=1}^N h \|(R_\ell - \tilde{p})'\|_{L^2(E_j)}^2. \quad (5.24)$$

Analog zum Beweis von Satz 3.3 erhalten wir mit Lemma 3.2 mit $m = 2$ und $R_\ell \in \mathcal{C}^3(\Gamma)$ auf jedem Element E_j

$$h \|(R_\ell - \tilde{p})'\|_{L^2(E_j)}^2 \leq C^2 h^6 \|R_\ell'''\|_{L^\infty(E_j)}^2.$$

Unter der Voraussetzung, dass $\|R_\ell'''\|_{L^\infty(E_j)} \leq M < \infty$ für alle $j = 1, \dots, N$ unabhängig von $\mathcal{E}_\ell$ ist, folgt aus (5.24) mit $N = 1/h$ für uniforme Netze die zweite Behauptung. $\square$

5.9 Implementierung des gewichteten Residualschätzers ϱ_ℓ in MATLAB

Listing 7: computeEstSlpResidual

```

1 function ind = computeEstSlpResidual(coordinates,elements,phih,gh)
2 %*** Gaussian quadrature on [-1,1] with 2 nodes and exactness 3
3 quad_nodes = [-1 1]/sqrt(3);
4 quad_weights = [1;1];
5
6 %*** elementwise interpolation is done in (gauss_left,gauss_right,midpoint)
7 quad_nodes(3) = 0;
8 nE = size(elements,1);
9 nQ = length(quad_nodes);
10

```

```

11 %*** build vector of evaluations points as (nQ*nE x 2)-matrix
12 a = coordinates(elements(:,1),:);
13 b = coordinates(elements(:,2),:);
14 sx = reshape(a,2*nE,1)*(1-quad_nodes) + reshape(b,2*nE,1)*(1+quad_nodes);
15 sx = 0.5*reshape(sx',nQ*nE,2);
16
17 %*** evaluate gh elementwise at (left, right, midpoint)
18 gh_left = gh(elements(:,1));
19 gh_right = gh(elements(:,2));
20 gh_sx = gh_left*(1-quad_nodes) + gh_right*(1+quad_nodes);
21 gh_sx = 0.5*reshape(gh_sx',nQ*nE,1);
22
23 %*** evaluate p = V*phih - (K+1/2)*gh elementwise at (left, right, midpoint)
24 p = evaluateV(coordinates,elements,phih,sx) ...
25     - evaluateK(coordinates,elements,gh,sx) ...
26     - 0.5*gh_sx;
27 p = reshape(p,nQ,nE)';
28
29 %*** evaluate arclength-derivative p' elementwise at (left,right)
30 p_prime = p * [-3 1 ; -1 3 ; 4 -4]*sqrt(0.75);
31
32 %*** return ind(j) = diam(Ej) * || [ V*phi - (K+1/2)*gh ]' ||_{L2(Ej)}^2
33 ind = 2*p_prime.^2*quad_weights;

```

Listing 7 zeigt die Implementierung in MATLAB:

- In Zeile 1 wird das Netz $\mathcal{E}_\ell$ in Form von `coordinates` und `elements` eingelesen. Weiters werden die Koeffizienten $\mathbf{x}$ der Galerkin-Lösung Φ_ℓ als Spaltenvektor `phih` und die Funktion g als `gh` übernommen. Als Output erhalten wir die quadrierten lokalen Fehleranteile $\tilde{\varrho}_\ell(E)^2$ für jedes Element $E \in \mathcal{E}_\ell$ als Spaltenvektor.
- Es werden die Knoten und die Gewichte für die Quadratur definiert (Zeile 3 f).
- Für die Interpolation auf den Elementen $E_j \in \mathcal{E}_\ell$ wird noch der Mittelpunkt 0 in das Array `quad_nodes` beigefügt (Zeile 7). Weiters werden noch die Anzahl der Elemente (Zeile 8) und Anzahl der Knoten (Zeile 9) ausgelesen.
- Laut Formel (5.22) muss die Berechnung von $R_\ell \circ \gamma_j(s_i)$ für $i = 1, 2, 3$ je Element erfolgen. Es wird

$$\gamma_j(s_i) = \frac{1}{2}(a_j + b_j + s_i(b_j - a_j)) = \frac{1}{2}((1 - s_i)a_j + (1 + s_i)b_j).$$

für $(s_1, s_2, s_3) = (-\frac{\sqrt{3}}{3}, \frac{\sqrt{3}}{3}, 0)$ berechnet. Anschließend werden die Koordinaten in das

($\mathbf{nQ} \cdot \mathbf{nE} \times 2$)-Array $\mathbf{sx}$ in der Form

$$\mathbf{sx} = \begin{pmatrix} \text{coordinates}(\gamma_1(s_1)) \\ \text{coordinates}(\gamma_1(s_2)) \\ \text{coordinates}(\gamma_1(s_3)) \\ \text{coordinates}(\gamma_2(s_1)) \\ \text{coordinates}(\gamma_2(s_2)) \\ \text{coordinates}(\gamma_2(s_3)) \\ \vdots \\ \text{coordinates}(\gamma_{\mathbf{nE}}(s_1)) \\ \text{coordinates}(\gamma_{\mathbf{nE}}(s_2)) \\ \text{coordinates}(\gamma_{\mathbf{nE}}(s_3)) \end{pmatrix}$$

geschrieben. Dabei gilt $\text{coordinates}(\gamma_j(s_i)) = \mathbf{sx}(\mathbf{nQ}(j-1) + i)$ mit $\mathbf{nQ} = 3$ (Zeilen 12–15).

- Aufgrund der Linearität von G_ℓ erfolgt die Auswertung in den Knoten analog

$$G_\ell \circ \gamma_j(t) = \frac{1}{2}((1-t)G_\ell(a_j) + (1+t)G_\ell(b_j)) \quad \text{für } t \in \{s_1, s_2, s_3\}.$$

Das Ergebnis wird in das ($\mathbf{nQ} \cdot \mathbf{nE} \times 1$)-Koordinatenarray $\mathbf{gh_sx}$ gespeichert (Zeilen 18–21).

- Danach wird die Formel

$$R_\ell \circ \gamma_j(s_i) = (V\Phi_\ell - (K + 1/2)G_\ell) \circ \gamma_j(s_i)$$

ausgewertet und in das ($\mathbf{nE} \times \mathbf{nQ}$)-Array $\mathbf{p}$ geschrieben, wobei

$$\begin{aligned} \mathbf{p}(j, 1) &= R_\ell \circ \gamma_j(s_1) = R_\ell \circ \gamma_j(-\sqrt{3}/3) \\ \mathbf{p}(j, 2) &= R_\ell \circ \gamma_j(s_2) = R_\ell \circ \gamma_j(\sqrt{3}/3) \\ \mathbf{p}(j, 3) &= R_\ell \circ \gamma_j(s_3) = R_\ell \circ \gamma_j(0) \end{aligned}$$

für alle Elemente $E_j \in \mathcal{E}_\ell$ gilt (Zeilen 24–27). Zum Schluss erfolgt mit den Lagrange-Polynomen aus (5.21) und deren Auswertung in den Quadraturknoten $(x_1, x_2) = (s_1, s_2) = (-\frac{\sqrt{3}}{3}, \frac{\sqrt{3}}{3})$, die Realisierung von Formel (5.22) in Form einer Matrix-Vektor-Multiplikation (Zeilen 30 und 33). Zurückgegeben wird dann der Vektor

$$\mathbf{v} := (\tilde{\varrho}_\ell(E_1)^2, \dots, \tilde{\varrho}_\ell(E_N)^2) \in \mathbb{R}^N,$$

sodass $\tilde{\varrho}_\ell = (\sum_{j=1}^N v_j)^{1/2}$ gilt.

5.10 Faermann-Residualschätzer φ_ℓ

Mit dem Residuum

$$R_\ell = V\Phi_\ell - (K + 1/2) G_\ell = V(\Phi_\ell - \phi_\ell)$$

aus dem vorigen Abschnitt motivieren wir einen weiteren Fehlerschätzer für den Fehler $\|\phi - \Phi_\ell\|_V$ und gehen dabei wie in [6, Abschnitt 2] vor: Wir rufen die bereits in Kapitel 3 auf $H^{1/2}(\Gamma)$ definierte Sobolev-Slobodeckij-Norm (3.1)

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

mit der zugehörigen Halbnorm (3.2)

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

in Erinnerung. Bezugnehmend auf (5.16) könnte man, alternativ zum gewichteten Residualschätzer, versuchen $\|R_\ell\|_{H^{1/2}(\Gamma)}$ durch lokale Anteile auf den Elementen $E_j \in \mathcal{E}_\ell = \{E_1, \dots, E_N\}$ der Form $\|R_\ell\|_{H^{1/2}(E_j)}$ zu approximieren. Für die Sobolev-Slobodeckij-Norm aus (3.1) ist

$$\sum_{j=1}^N \|R_\ell\|_{H^{1/2}(E_j)}^2 \leq \|R_\ell\|_{H^{1/2}(\Gamma)}^2$$

leicht einzusehen. Problematisch ist allerdings, dass die umgekehrte Ungleichung nicht einmal in einer allgemeineren Form gilt, d.h. für alle $C > 0$ existiert eine Funktion $v \in H^{1/2}(\Gamma)$ mit

$$C \sum_{j=1}^N \|v\|_{H^{1/2}(E_j)}^2 < \|v\|_{H^{1/2}(\Gamma)}^2,$$

wie ein Gegenbeispiel in [5, Theorem 3.2] zeigt.

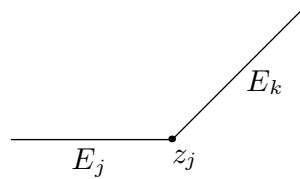


Abbildung 2: Überlappender Bereich $\omega_j = E_j \cup E_k$

Ein Ausweg aus diesem Dilemma besteht darin, die Halbnormen (3.2) auf den überlappenden Bereichen

$$\omega_j := \bigcup \{E \in \mathcal{E}_\ell : z_j \in E\},$$

mit den Knoten $z_j \in \mathcal{K}_\ell$ zu betrachten. Somit ist ein überlappender Bereich als Vereinigung zweier benachbarter Elemente der Triangulierung darstellbar. Aufgrund von adaptiver Netzverfeinerung,

werden benachbarte Elemente nicht immer fortlaufend nummeriert sein. Mit einer (bijektiven) Permutation dürfen wir aber o.B.d.A. von der Bezeichnung

$$\omega_j = E_j \cup E_k \quad \text{mit} \quad j \neq k \quad (5.25)$$

und $E_k \in \mathcal{E}_\ell$ als rechten Nachbar von E_j ausgehen, vgl. auch Abbildung 2. Zusammenfassend aus [5, Theorem 4.1] und [6, Proposition 3.1] gilt nun

Lemma 5.12 Für $v \in H^{1/2}(\Gamma)$ gilt:

(a) Mit einer $\mathcal{E}_\ell$ -unabhängigen Konstanten $C_1 > 0$ gilt

$$\sum_{j=1}^N |v|_{H^{1/2}(\omega_j)}^2 \leq C_1 \|v\|_{H^{1/2}(\Gamma)}^2.$$

(b) Ist zusätzlich $\int_E v \, d\Gamma = 0$ für alle $E \in \mathcal{E}_\ell$, dann gilt

$$\|v\|_{H^{1/2}(\Gamma)}^2 \leq C_2 \sum_{j=1}^N |v|_{H^{1/2}(\omega_j)}^2,$$

wobei die Konstante $C_2 > 0$ nur von $\kappa(\mathcal{E}_\ell)$ abhängt. □

Wegen (5.18) können wir Lemma 5.12 auf das Residuum $R_\ell \in H^1(\Gamma) \subseteq H^{1/2}(\Gamma)$ anwenden, sodass aus (5.16)

$$\tilde{C}_{\text{eff}}^{-1} \left(\sum_{j=1}^N |R_\ell|_{H^{1/2}(\omega_j)}^2 \right)^{1/2} \leq \|\phi_\ell - \Phi_\ell\|_{H^{-1/2}(\Gamma)} \leq \tilde{C}_{\text{rel}} \left(\sum_{j=1}^N |R_\ell|_{H^{1/2}(\omega_j)}^2 \right)^{1/2}$$

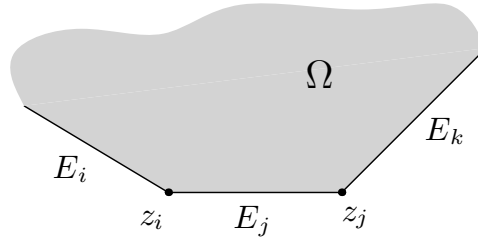
und weiters aus (5.19)

$$C_{\text{eff}}^{-1} \left(\sum_{j=1}^N |R_\ell|_{H^{1/2}(\omega_j)}^2 \right)^{1/2} \leq \|\phi - \Phi_\ell\|_V + \text{osc}_{D,\ell} \quad (5.26)$$

bzw.

$$C_{\text{rel}}^{-1} \|\phi - \Phi_\ell\|_V \leq \left(\sum_{j=1}^N |R_\ell|_{H^{1/2}(\omega_j)}^2 \right)^{1/2} + \text{osc}_{D,\ell} \quad (5.27)$$

mit gewissen Konstanten $\tilde{C}_{\text{eff}}, \tilde{C}_{\text{rel}}, C_{\text{eff}}, C_{\text{rel}} > 0$ folgt. Folglich ist $\left(\sum_{j=1}^N |R_\ell|_{H^{1/2}(\omega_j)}^2 \right)^{1/2}$ ein bis auf Datenoszillationen effizienter und zuverlässiger *knotenbezogener* Fehlerschätzer. Hinsichtlich der bei uns durchgeführten *elementbezogenen* adaptiven Netzverfeinerungsstrategie interessieren wir uns auch wieder für einen solchen Fehlerschätzer.


 Abbildung 3: Überlappender Bereich $\omega_j \cup \omega_{\pi(j)}$

Mit der Indexmenge $\mathcal{I}_j := \{k \in \{1, \dots, N\} : E_j \subseteq \omega_k\}$ definieren wir mit den lokalen Fehleranteilen

$$\varphi_\ell(E_j)^2 := \sum_{k \in \mathcal{I}_j} |R_\ell|_{H^{1/2}(\omega_k)}^2$$

den **Faermann-Residualschätzer**

$$\varphi_\ell := \left(\sum_{j=1}^N \varphi_\ell(E_j)^2 \right)^{1/2}$$

als weiteren elementbezogenen Fehlerschätzer.

Da wir von geschlossenen Rändern ausgehen, können wir dies noch kompakter anschreiben. Sei dazu $\pi : \mathbb{N} \rightarrow \mathbb{N}$ eine bijektive Abbildung, sodass $\omega_{\pi(j)} \cap \omega_j = E_j$ mit $\pi(j) \neq j$ gilt. Demnach ist $\pi(j)$ die Abbildung auf den linken Knoten (im mathematisch positiven Sinne) von $z_j \in \mathcal{K}_\ell$. Im Folgenden werden wir für die weitere Notation immer E_i als linkes Nachbarelement von $E_j \in \mathcal{E}_\ell$ annehmen, d.h. $\pi(j) = i$, vgl. auch Abbildung 3.

Damit gilt für die lokalen Fehleranteile auf einem Element $E_j \in \mathcal{E}_\ell$

$$\varphi_\ell(E_j)^2 = |R_\ell|_{H^{1/2}(\omega_{\pi(j)})}^2 + |R_\ell|_{H^{1/2}(\omega_j)}^2,$$

wobei $\omega_{\pi(j)} = E_i \cup E_j$ und $\omega_j = E_j \cup E_k$.

Satz 5.13 *Der Faermann-Residualschätzer φ_ℓ ist effizient und zuverlässig bis auf Datenoszillationen. Weiters gilt bei hinreichend glatten Dirichlet-Daten $g \in H^{1/2}(\Gamma)$ und Neumann-Daten $\phi \in H^{-1/2}(\Gamma)$ bei uniformer Netzverfeinerung mit Netzweite h*

$$\varphi_\ell = \mathcal{O}(h^{3/2}).$$

BEWEIS: Die erste Aussage folgt unmittelbar aus der Äquivalenz

$$\sum_{j=1}^N |R_\ell|_{H^{1/2}(\omega_j)}^2 \leq \sum_{j=1}^N \left(|R_\ell|_{H^{1/2}(\omega_{\pi(j)})}^2 + |R_\ell|_{H^{1/2}(\omega_j)}^2 \right) = \varphi_\ell^2 = 2 \sum_{j=1}^N |R_\ell|_{H^{1/2}(\omega_j)}^2$$

und den Ungleichungen (5.26), (5.27). Für die zweite Aussage erhalten wir aus Lemma 5.12 (a)

$$\varphi_\ell^2 = \sum_{j=1}^N \left(|R_\ell|_{H^{1/2}(\omega_{\pi(j)})}^2 + |R_\ell|_{H^{1/2}(\omega_j)}^2 \right) \leq 2 C_1 \|R_\ell\|_{H^{1/2}(\Gamma)}^2.$$

Es gilt $R_\ell = V(\phi_\ell - \Phi_\ell)$ mit dem beschränkten linearen Operator $V : H^{-1/2}(\Gamma) \rightarrow H^{1/2}(\Gamma)$. Folglich erhalten wir aufgrund der Äquivalenz $\|\cdot\|_{H^{-1/2}(\Gamma)} \sim \|\cdot\|_V$

$$\varphi_\ell^2 \lesssim \|R_\ell\|_{H^{1/2}(\Gamma)}^2 \lesssim \|\phi_\ell - \Phi_\ell\|_{H^{-1/2}(\Gamma)}^2 \lesssim \|\phi_\ell - \Phi_\ell\|_V^2.$$

Wegen (5.19) gilt $\|\phi_\ell - \Phi_\ell\|_V \leq \|\phi - \Phi_\ell\|_V + \text{osc}_{D,\ell}$ und weiters mittels Satz 3.3 und Satz 3.6 die zweite Behauptung. $\square$

Für die Implementierung des Faermann-Residualschätzers

$$\varphi_\ell^2 = \sum_{j=1}^N \varphi_\ell(E_j)^2 = \sum_{j=1}^N \left(|R_\ell|_{H^{1/2}(\omega_{\pi(j)})}^2 + |R_\ell|_{H^{1/2}(\omega_j)}^2 \right)$$

benötigen wir die lokalen Anteile $|R_\ell|_{H^{1/2}(\omega_j)}^2$ und orientieren uns für deren Berechnung an [6, Abschnitt 6.5]: Mit $\omega_j = E_j \cup E_k$ als Vereinigung des Elementes $E_j \in \mathcal{E}_\ell$ mit dessen rechten Nachbarn $E_k \in \mathcal{E}_\ell$ gilt

$$|R_\ell|_{H^{1/2}(\omega_j)}^2 = \sum_{m,n \in \{j,k\}} \int_{E_m} \int_{E_n} \frac{|R_\ell(x) - R_\ell(y)|^2}{|x - y|^2} d\Gamma(x) d\Gamma(y).$$

Die Transformation $(s, t) \mapsto (\gamma_m(s), \gamma_n(t)) = (x, y)$ mit der Referenzparametrisierung (2.1) führt mit $h_m := \text{length}(E_m)$ auf

$$|R_\ell|_{H^{1/2}(\omega_j)}^2 = \sum_{m,n \in \{j,k\}} \frac{h_m h_n}{4} \int_{-1}^1 \int_{-1}^1 \frac{|R_\ell \circ \gamma_m(s) - R_\ell \circ \gamma_n(t)|^2}{|\gamma_m(s) - \gamma_n(t)|^2} ds dt.$$

Mit der Definition

$$\mathbf{I}_{mn} := \frac{h_m h_n}{4} \int_{-1}^1 \int_{-1}^1 \frac{|R_\ell \circ \gamma_m(s) - R_\ell \circ \gamma_n(t)|^2}{|\gamma_m(s) - \gamma_n(t)|^2} ds dt \quad (5.28)$$

folgt dann

$$|R_\ell|_{H^{1/2}(\omega_j)}^2 = \mathbf{I}_{jj} + 2 \mathbf{I}_{jk} + \mathbf{I}_{kk}. \quad (5.29)$$

Für die Berechnung von $\mathbf{I}_{jk}$ in (5.29) mit $j \neq k$ wenden wir je Integral eine klassische Gauß-quadratur auf $[-1, 1]$ mit Exaktheitsgrad 3 an und bezeichnen sie mit $\text{quad}_{3 \times 3}(\cdot)$. Mit $\omega_j = E_j \cup E_k = [a_j, b_j] \cup [a_k, b_k]$, wobei $b_j = a_k$, impliziert dies mit den Gaußknoten

$$x_1 = -\frac{\sqrt{3}}{3} \quad \text{und} \quad x_2 = \frac{\sqrt{3}}{3}$$

und den Gaußgewichten $\omega_\sigma = 1$ für $\sigma = 1, 2$

$$\begin{aligned} \mathbf{I}_{jk} &= \frac{h_j h_k}{4} \int_{-1}^1 \int_{-1}^1 \frac{|R_\ell \circ \gamma_j(s) - R_\ell \circ \gamma_k(t)|^2}{|\gamma_j(s) - \gamma_k(t)|^2} ds dt =: \frac{h_j h_k}{4} \int_{-1}^1 \int_{-1}^1 w(s, t) ds dt \\ &\approx \frac{h_j h_k}{4} \text{quad}_{3 \times 3}(w) = \frac{h_j h_k}{4} \int_{-1}^1 \left(\sum_{\sigma=1}^2 \omega_\sigma w(x_\sigma, t) \right) dt = \frac{h_j h_k}{4} \sum_{\sigma, \tau=1}^2 \omega_\sigma \omega_\tau w(x_\sigma, x_\tau) \\ &= \frac{h_j h_k}{4} \sum_{\sigma, \tau=1}^2 w(x_\sigma, x_\tau), \end{aligned}$$

d.h. wir approximieren

$$\mathbf{I}_{jk} \approx \frac{h_j h_k}{4} \sum_{\sigma, \tau=1}^2 \frac{|R_\ell \circ \gamma_j(x_\sigma) - R_\ell \circ \gamma_k(x_\tau)|^2}{|\gamma_j(x_\sigma) - \gamma_k(x_\tau)|^2} =: \tilde{\mathbf{I}}_{jk} \quad \text{für } j \neq k. \quad (5.30)$$

Für die Terme $\mathbf{I}_{jj}$ und $\mathbf{I}_{kk}$ in (5.29) können wir anders vorgehen und approximieren $p_k \approx R_\ell \circ \gamma_k$ auf dem Element $E_k = [a_k, b_k]$ mit einem Polynom $p_k \in \mathcal{P}^2[-1, 1]$. Das Polynom p_k ist dann von der Form

$$p_k(x) = c_{k,2} x^2 + c_{k,1} x + c_{k,0} \quad (5.31)$$

für geeignete Koeffizienten $c_{k,i}$. Nun folgt mit der Definition (5.28)

$$\begin{aligned} \mathbf{I}_{kk} &= \frac{h_k^2}{4} \int_{-1}^1 \int_{-1}^1 \frac{|p_k(s) - p_k(t)|^2}{|\gamma_k(s) - \gamma_k(t)|^2} ds dt = h_k^2 \int_{-1}^1 \int_{-1}^1 \frac{|c_{k,2} s^2 + c_{k,1} s - c_{k,2} t^2 - c_{k,1} t|^2}{|s(b_k - a_k) - t(b_k - a_k)|^2} ds dt \\ &= \int_{-1}^1 \int_{-1}^1 \frac{|c_{k,2}(s^2 - t^2) + c_{k,1}(s - t)|^2}{|s - t|^2} ds dt = \int_{-1}^1 \int_{-1}^1 (c_{k,2}(s + t) + c_{k,1})^2 ds dt \\ &= \int_{-1}^1 \int_{-1}^1 c_{k,2}^2 (s^2 + t^2) + c_{k,1}^2 + 2c_{k,2}(c_{k,2}st + c_{k,1}s + c_{k,2}t) ds dt = \frac{8}{3} c_{k,2}^2 + 4c_{k,1}^2. \end{aligned}$$

Für das Interpolationspolynom p_k verwenden wir wieder die Lagrange-Polynome L_i aus (5.20) mit den Stützstellen $(s_1, s_2, s_3) = (x_1, 0, x_2) = (-\sqrt{3}/3, 0, \sqrt{3}/3)$. Definieren wir $r_{k,i} := R_\ell \circ \gamma_k(s_i)$, so folgt mit der Definition der Lagrange-Polynome

$$\begin{aligned} p_k(x) &= \sum_{i=1}^3 R_\ell \circ \gamma_k(s_i) L_i(x) = r_{k,1} \frac{1}{2} x(3x - \sqrt{3}) + r_{k,2} (1 - 3x^2) + r_{k,3} \frac{1}{2} x(3x + \sqrt{3}) \\ &= \frac{3}{2} (r_{k,1} - 2r_{k,2} - r_{k,3}) x^2 + \frac{\sqrt{3}}{2} (r_{k,3} - r_{k,1}) x + r_{k,2}. \end{aligned}$$

Damit gilt für die relevanten Koeffizienten in (5.31)

$$c_{k,2} = \frac{3}{2} (r_{k,1} - 2r_{k,2} - r_{k,3}) \quad \text{und} \quad c_{k,1} = \frac{\sqrt{3}}{2} (r_{k,3} - r_{k,1})$$

und weiters

$$\begin{aligned} \mathbf{I}_{kk} &= \frac{8}{3} \left(\frac{3}{2} (r_{k,1} - 2r_{k,2} - r_{k,3}) \right)^2 + 4 \left(\frac{\sqrt{3}}{2} (r_{k,3} - r_{k,1}) \right)^2 \\ &= 9r_{k,1}^2 + 24r_{k,2}^2 + 9r_{k,3}^2 - 24r_{k,1}r_{k,2} + 6r_{k,1}r_{k,3} - 24r_{k,2}r_{k,3}. \end{aligned}$$

Insgesamt erhalten wir dann

$$\mathbf{I}_{kk} = 3(r_{k,3} - r_{k,1})^2 + 6(r_{k,1} - 2r_{k,2} + r_{k,3})^2. \quad (5.32)$$

Selbiges gilt für $\mathbf{I}_{jj}$. Führen wir analoge Berechnungen für den Bereich $\omega_{\pi(j)} = E_i \cup E_j$ als Vereinigung von $E_j \in \mathcal{E}_\ell$ mit dessen linken Nachbarn $E_i \in \mathcal{E}_\ell$ durch, so erhalten wir zusammenfassend

$$\varphi_\ell(E_j)^2 = |R_\ell|_{H^{1/2}(\omega_{\pi(j)})}^2 + |R_\ell|_{H^{1/2}(\omega_j)}^2 \approx (\mathbf{I}_{jj} + 2\tilde{\mathbf{I}}_{jk} + \mathbf{I}_{kk}) + (\mathbf{I}_{ii} + 2\tilde{\mathbf{I}}_{ij} + \mathbf{I}_{jj})$$

und folglich die approximierten lokalen Fehleranteile

$$\varphi_\ell(E_j)^2 \approx 2(\mathbf{I}_{jj} + \tilde{\mathbf{I}}_{ij} + \tilde{\mathbf{I}}_{jk}) + \mathbf{I}_{ii} + \mathbf{I}_{kk} =: \tilde{\varphi}_\ell(E_j)^2 \quad (5.33)$$

mit den Definitionen (5.30), (5.32). Es gilt

$$\varphi_\ell \approx \tilde{\varphi}_\ell := \left(\sum_{j=1}^N \tilde{\varphi}_\ell(E_j)^2 \right)^{1/2}. \quad (5.34)$$

5.11 Implementierung des Faermann-Residualschätzers φ_ℓ in MATLAB

Listing 8: computeEstSlpFaermann

```

1 function ind = computeEstSlpFaermann(coordinates,elements,phi,gh)
2 %*** Gaussian quadrature on [-1,1] with 2 nodes and exactness 3
3 quad_nodes = [-1 1]/sqrt(3);
4 quad_sum = [1 1 2 2; 1 2 1 2];
5
6 %*** elementwise interpolation is done in (gauss_left,gauss_right,midpoint)
7 interpolation_nodes = [quad_nodes,0];
8
9 %*** define constants
10 nE = size(elements,1);
11 nC = size(coordinates,1);
12 nQ = length(quad_nodes);
13 nI = length(interpolation_nodes);
14
15 %*** build vector of evaluations points as (nI*nE x 2)-matrix
16 a = coordinates(elements(:,1),:);
17 b = coordinates(elements(:,2),:);
18 sx = reshape(a,2*nE,1)*(1-interpolation_nodes) ...
19       + reshape(b,2*nE,1)*(1+interpolation_nodes);
20 sx = 0.5*reshape(sx',nI*nE,2);
21

```

```

22 %*** evaluate gh elementwise at (left, right, midpoint)
23 gh_left = gh(elements(:,1));
24 gh_right = gh(elements(:,2));
25 gh_sx = gh_left*(1-interpolation_nodes) + gh_right*(1+interpolation_nodes);
26 gh_sx = 0.5*reshape(gh_sx',nI*nE,1);
27
28 %*** evaluate residual = V*phih - (K+1/2)*gh elementwise at (left, right, middle)
29 residual = evaluateV(coordinates,elements,phih,sx) ...
30             - evaluateK(coordinates,elements,gh,sx) ...
31             - 0.5*gh_sx;
32 residual = reshape(residual,nI,nE)';
33
34 %*** determine right neighbour Ek and left neighbour Ei of element Ej
35 node2rightElement = zeros(nC,1);
36 node2leftElement = zeros(nC,1);
37 node2rightElement(elements(:,1)) = (1:nE)';
38 node2leftElement(elements(:,2)) = (1:nE)';
39 element2rightNeighbour = node2rightElement(elements(:,2));
40 element2leftNeighbour = node2leftElement(elements(:,1));
41
42 %*** compute int(Ej x Ej) = int_{Ej} int_{Ej} |p(s)-p(t)|^2 / |s-t|^2 dt ds
43 %*** note: after transformation to [-1,1] and polynomial interpolation
44 %*** the integrand is of the type |q(s)-q(t)|^2 with a polynomial q
45 %*** and can thus be integrated analytically
46 intEjEj = 3*(residual(:,2)-residual(:,1)).^2 + 6*(residual(:,1) ...
47             +residual(:,2)-2*residual(:,3)).^2;
48
49 %*** compute local mesh-size
50 h = sqrt(sum((coordinates(elements(:,1),:)-coordinates(elements(:,2),:)).^2,2));
51
52 %*** compute int(Ej x Ek) = int_{Ej} int_{Ek} |R(x)-R(y)|^2 / |x-y|^2 dGamma(x,y)
53 j = [1:nE]';
54 k=element2rightNeighbour(j);
55 res_j = residual(j,quad_sum(1,:));
56 res_k = residual(k,quad_sum(2,:));
57 nom = (res_j - res_k).^2;
58 jidx = [(nI*j - (nI-1)) (nI*j-1)];
59 kidx = [(nI*k - (nI-1)) (nI*k-1)];
60 jidx = reshape(jidx(:,quad_sum(1,:))',nQ*nQ*nE,1);
61 kidx = reshape(kidx(:,quad_sum(2,:))',nQ*nQ*nE,1);
62 denom = sum((sx(jidx,:) - sx(kidx,:)).^2,2);
63 denom = reshape(denom,nQ*nQ,nE)';
64 intEjEk = 0.25*sum(nom./denom,2).*h(j).*h(k);
65
66 %*** return ind(j) = int(Ej x Ej) + 2*int(Ej x Ek) + int(Ek x Ek)
67 %***                  +int(Ei x Ei) + 2*int(Ei x Ej) + int(Ej x Ej)
68 ind = 2*(intEjEj + intEjEk + intEjEk(element2leftNeighbour) )...
69       + intEjEj(element2rightNeighbour) + intEjEj(element2leftNeighbour);

```

Wir kommen zur Dokumentation von Listing 8:

- In Zeile 1 wird das Netz $\mathcal{E}_\ell$ in Form von `coordinates` und `elements` eingelesen. Weiters werden die Koeffizienten $\mathbf{x}$ der Galerkin-Lösung Φ_ℓ als Spaltenvektor `phih` und die Funktion

g als **gh** übernommen. Als Output erhalten wir die quadrierten lokalen Fehleranteile $\tilde{\varphi}_\ell(E)^2$ für jedes Element $E \in \mathcal{E}_\ell$ als Spaltenvektor.

- Es werden die Gaußknoten definiert (Zeile 3). Das Array **quad_sum** (Zeile 4) gibt die Reihenfolge der Summierung für die spätere Quadratur vor.
- Für die elementweise Interpolation durch ein Polynom wird noch ein dritter Auswertepunkt den Gaußknoten beigelegt und in den Vektor **interpolation_nodes** geschrieben (Zeile 7).
- Anschließend wird die Anzahl der Elemente **nE**, Koordinaten **nC**, Quadraturknoten **nQ** und Interpolationsknoten **nI** ausgelesen (Zeilen 10–13).
- Für die Auswertung von R_ℓ werden die Punkte $\gamma_j(t)$ für $t \in \{s_1, s_2, s_3\} = \{-\frac{\sqrt{3}}{3}, \frac{\sqrt{3}}{3}, 0\}$ benötigt. Wegen Zeile 7 gilt in der Implementierung $(s_1, s_2, s_3) = (-\frac{\sqrt{3}}{3}, \frac{\sqrt{3}}{3}, 0)$. Die Koordinaten der Punkte

$$\gamma_j(s_i) = \frac{1}{2}((1 - s_i)a_j + (1 + s_i)b_j) \in E_j$$

werden dann in der Form

$$\mathbf{sx} = \begin{pmatrix} \text{coordinates}(\gamma_1(s_1)) \\ \text{coordinates}(\gamma_1(s_2)) \\ \text{coordinates}(\gamma_1(s_3)) \\ \text{coordinates}(\gamma_2(s_1)) \\ \text{coordinates}(\gamma_2(s_2)) \\ \text{coordinates}(\gamma_2(s_3)) \\ \vdots \\ \text{coordinates}(\gamma_{\mathbf{nE}}(s_1)) \\ \text{coordinates}(\gamma_{\mathbf{nE}}(s_2)) \\ \text{coordinates}(\gamma_{\mathbf{nE}}(s_3)) \end{pmatrix} \in \mathbb{R}^{\mathbf{nI} \cdot \mathbf{nE} \times 2}, \quad (5.35)$$

gespeichert, d.h. es gilt $\text{coordinates}(\gamma_j(s_i)) = \mathbf{sx}(\mathbf{nI}(j - 1) + i)$ mit $\mathbf{nI} = 3$ (Zeilen 16–20).

- Es werden die Werte von

$$G_\ell \circ \gamma_j(t) = \frac{1}{2}((1 - t)G_\ell(a_j) + (1 + t)G_\ell(b_j)) \quad \text{für } t \in \{s_1, s_2, s_3\}$$

berechnet (Zeilen 23–26) und das Residuum

$$R_\ell \circ \gamma_j(s_i) = (V\Phi_\ell - (K + 1/2)G_\ell) \circ \gamma_j(s_i)$$

für $i = 1, 2, 3$ ausgewertet (Zeilen 29–32). Dabei gilt $\mathbf{residual} \in \mathbb{R}^{\mathbf{nE} \times \mathbf{nI}}$ mit

$$\begin{aligned} \mathbf{residual}(j, 1) &= R_\ell \circ \gamma_j(s_1) = R_\ell \circ \gamma_j(-\sqrt{3}/3) \\ \mathbf{residual}(j, 2) &= R_\ell \circ \gamma_j(s_2) = R_\ell \circ \gamma_j(\sqrt{3}/3) \\ \mathbf{residual}(j, 3) &= R_\ell \circ \gamma_j(s_3) = R_\ell \circ \gamma_j(0) \end{aligned} \quad (5.36)$$

für alle Elemente $E_j \in \mathcal{E}_\ell$.

- Anschließend wird gemäß (5.25) zu jedem Element $E_j = [a_j, b_j]$ das Nachbarelement $E_k = [a_k, b_k]$ mit $b_j = a_k$ in den Vektor `element2rightNeighbour` $\in \mathbb{R}^{n_E}$ gespeichert. Folglich kann mittels `element2rightNeighbour(j) = k` und `elements(k, :)` auf das rechte Nachbarerelement von `elements(j, :)` zugegriffen werden. Analoges gilt für das linke Nachbarerelement von E_j (Zeilen 35–40).
- Bezugnehmend auf Formel (5.33) berechnen wir I_{kk} für alle Elemente $E_k \in \mathcal{E}_\ell$ durch (5.32), d.h.

$$I_{kk} = 3(r_{k,3} - r_{k,1})^2 + 6(r_{k,1} - 2r_{k,2} + r_{k,3})^2,$$

wobei $r_{k,i} = R_\ell \circ \gamma_j(\tilde{s}_i)$ mit $\tilde{s}_1 = s_1$, $\tilde{s}_2 = s_3$, $\tilde{s}_3 = s_2$ und (5.36) zu berücksichtigen ist (Zeile 46 f.).

- In Zeile 50 erfolgt die Berechnung der Netzweite aller Elemente.
- Es werden die Elemente mit den zugehörigen rechten Nachbarerelementen in die Vektoren `j` und `k` gespeichert (Zeile 53 f.). Nun erfolgt die Realisierung von Formel (5.30), d.h.

$$\tilde{I}_{jk} = \frac{h_j h_k}{4} \sum_{\sigma, \tau=1}^2 \frac{|R_\ell \circ \gamma_j(x_\sigma) - R_\ell \circ \gamma_k(x_\tau)|^2}{|\gamma_j(x_\sigma) - \gamma_k(x_\tau)|^2} \quad \text{für } j \neq k.$$

Die Reihenfolge der Summierung ist durch `quad_sum` in Zeile 4 festgelegt, d.h. σ durchläuft `quad_nodes(1, :)` und τ `quad_nodes(2, :)`.

Mit den Gaußknoten $x_1 = s_1$ und $x_2 = s_2$ werden unter Berücksichtigung von (5.36) die einzelnen Zähler der 4 Summanden berechnet und in den Vektor `nom` geschrieben (Zeilen 55–57).

Anschließend erfolgt die Ermittlung der Indizes `jidx`, `kidx` für die Auswertung des Nenners: Es wird auf die Indizes der in `sx` gespeicherten Elemente $\gamma_j(x_1)$, $\gamma_j(x_2)$ zugegriffen, vgl. (5.35), und in das Array `jidx` $\in \mathbb{R}^{n_E \times 2}$ geschrieben (Zeile 58). Aus diesem Indexvorrat werden nun die Indizes für die durch `quad_nodes(1, :)` festgelegte Reihenfolge entnommen und nach „reshapen“ neuerlich in den Vektor `jidx` $\in \mathbb{R}^{n_Q \cdot n_Q \cdot n_E}$ geschrieben (Zeile 60). Analoges passiert für `kidx`.

Nun wird der Nenner und insgesamt (5.30) ausgewertet (Zeilen 62–64).

- Zum Schluss wird $\tilde{\varphi}_\ell(E_j)^2$ mittels Formel (5.33) für alle $E_j \in \mathcal{E}_\ell$ berechnet und

$$\mathbf{v} := (\tilde{\varphi}_\ell(E_1)^2, \dots, \tilde{\varphi}_\ell(E_N)^2) \in \mathbb{R}^N$$

zurückgegeben, sodass $\tilde{\varphi}_\ell = (\sum_{j=1}^N v_i)^{1/2}$ gilt (Zeilen 68 f.).

5.12 Steinbach-Schätzer σ_ℓ

Als letzten Fehlerschätzer betrachten wir den Steinbach-Schätzer. Die Grundidee ist dabei die Approximation des Fehlers

$$\phi_\ell - \Phi_\ell \in H^{-1/2}(\Gamma) \quad \text{auf } \Gamma$$

bzgl. der Energienorm mit Hilfe einer Neumann-Reihe, wobei $\Phi_\ell \in \mathcal{P}^0(\mathcal{E}_\ell)$ wieder die Galerkin-Lösung der Variationsformulierung (3.10) mit dem nodalen Interpolanten $G_\ell \in \mathcal{S}^1(\mathcal{E}_\ell)$ und ϕ_ℓ die exakte Lösung der Symm'schen Integralgleichung

$$V\phi_\ell = (K + 1/2)G_\ell \quad (5.37)$$

darstellt. Formel (5.19) zeigt, dass dies eine bis auf Datenoszillationen äquivalente Approximation des Fehlers $\|\phi - \Phi_\ell\|_V$ ist. Wir werden zwei Möglichkeiten der Implementierung, jeweils eingeführt in [16] bzw. [17], vorstellen.

Bei der Motivation der Steinbach-Schätzer orientieren wir uns an [16, Abschnitt 2] und definieren beziehungsweise auf die Darstellungsformel (3.7)

$$(\tilde{u}_\ell)(\tilde{x}) := (\tilde{V}\Phi_\ell)(\tilde{x}) - (\tilde{K}G_\ell)(\tilde{x}) \quad \text{für alle } \tilde{x} \in \Omega \quad (5.38)$$

mit den Operatoren $\tilde{V}$ und $\tilde{K}$ aus (1.3) bzw. (1.4). Damit ergibt sich für die Spur $\tilde{u}_\ell|_\Gamma$, siehe etwa [18, Abschnitt 6.2 und 6.4],

$$\tilde{g}_\ell := \tilde{u}_\ell|_\Gamma = V\Phi_\ell + (1/2 - K)G_\ell \in H^{1/2}(\Gamma) \quad (5.39)$$

mit dem Einfachschichtpotential V aus (3.5) und dem Doppelschichtpotential K aus (3.6). Anwenden der Normalenableitung auf (5.38) führt zu

$$\tilde{\phi}_\ell := \frac{\partial}{\partial n} \tilde{u}_\ell = (1/2 + K')\Phi_\ell + WG_\ell \in H^{-1/2}(\Gamma), \quad (5.40)$$

siehe beispielsweise [18, Abschnitt 6.3 und 6.5]. Beteiligt sind das **adjungierte Doppelschichtpotential** $K' : H^{-1/2}(\Gamma) \rightarrow H^{-1/2}(\Gamma)$

$$(K'\phi)(x) := \int_\Gamma \frac{\partial}{\partial n_x} U(x, y) \phi(y) d\Gamma(y) = -\frac{1}{2\pi} \int_\Gamma \frac{(x - y) \cdot n_x}{|x - y|^2} \phi(y) d\Gamma(y)$$

und der **hypersinguläre Integraloperator** $W : H^{1/2}(\Gamma) \rightarrow H^{-1/2}(\Gamma)$

$$(Wg)(x) := -\frac{\partial}{\partial n_x} \int_\Gamma \frac{\partial}{\partial n_y} U(x, y) g(y) d\Gamma(y) = -\frac{\partial}{\partial n_x} (Kg)(x)$$

für alle $x \in \Gamma$. Analog dem Lemma 3.1 für V und K zitieren wir wieder [18, Satz 6.11], sodass auch für K' und W gilt:

Lemma 5.14 *Sei Γ der Rand eines Lipschitz-Gebietes Ω . Dann sind die linearen Randintegraloperatoren*

$$\begin{aligned} K' &: H^{-1/2+s}(\Gamma) \rightarrow H^{-1/2+s}(\Gamma) \\ W &: H^{1/2+s}(\Gamma) \rightarrow H^{-1/2+s}(\Gamma) \end{aligned}$$

für alle $s \in [-\frac{1}{2}, \frac{1}{2}]$ beschränkt. □

Weiters erfüllen jeweils $\tilde{V}\Phi_\ell$ und $\tilde{K}g$ die Laplace-Gleichung, siehe etwa [18, Lemma 6.3 und 6.7], sodass mit (5.39) insgesamt

$$\begin{aligned} -\Delta \tilde{u}_\ell &= 0 \quad \text{in } \Omega \\ \tilde{u}_\ell &= \tilde{g}_\ell \quad \text{auf } \Gamma \end{aligned}$$

und folglich die Symm'sche Integralgleichung (3.4) in der Form

$$V\tilde{\phi}_\ell = (K + 1/2)\tilde{g}_\ell \quad \text{auf } \Gamma \tag{5.41}$$

gilt. Wir schreiben nun Lemma 2.1 aus [16] als folgendes

Lemma 5.15 *Der Fehler $\phi_\ell - \Phi_\ell$ ist Lösung der Randintegralgleichung*

$$(1/2 - K')(\phi_\ell - \Phi_\ell)(x) = (\tilde{\phi}_\ell - \Phi_\ell)(x)$$

für alle $x \in \Gamma$.

BEWEIS: Schreiben wir (5.37) in der Form

$$G_\ell = V\phi_\ell + (1/2 - K)G_\ell$$

und subtrahieren wir davon (5.39) so erhalten wir

$$V(\phi_\ell - \Phi_\ell) = G_\ell - \tilde{g}_\ell \quad \text{auf } \Gamma. \tag{5.42}$$

Weiters gilt mit (5.41) und (5.37)

$$V(\tilde{\phi}_\ell - \phi_\ell) = (K + 1/2)(\tilde{g}_\ell - G_\ell) \quad \text{auf } \Gamma. \tag{5.43}$$

Demnach erhalten wir aufgrund der Linearität der Operatoren V , K , (5.42) und (5.43)

$$\begin{aligned} V(\tilde{\phi}_\ell - \Phi_\ell) &= V(\tilde{\phi}_\ell - \phi_\ell) + V(\phi_\ell - \Phi_\ell) = -(K + 1/2)(G_\ell - \tilde{g}_\ell) + G_\ell - \tilde{g}_\ell \\ &= (1/2 - K)(G_\ell - \tilde{g}_\ell) = (1/2 - K)V(\phi_\ell - \Phi_\ell) \end{aligned}$$

Weiters erhalten wir wegen $KV = VK'$, vgl. [18, Folgerung 6.3],

$$V(\tilde{\phi}_\ell - \Phi_\ell) = V(1/2 - K')(\phi_\ell - \Phi_\ell).$$

Aus der Bijektivität von V folgt die Behauptung. □

Um die Invertierbarkeit von $1/2 - K'$ nachzuweisen, zitieren wir eine Kontraktionsabschätzung aus [18, Folgerung 6.8]:

Lemma 5.16 *Der Operator $T : H^{-1/2}(\Gamma) \rightarrow H^{-1/2}(\Gamma)$, definiert durch*

$$T := 1/2 + K', \quad (5.44)$$

ist beschränkt mit einer Kontraktionskonstante $C_T < 1$. D.h. es gilt

$$\|T\psi\|_V \leq C_T \|\psi\|_V$$

für alle $\psi \in H^{-1/2}(\Gamma)$. □

Da $(H^{-1/2}(\Gamma), \|\cdot\|_V)$ ein Banachraum ist und weiters für den linearen, beschränkten Operator T wegen Lemma 5.16 $\|T\| < 1$ gilt, konvergiert die **Neumann-Reihe**

$$\sum_{\nu=0}^{\infty} T^{\nu}.$$

Insbesondere ist der Operator $(1/2 - K') = (I - T) : H^{-1/2}(\Gamma) \rightarrow H^{-1/2}(\Gamma)$ bijektiv, und es gilt

$$(I - T)^{-1} = \sum_{\nu=0}^{\infty} T^{\nu} < \infty. \quad (5.45)$$

Mit Definition von T und Lemma 5.15 erhalten wir auf dem Rand Γ

$$(I - T)(\phi_{\ell} - \Phi_{\ell}) = (1/2 - K')(\phi_{\ell} - \Phi_{\ell}) = \tilde{\phi}_{\ell} - \Phi_{\ell} \quad (5.46)$$

was mit (5.45) zu

$$\varepsilon_{\ell} := \phi_{\ell} - \Phi_{\ell} = \sum_{\nu=0}^{\infty} T^{\nu}(\tilde{\phi}_{\ell} - \Phi_{\ell}) \in H^{-1/2}(\Gamma) \quad (5.47)$$

führt. Damit lautet die approximierte Fehlerfunktion mit $q \in \mathbb{N}_0$ und $\varepsilon_{\ell}^{(0)} := \tilde{\phi}_{\ell} - \Phi_{\ell}$

$$\varepsilon_{\ell}^{(q)} := \sum_{\nu=0}^q T^{\nu}(\tilde{\phi}_{\ell} - \Phi_{\ell}) = \sum_{\nu=0}^q T^{\nu} \varepsilon_{\ell}^{(0)} \in H^{-1/2}(\Gamma). \quad (5.48)$$

Aus (5.46) und der Definition (5.47) für ε_{ℓ} erhalten wir

$$\varepsilon_{\ell}^{(q)} = \sum_{\nu=0}^q T^{\nu}(\tilde{\phi}_{\ell} - \Phi_{\ell}) = \sum_{\nu=0}^q T^{\nu}(I - T)\varepsilon_{\ell} = \sum_{\nu=0}^q T^{\nu} \varepsilon_{\ell} - \sum_{\nu=0}^q T^{\nu+1} \varepsilon_{\ell} = \varepsilon_{\ell} - T^{q+1} \varepsilon_{\ell}.$$

Damit folgt einerseits mit Lemma 5.16

$$\|\varepsilon_{\ell}^{(q)}\|_V \leq \|\varepsilon_{\ell}\|_V + C_T^{q+1} \|\varepsilon_{\ell}\|_V = (1 + C_T^{q+1}) \|\varepsilon_{\ell}\|_V$$

und andererseits mit umgekehrter Dreiecksungleichung und neuerlichem Anwenden von Lemma 5.16

$$\begin{aligned}\|\varepsilon_\ell^{(q)}\|_V &\geq \left| \|\varepsilon_\ell\|_V - \|T^{q+1}\varepsilon_\ell\|_V \right| \geq \|\varepsilon_\ell\|_V - \|T^{q+1}\varepsilon_\ell\|_V \geq \|\varepsilon_\ell\|_V - C_T^{q+1}\|\varepsilon_\ell\|_V \\ &= (1 - C_T^{q+1})\|\varepsilon_\ell\|_V.\end{aligned}$$

Insgesamt gilt also

$$\frac{1}{1 + C_T^{q+1}}\|\varepsilon_\ell^{(q)}\|_V \leq \|\phi_\ell - \Phi_\ell\|_V \leq \frac{1}{1 - C_T^{q+1}}\|\varepsilon_\ell^{(q)}\|_V, \quad (5.49)$$

d.h. $\|\varepsilon_\ell^{(q)}\|_V$ ist wegen $C_T < 1$ asymptotisch exakt für $q \rightarrow \infty$.

Unser Ziel ist eine für die Praxis anwendbare Form der Bauart (5.48). Sei dazu $\mathcal{E}_\ell^{(k)}$, $k \in \mathbb{N}$, eine beliebige k -fache uniforme Verfeinerung der Triangulierung $\mathcal{E}_\ell$ mit der lokalen Netzweite $h_\ell^{(k)} \in \mathcal{P}^0(\mathcal{E}_\ell^{(k)})$, sodass

$$h_\ell^{(k)}|_E = \text{length}(E) \quad \text{für alle } E \in \mathcal{E}_\ell^{(k)}$$

gilt. Weiters sei $\Pi_\ell^{(k)}$ die durch (3.29) definierte L^2 -Orthogonalprojektion auf $\mathcal{P}^0(\mathcal{E}_\ell^{(k)})$. Dann definieren wir

$$e_\ell^{(k,q)} := \sum_{\nu=0}^q (\Pi_\ell^{(k)} T)^\nu e_\ell^{(k)} \in \mathcal{P}^0(\mathcal{E}_\ell^{(k)}) \quad \text{mit} \quad e_\ell^{(k)} := \Pi_\ell^{(k)}(\tilde{\phi}_\ell - \Phi_\ell) \quad (5.50)$$

und $e_\ell^{(k,0)} := e_\ell^{(k)} = \Pi_\ell^{(k)}\varepsilon_\ell^{(0)}$ für alle $q \in \mathbb{N}_0$. Wegen $\Phi_\ell \in \mathcal{P}^0(\mathcal{E}_\ell) \subseteq L^2(\Gamma)$ und $G_\ell \in \mathcal{S}^1(\mathcal{E}_\ell) \subseteq H^1(\Gamma)$ erhalten wir mit Lemma 5.14 $\tilde{\phi}_\ell = (1/2 + K')\Phi_\ell + WG_\ell \in L^2(\Gamma)$ und folglich auch $\tilde{\phi}_\ell - \Phi_\ell \in L^2(\Gamma)$. Weiters ist $T = 1/2 + K' : L^2(\Gamma) \rightarrow L^2(\Gamma)$ wegen Lemma 5.14 beschränkt und daher (5.50) sinnvoll definiert.

Als Berechnungsmethode von $e_\ell^{(k,q)}$ bietet sich ein iteratives Verfahren an, denn mit der Definition $c_\ell^{(k,p)} := (\Pi_\ell^{(k)} T)^p e_\ell^{(k)} \in \mathcal{P}^0(\mathcal{E}_\ell^{(k)})$ für $p \in \mathbb{N}$ erhalten wir

$$e_\ell^{(k,p)} = \sum_{\nu=0}^p (\Pi_\ell^{(k)} T)^\nu e_\ell^{(k)} = e_\ell^{(k,p-1)} + (\Pi_\ell^{(k)} T)^p e_\ell^{(k)} = e_\ell^{(k,p-1)} + c_\ell^{(k,p)}$$

und daher eine berechenbare Folge

$$e_\ell^{(k,p)} = e_\ell^{(k,p-1)} + c_\ell^{(k,p)} \quad \text{mit} \quad c_\ell^{(k,p)} = \Pi_\ell^{(k)} T c_\ell^{(k,p-1)} \quad (5.51)$$

für $p \in \{1, \dots, q\}$ und $c_\ell^{(k,0)} := e_\ell^{(k,0)} = e_\ell^{(k)} = \Pi_\ell^{(k)}\varepsilon_\ell^{(0)}$.

Wir schreiben nun Theorem 3.3 aus [17] als

Lemma 5.17 Sei $\tilde{\phi}_\ell$ durch (5.40) und $e_\ell^{(k,q)}$ durch (5.50) definiert. Weiters bezeichne Φ_ℓ die zur Triangulierung $\mathcal{E}_\ell$ gehörige Galerkin-Lösung. Dann gilt mit

$$\varepsilon_\ell = \phi_\ell - \Phi_\ell \quad \text{und} \quad \varepsilon_\ell^{(0)} = \tilde{\phi}_\ell - \Phi_\ell$$

und der maximalen Netzweite der Triangulierung $\|h_\ell^{(k)}\|_{L^\infty(\Gamma)} = \max_{E \in \mathcal{E}_\ell^{(k)}} h_\ell^{(k)}|_E$ die Beziehung

$$\begin{aligned} & \frac{1}{1 + C_T^{q+1}} \left(\|e_\ell^{(k,q)}\|_V - C(q+1) \|h_\ell^{(k)}\|_{L^\infty(\Gamma)}^{1/2} \|\varepsilon_\ell^{(0)}\|_{L^2(\Gamma)} \right) \\ & \leq \|\varepsilon_\ell\|_V \leq \frac{1}{1 - C_T^{q+1}} \left(\|e_\ell^{(k,q)}\|_V + C(q+1) \|h_\ell^{(k)}\|_{L^\infty(\Gamma)}^{1/2} \|\varepsilon_\ell^{(0)}\|_{L^2(\Gamma)} \right), \end{aligned}$$

wobei $C_T < 1$ die Konstante aus Lemma 5.16 und die Konstante $C > 0$ von $q \in \mathbb{N}_0$ abhängig ist.

Insbesondere ist der Fehlerschätzer $\|e_\ell^{(k,q)}\|_V$ für hinreichend kleines $\|h_\ell^{(k)}\|_{L^\infty(\Gamma)} < \|h_\ell\|_{L^\infty(\Gamma)}$ effizient und zuverlässig bis auf Datenoszillationen.

BEWEIS: Wir orientieren uns an der Beweisführung in [17, Abschnitt 3] und teilen den Beweis in mehrere Schritte:

(i) Als Erstes zeigen wir, dass

$$\|\Pi_\ell^{(k)} v\|_{L^2(\Gamma)} \leq \|v\|_{L^2(\Gamma)} \quad (5.52)$$

und

$$\|v - \Pi_\ell^{(k)} v\|_V \leq C_1 \|h_\ell^{(k)}\|_{L^\infty(\Gamma)}^{1/2} \|v\|_{L^2(\Gamma)} \quad (5.53)$$

für alle $v \in L^2(\Gamma)$ mit einer Konstanten $C_1 > 0$ gilt.

Aus der Orthogonalität bzgl. der L^2 -Norm folgt

$$\langle \Pi_\ell^{(k)} v, \Pi_\ell^{(k)} v \rangle_{L^2(\Gamma)} = \langle v, \Pi_\ell^{(k)} v \rangle_{L^2(\Gamma)} \quad \text{für alle } v \in L^2(\Gamma)$$

und damit weiters mit Cauchy-Schwarz

$$\|\Pi_\ell^{(k)} v\|_{L^2(\Gamma)}^2 = \langle \Pi_\ell^{(k)} v, \Pi_\ell^{(k)} v \rangle_{L^2(\Gamma)} = \langle v, \Pi_\ell^{(k)} v \rangle_{L^2(\Gamma)} \leq \|v\|_{L^2(\Gamma)} \|\Pi_\ell^{(k)} v\|_{L^2(\Gamma)},$$

was durch Kürzen (5.52) ergibt. Die Abschätzung (5.53) folgt unmittelbar aus Lemma 3.5.

(ii) Es gilt für alle $v \in L^2(\Gamma)$

$$\|(1/2 + K') v\|_{L^2(\Gamma)} \leq 1/2 \|v\|_{L^2(\Gamma)} + \|K' v\|_{L^2(\Gamma)}$$

und weiters mit $T = 1/2 + K'$ und der Beschränktheit von K' aufgrund von Lemma 5.14 mit einer Konstante $\tilde{C}_T > 0$

$$\|T v\|_{L^2(\Gamma)} \leq \tilde{C}_T \|v\|_{L^2(\Gamma)} \quad \text{für alle } v \in L^2(\Gamma). \quad (5.54)$$

- (iii) Wir gehen zuerst von $q > 0$ aus und definieren analog (5.51) eine iterative Berechnungsvorschrift für $\varepsilon_\ell^{(q)} = \sum_{\nu=0}^q T^\nu \varepsilon_\ell^{(0)}$ durch

$$\varepsilon_\ell^{(p)} = \varepsilon_\ell^{(p-1)} + \varkappa_\ell^{(p)} \quad \text{mit} \quad \varkappa_\ell^{(p)} := T \varkappa_\ell^{(p-1)}$$

für $p \in \{1, \dots, q\}$ und $\varkappa_\ell^{(0)} := \varepsilon_\ell^{(0)}$. Dann gilt für jedes $p \in \{1, \dots, q\}$ mit Dreiecksungleichung

$$\|e_\ell^{(k,p)} - \varepsilon_\ell^{(p)}\|_V \leq \|e_\ell^{(k,p-1)} - \varepsilon_\ell^{(p-1)}\|_V + \|c_\ell^{(k,p)} - \varkappa_\ell^{(p)}\|_V.$$

Folglich erhalten wir mit $c_\ell^{(k,0)} = e_\ell^{(k,0)} = e_\ell^{(k)}$ und $\varkappa_\ell^{(0)} = \varepsilon_\ell^{(0)}$ aus obiger Rekursion

$$\|e_\ell^{(k,q)} - \varepsilon_\ell^{(q)}\|_V \leq \sum_{\nu=0}^q \|c_\ell^{(k,\nu)} - \varkappa_\ell^{(\nu)}\|_V. \quad (5.55)$$

Mit der Definition

$$\hat{c}_\ell^{(k,p)} := (1/2 + K') c_\ell^{(k,p-1)} = T c_\ell^{(k,p-1)} \quad \text{für } p \in \{1, \dots, q\}$$

erhalten wir für die einzelnen Summanden in Gleichung (5.55) weiters

$$\begin{aligned} \|c_\ell^{(k,p)} - \varkappa_\ell^{(p)}\|_V &\leq \|c_\ell^{(k,p)} - \hat{c}_\ell^{(k,p)}\|_V + \|\hat{c}_\ell^{(k,p)} - \varkappa_\ell^{(p)}\|_V \\ &= \|c_\ell^{(k,p)} - \hat{c}_\ell^{(k,p)}\|_V + \|T(c_\ell^{(k,p-1)} - \varkappa_\ell^{(p-1)})\|_V, \end{aligned}$$

was mit der Kontraktionsabschätzung aus Lemma 5.16 zu

$$\|c_\ell^{(k,p)} - \varkappa_\ell^{(p)}\|_V \leq \|c_\ell^{(k,p)} - \hat{c}_\ell^{(k,p)}\|_V + C_T \|c_\ell^{(k,p-1)} - \varkappa_\ell^{(p-1)}\|_V \quad (5.56)$$

führt. Weil wegen $c_\ell^{(k,p-1)} \in L^2(\Gamma)$ aus Lemma 5.14 auch $T c_\ell^{(k,p-1)} \in L^2(\Gamma)$ für $p \in \{1, \dots, q\}$ folgt, gilt mit (5.53) und weiters mit (5.54) für den ersten Term in (5.56)

$$\begin{aligned} \|c_\ell^{(k,p)} - \hat{c}_\ell^{(k,p)}\|_V &= \|\Pi_\ell^{(k)} T c_\ell^{(k,p-1)} - T c_\ell^{(k,p-1)}\|_V \leq C_1 \|h_\ell^{(k)}\|_{L^\infty(\Gamma)}^{1/2} \|T c_\ell^{(k,p-1)}\|_{L^2(\Gamma)} \\ &\leq C_1 \tilde{C}_T \|h_\ell^{(k)}\|_{L^\infty(\Gamma)}^{1/2} \|c_\ell^{(k,p-1)}\|_{L^2(\Gamma)}. \end{aligned}$$

Schreiben wir $c_\ell^{(k,p)} = \Pi_\ell^{(k)} T c_\ell^{(k,p-1)} = (\Pi_\ell^{(k)} T)^p e_\ell^{(k)} = (\Pi_\ell^{(k)} T)^p \Pi_\ell^{(k)} \varepsilon_\ell^{(0)}$ für $p \in \{1, \dots, q\}$, so folgt aus (5.52) und (5.54)

$$\begin{aligned} \|c_\ell^{(k,p-1)}\|_V &= \|\Pi_\ell^{(k)} T c_\ell^{(k,p-2)}\|_{L^2(\Gamma)} \leq \|T c_\ell^{(k,p-2)}\|_{L^2(\Gamma)} \leq \tilde{C}_T \|c_\ell^{(k,p-2)}\|_{L^2(\Gamma)} \\ &\leq \tilde{C}_T^{p-1} \|e_\ell^{(k)}\|_{L^2(\Gamma)} = \tilde{C}_T^{p-1} \|\Pi_\ell^{(k)} \varepsilon_\ell^{(0)}\|_{L^2(\Gamma)} \leq \tilde{C}_T^{p-1} \|\varepsilon_\ell^{(0)}\|_{L^2(\Gamma)}, \end{aligned}$$

und weiters

$$\|c_\ell^{(k,p)} - \hat{c}_\ell^{(k,p)}\|_V \leq C_1 \tilde{C}_T^p \|h_\ell^{(k)}\|_{L^\infty(\Gamma)}^{1/2} \|\varepsilon_\ell^{(0)}\|_{L^2(\Gamma)}.$$

Eingesetzt in Gleichung (5.56) erhalten wir für $p \in \{1, \dots, q\}$

$$\|c_\ell^{(k,p)} - \varkappa_\ell^{(p)}\|_V \leq C_1 \tilde{C}_T^p \|h_\ell^{(k)}\|_{L^\infty(\Gamma)}^{1/2} \|\varepsilon_\ell^{(0)}\|_{L^2(\Gamma)} + C_T \|c_\ell^{(k,p-1)} - \varkappa_\ell^{(p-1)}\|_V$$

und damit aufgrund der Rekursion mit $c_\ell^{(k,0)} = e_\ell^{(k)}$ bzw. $\varkappa_\ell^{(0)} = \varepsilon_\ell^{(0)}$

$$\begin{aligned} \|c_\ell^{(k,p)} - \varkappa_\ell^{(p)}\|_V &\leq C_1 \|h_\ell^{(k)}\|_{L^\infty(\Gamma)}^{1/2} \sum_{\nu=1}^p \left(\tilde{C}_T^\nu \|\varepsilon_\ell^{(0)}\|_{L^2(\Gamma)} \right) + C_T^p \|e_\ell^{(k)} - \varepsilon_\ell^{(0)}\|_V \\ &= C_2 \|h_\ell^{(k)}\|_{L^\infty(\Gamma)}^{1/2} \|\varepsilon_\ell^{(0)}\|_{L^2(\Gamma)} + C_T^p \|e_\ell^{(k)} - \varepsilon_\ell^{(0)}\|_V \end{aligned}$$

mit $C_2 := C_1 \sum_{\nu=1}^p \tilde{C}_T^\nu$ abhängig von p . Setzen wir dies in (5.55) ein, so folgt

$$\begin{aligned} \|e_\ell^{(k,q)} - \varepsilon_\ell^{(q)}\|_V &\leq \sum_{\nu=0}^q \left(C_2 \|h_\ell^{(k)}\|_{L^\infty(\Gamma)}^{1/2} \|\varepsilon_\ell^{(0)}\|_{L^2(\Gamma)} + C_T^\nu \|e_\ell^{(k)} - \varepsilon_\ell^{(0)}\|_V \right) \\ &= (q+1) C_2 \|h_\ell^{(k)}\|_{L^\infty(\Gamma)}^{1/2} \|\varepsilon_\ell^{(0)}\|_{L^2(\Gamma)} + \|e_\ell^{(k)} - \varepsilon_\ell^{(0)}\|_V \sum_{\nu=0}^q C_T^\nu \\ &\leq (q+1) C_2 \|h_\ell^{(k)}\|_{L^\infty(\Gamma)}^{1/2} \|\varepsilon_\ell^{(0)}\|_{L^2(\Gamma)} + \frac{1}{1-C_T} \|e_\ell^{(k)} - \varepsilon_\ell^{(0)}\|_V \end{aligned}$$

Wenden wir nun (5.53) auf $\|e_\ell^{(k)} - \varepsilon_\ell^{(0)}\|_V = \|\Pi_\ell^{(k)} \varepsilon_\ell^{(0)} - \varepsilon_\ell^{(0)}\|_V$ an, erhalten wir

$$\|e_\ell^{(k,q)} - \varepsilon_\ell^{(q)}\|_V \leq C (q+1) \|h_\ell^{(k)}\|_{L^\infty(\Gamma)}^{1/2} \|\varepsilon_\ell^{(0)}\|_{L^2(\Gamma)} \quad (5.57)$$

für ein $C > 0$ abhängig von $q \in \mathbb{N}$.

Für den Fall $q = 0$ folgt mit $e_\ell^{(k)} = \Pi_\ell^{(k)} \varepsilon_\ell^{(0)}$ wegen (5.53) sofort

$$\|e_\ell^{(k)} - \varepsilon_\ell^{(0)}\|_V \leq C \|h_\ell^{(k)}\|_{L^\infty(\Gamma)}^{1/2} \|\varepsilon_\ell^{(0)}\|_{L^2(\Gamma)}$$

mit der Konstanten $C := C_1$, sodass (5.57) auch für $q = 0$ gilt.

(iv) Wir wenden nun die rechte Ungleichung von (5.49) an und erhalten mit (5.57)

$$\begin{aligned} \|\varepsilon_\ell\|_V &\leq \frac{1}{1-C_T^{q+1}} \|\varepsilon_\ell^{(q)}\|_V \leq \frac{1}{1-C_T^{q+1}} \left(\|e_\ell^{(k,q)}\|_V + \|e_\ell^{(k,q)} - \varepsilon_\ell^{(q)}\|_V \right) \\ &\leq \frac{1}{1-C_T^{q+1}} \left(\|e_\ell^{(k,q)}\|_V + C (q+1) \|h_\ell^{(k)}\|_{L^\infty(\Gamma)}^{1/2} \|\varepsilon_\ell^{(0)}\|_{L^2(\Gamma)} \right) \end{aligned}$$

und damit die erste Abschätzung der Behauptung. Für die zweite Abschätzung verwenden wir die linke Ungleichung von (5.49) und (5.57). Folglich gilt

$$\begin{aligned} \frac{1}{1+C_T^{q+1}} \|e_\ell^{(k,q)}\|_V &\leq \frac{1}{1+C_T^{q+1}} \left(\|\varepsilon_\ell^{(q)}\|_V + \|e_\ell^{(k,q)} - \varepsilon_\ell^{(q)}\|_V \right) \\ &\leq \|\varepsilon_\ell\|_V + \frac{1}{1+C_T^{q+1}} \left(C (q+1) \|h_\ell^{(k)}\|_{L^\infty(\Gamma)}^{1/2} \|\varepsilon_\ell^{(0)}\|_{L^2(\Gamma)} \right). \end{aligned}$$

Nach Umformen folgt die zweite Ungleichung.

- (v) Für hinreichend kleines $\|h_\ell^{(k)}\|_{L^\infty(\Gamma)}$ ist daher $\|e_\ell^{(k,q)}\|_V$ effizient und zuverlässig bzgl. $\|\phi_\ell - \Phi_\ell\|_V$. Aus der Äquivalenz (5.19) folgt dann die Effizienz und Zuverlässigkeit bis auf Datenoszillationen bzgl. des Fehlers $\|\phi - \Phi_\ell\|_V$. $\square$

Lemma 5.17 garantiert uns zwar die Effizienz und Zuverlässigkeit des Fehlerschätzers $\|e_\ell^{(k,q)}\|_V$ (bis auf Datenoszillationen) für ein hinreichend kleines $\|h_\ell^{(k)}\|_{L^\infty(\Gamma)} < \|h_\ell\|_{L^\infty(\Gamma)}$, doch es liefert uns keine Aussage „wie fein“ das notwendige Netz $\mathcal{E}_\ell^{(k)}$ mit $k > 0$ tatsächlich sein muss.

Für die Implementierung der einzelnen Projektionen sei $\mathcal{E}_\ell = \{E_1, \dots, E_N\}$ und $\mathcal{E}_\ell^{(k)}$ eine beliebige k -fache Verfeinerung von $\mathcal{E}_\ell$, d.h. es gilt $\mathcal{E}_\ell^{(k)} = \{e_{j,m} : j \in \{1, \dots, N\}, m \in \{1, \dots, 2^k\}\}$ mit

$$E_j = \bigcup_{m=1}^{2^k} e_{j,m} \quad \text{für alle } E_j \in \mathcal{E}_\ell$$

und

$$\Gamma = \bigcup_{j=1}^N E_j = \bigcup_{j=1}^N \bigcup_{m=1}^{2^k} e_{j,m}. \quad (5.58)$$

Mit der Definition (5.39) von $\tilde{\phi}_\ell$, der Definition (3.29) der L^2 -Projektion $\Pi_\ell^{(k)}$ und der Referenzparametrisierung $\gamma_{j,m} : [-1, 1] \rightarrow e_{j,m}$ aus (2.1) erhalten wir auf jedem Element $e_{j,m} \in \mathcal{E}_\ell^{(k)}$

$$\begin{aligned} e_\ell^{(k)}|_{e_{j,m}} &= \Pi_\ell^{(k)}(\tilde{\phi}_\ell - \Phi_\ell)|_{e_{j,m}} = \Pi_\ell^{(k)}((1/2 + K')\Phi_\ell + WG_\ell - \Phi_\ell)|_{e_{j,m}} \\ &= \Pi_\ell^{(k)}((K' - 1/2)\Phi_\ell + WG_\ell)|_{e_{j,m}} = \frac{1}{\text{length}(e_{j,m})} \int_{e_{j,m}} (K' - 1/2)\Phi_\ell + WG_\ell d\Gamma \\ &= \frac{1}{2} \int_{-1}^1 ((K' - 1/2)\Phi_\ell + WG_\ell) \circ \gamma_{j,m}(s) ds. \end{aligned}$$

Wir verwenden für obige rechte Seite eine klassische n -Punkt Gaußquadratur mit Exaktheitsgrad $2n - 1$ auf $[-1, 1]$ auf allen Elementen $e_{j,m} \in \mathcal{E}_\ell^{(k)}$. Wir schreiben sie in allgemeiner Form an, um sie für beide Implementierungen der in den folgenden Abschnitten 5.12.1 und 5.12.2 definierten Fehlerschätzer anwenden zu können. Mit den Gauß-Knoten $x_i, i = 1, \dots, n$ und den zugehörigen Gewichten ω_i approximieren wir $e_\ell^{(k)}$ aus (5.50) $\mathcal{E}_\ell^{(k)}$ -elementweise durch

$$\begin{aligned} e_\ell^{(k)}|_{e_{j,m}} &= \frac{1}{2} \int_{-1}^1 ((K' - 1/2)\Phi_\ell + WG_\ell) \circ \gamma_{j,m}(s) ds \\ &\approx \frac{1}{2} \sum_{i=1}^n \omega_i ((K' - 1/2)\Phi_\ell + WG_\ell) \circ \gamma_{j,m}(x_i), \end{aligned}$$

d.h. mit (5.58) gilt mit den Basisfunktionen $\chi_{j,m}$ auf $e_{j,m}$ für $j = 1, \dots, N, m = 1, \dots, 2^k$ von $\mathcal{P}^0(\mathcal{E}_\ell^{(k)})$

$$e_\ell^{(k)} \approx \tilde{e}_\ell^{(k)} := \sum_{j=1}^N \sum_{m=1}^{2^k} y_{j,m}^{(k)} \chi_{j,m} \quad (5.59)$$

mit

$$y_{j,m}^{(k)} := \frac{1}{2} \sum_{i=1}^n \omega_i ((K' - 1/2)\Phi_\ell + WG_\ell) \circ \gamma_{j,m}(x_i). \quad (5.60)$$

Für die Approximation von $e_\ell^{(k,q)} = \sum_{\nu=0}^q (\Pi_\ell^{(k)} T)^\nu e_\ell^{(k)}$, $q > 0$, gilt es für die weiteren Summanden in der Neumann-Summe Größen der Form

$$\Pi_\ell^{(k)} T \Psi_\ell^{(k)} \quad \text{mit } \Psi_\ell^{(k)} \in \mathcal{P}^0(\mathcal{E}_\ell^{(k)})$$

zu betrachten. Mit dem Operator $T = 1/2 + K'$ und (5.58) erhalten wir für alle $e_{j,m} \in \mathcal{E}_\ell^{(k)}$

$$\Pi_\ell^{(k)} T \Psi_\ell^{(k)}|_{e_{j,m}} = \Pi_\ell^{(k)} (1/2 + K') \Psi_\ell^{(k)}|_{e_{j,m}} = \frac{1}{2} \Pi_\ell^{(k)} \Psi_\ell^{(k)}|_{e_{j,m}} + \Pi_\ell^{(k)} K' \Psi_\ell^{(k)}|_{e_{j,m}}.$$

Per Definition (3.29) der L^2 -Projektion gilt $\Pi_\ell^{(k)} \Psi_\ell^{(k)}|_{e_{j,m}} = \Psi_\ell^{(k)}|_{e_{j,m}}$. Anwenden der Referenzparametrisierung $\gamma_{j,m} : [-1, 1] \rightarrow e_{j,m}$ aus (2.1) führt auf

$$\begin{aligned} \Pi_\ell^{(k)} T \Psi_\ell^{(k)}|_{e_{j,m}} &= \frac{1}{2} \Psi_\ell^{(k)}|_{e_{j,m}} + \frac{1}{\text{length}(e_{j,m})} \int_{e_{j,m}} K' \Psi_\ell^{(k)} d\Gamma \\ &= \frac{1}{2} \Psi_\ell^{(k)}|_{e_{j,m}} + \frac{1}{2} \int_{-1}^1 (K' \Psi_\ell^{(k)}) \circ \gamma_{j,m}(s) ds \\ &= \frac{1}{2} \left(\Psi_\ell^{(k)}|_{e_{j,m}} + \int_{-1}^1 (K' \Psi_\ell^{(k)}) \circ \gamma_{j,m}(s) ds \right), \end{aligned}$$

was wieder mittels obiger Gaußquadratur berechnet werden kann, d.h. wir approximieren

$$\Pi_\ell^{(k)} T \Psi_\ell^{(k)}|_{e_{j,m}} \approx \frac{1}{2} \left(\Psi_\ell^{(k)}|_{e_{j,m}} + \sum_{i=1}^n \omega_i (K' \Psi_\ell^{(k)}) \circ \gamma_{j,m}(x_i) \right).$$

Damit definieren wir $\mathcal{E}_\ell^{(k)}$ -elementweise, bezugnehmend auf die Iterationsvorschrift (5.51), $\tilde{c}_\ell^{(k,p)} \in \mathcal{P}^0(\mathcal{E}_\ell^{(k)})$ für $p \in \{1, \dots, q\}$ iterativ durch

$$c_\ell^{(k,p)}|_{e_{j,m}} = \Pi_\ell^{(k)} T c_k^{(p-1)}|_{e_{j,m}} \approx \frac{1}{2} \left(\tilde{c}_k^{(p-1)}|_{e_{j,m}} + \sum_{i=1}^n \omega_i (K' \tilde{c}_k^{(p-1)}) \circ \gamma_{j,m}(x_i) \right) =: \tilde{c}_\ell^{(k,p)}|_{e_{j,m}}$$

für alle $e_{j,m} \in \mathcal{E}_\ell^{(k)}$ mit $\tilde{c}_k^{(0)} := \tilde{e}_\ell^{(k)}$ und zugehörigen Koeffizienten aus (5.60). Zusammenfassend erhalten wir für die Implementierung die Iteration

$$\tilde{e}_\ell^{(k,p)} = \tilde{e}_\ell^{(k,p-1)} + \tilde{c}_\ell^{(k,p)} \quad \text{mit} \quad \tilde{c}_\ell^{(k,p)}|_{e_{j,m}} = \frac{1}{2} \left(c_k^{(p-1)}|_{e_{j,m}} + \sum_{i=1}^n \omega_i (K' c_k^{(p-1)}) \circ \gamma_{j,m}(x_i) \right) \quad (5.61)$$

für $p \in \{1, \dots, q\}$, wobei

$$\tilde{c}_k^{(0)} = \tilde{e}_\ell^{(k)} = \sum_{j=1}^N \sum_{m=1}^{2^k} y_{j,m}^{(k)} \chi_{j,m}$$

mit

$$y_{j,m}^{(k)} = \frac{1}{2} \sum_{i=1}^n \omega_i ((K' - 1/2)\Phi_\ell + WG_\ell) \circ \gamma_{j,m}(x_i).$$

5.12.1 Steinbach-Schätzer über Einfachschichtpotential V

Als erste Möglichkeit der Implementierung ist die nach Lemma 5.17 angeregte Definition eines Fehlerschätzers $\|e_\ell^{(k,q)}\|_V$. Mit $e_\ell^{(k,q)}$ aus (5.50) definieren wir nach [16, Abschnitt 3] die lokalen Fehlerindikatoren³

$$\sigma_{\ell,V}^{(k,q)}(E_j) := \left(|\langle V e_\ell^{(k,q)}, e_\ell^{(k,q)} \rangle_{E_j}| \right)^{1/2} \quad (5.62)$$

und den **Steinbach-Schätzer über das Einfachschichtpotential**

$$\sigma_{\ell,V}^{(k,q)} := \|e_\ell^{(k,q)}\|_V = \left(\langle V e_\ell^{(k,q)}, e_\ell^{(k,q)} \rangle_\Gamma \right)^{1/2}. \quad (5.63)$$

Demnach gilt

$$(\sigma_{\ell,V}^{(k,q)})^2 = \langle V e_\ell^{(k,q)}, e_\ell^{(k,q)} \rangle_\Gamma = \sum_{j=1}^N \langle V e_\ell^{(k,q)}, e_\ell^{(k,q)} \rangle_{E_j} \leq \sum_{j=1}^N |\langle V e_\ell^{(k,q)}, e_\ell^{(k,q)} \rangle_{E_j}|$$

und folglich mit (5.62)

$$(\sigma_{\ell,V}^{(k,q)})^2 \leq \sum_{j=1}^N (\sigma_{\ell,V}^{(k,q)}(E_j))^2.$$

Mit Hilfe von Lemma 5.17 erhalten wir

Satz 5.18 Sei $\mathcal{E}_\ell^{(k)}$ mit $k > 0$ eine k -fache uniforme Verfeinerung der Triangulierung $\mathcal{E}_\ell$. Dann ist der Steinbach-Schätzer

$$\sigma_{\ell,V}^{(k,q)} = \left(\langle V e_\ell^{(k,q)}, e_\ell^{(k,q)} \rangle_\Gamma \right)^{1/2}$$

effizient und zuverlässig bis auf Datenoszillationen für $\|h_\ell^{(k)}\|_{L^\infty(\Gamma)} < \|h_\ell\|_{L^\infty(\Gamma)}$ hinreichend klein. Insbesondere gilt in diesem Fall bei uniformen Netzen mit Netzweite h für hinreichend glatte Dirichlet- und Neumann-Daten

$$\sigma_{\ell,V}^{(k,q)} = \mathcal{O}(h^{3/2})$$

aufgrund von Satz 3.6. □

Mit obigen Überlegungen definieren wir für die Implementierung die approximierten lokalen Steinbach-Fehlerindikatoren

$$\tilde{\sigma}_{\ell,V}^{(k,q)}(E_j) := \left(|\langle V \tilde{e}_\ell^{(k,q)}, \tilde{e}_\ell^{(k,q)} \rangle_{E_j}| \right)^{1/2}$$

³Die symmetrische Bilinearform $\langle \cdot, \cdot \rangle_V$ ist auf den einzelnen Elementen $E \in \mathcal{E}_\ell$ nicht mehr notwendigerweise definit.

und den approximierten Steinbach-Schätzer

$$\tilde{\sigma}_{\ell,V}^{(k,q)} := \|\tilde{e}_{\ell}^{(k,q)}\|_V = \left(\langle V\tilde{e}_{\ell}^{(k,q)}, \tilde{e}_{\ell}^{(k,q)} \rangle_{\Gamma} \right)^{1/2}$$

mit $\tilde{e}_{\ell}^{(k,q)}$ aus (5.61), wobei wieder

$$(\tilde{\sigma}_{\ell,V}^{(k,q)})^2 \leq \sum_{j=1}^N (\tilde{\sigma}_{\ell,V}^{(k,q)}(E_j))^2$$

gilt.

5.12.2 Steinbach-Schätzer über Multilevel-Operator

Eine weitere Möglichkeit zur Implementierung eines Fehlerschätzers auf Basis der Größe $e_{\ell}^{(k,q)}$ aus (5.50) wird in [17] vorgestellt und führt über den Multilevel-Operator. Sei dazu wieder $\Pi_{\ell}^{(k)}$ die L^2 -Projektion auf die k -fache uniforme Verfeinerung $\mathcal{E}_{\ell}^{(k)}$ von $\mathcal{E}_{\ell}$ und $h_{\ell}^{(k)} \in \mathcal{P}^0(\mathcal{E}_{\ell}^{(k)})$ die lokale Netzweite von $\mathcal{E}_{\ell}^{(k)}$.

Wir definieren den **Multilevel-Operator** für $s \in \mathbb{R}$, $k > 0$ und $\Psi_{\ell}^{(k)} \in \mathcal{P}^0(\mathcal{E}_{\ell}^{(k)})$ durch

$$A_{\ell}^{(k,s)} \Psi_{\ell}^{(k)} := \sum_{i=1}^k (h_{\ell}^{(i)})^{-2s} (\Pi_{\ell}^{(i)} - \Pi_{\ell}^{(i-1)}) \Psi_{\ell}^{(k)} \in \mathcal{P}^0(\mathcal{E}_{\ell}^{(k)}) \quad (5.64)$$

mit $\mathcal{E}_{\ell}^{(0)} = \mathcal{E}_{\ell}$. Diese Definition des Multilevel-Operators ist damit auch für die Berechnung auf adaptiven Netze mit lokaler Netzweite $h_{\ell}^{(i)} \in \mathcal{P}^0(\mathcal{E}_{\ell}^{(i)})$ verwendbar. Sei dazu $E \in \mathcal{E}_{\ell}$, dann gilt $E = e_1 \cup \dots \cup e_{2^i}$ mit $e_j \in \mathcal{E}_{\ell}^{(i)}$ für $i > 0$. Folglich gilt auf einem Element $e_j \in \mathcal{E}_{\ell}^{(i)}$ wegen $h_{\ell}^{(i)}|_{e_j} = 2^{-i} h_{\ell}|_E = 2^{-i} h_{\ell}|_{e_j}$

$$A_{\ell}^{(k,s)} \Psi_{\ell}^{(k)}|_{e_j} = \sum_{i=1}^k 2^{2si} h_{\ell}|_{e_j}^{-2s} (\Pi_{\ell}^{(i)}|_{e_j} - \Pi_{\ell}^{(i-1)}|_{e_j}) \Psi_{\ell}^{(k)}.$$

Im Fall uniformer Netze gilt entsprechend $h_{\ell}^{(i)}|_{e_j} = 2^{-i} h$ mit h als uniformer Netzweite von $\mathcal{E}_{\ell}$. Für die weiteren Überlegungen benötigen wir folgendes

Lemma 5.19 *Mit $v \in L^1(\Gamma)$ ist die L^2 -Orthogonalprojektion Π_{ℓ} auf $\mathcal{P}^0(\mathcal{E}_{\ell})$ durch (3.29), d.h.*

$$(\Pi_{\ell} v)|_E = \frac{1}{\text{length}(E)} \int_E v \, d\Gamma \quad \text{für alle } E \in \mathcal{E}_{\ell},$$

definiert. Dann gilt auf dem Rand Γ für alle $v \in L^1(\Gamma)$:

$$(i) \quad \Pi_k \Pi_{\ell} v = \Pi_{\ell} \Pi_k v = \Pi_{\ell} v \quad \text{für } \ell \leq k$$

$$(ii) \quad (\Pi_k - \Pi_{k-1}) \Pi_{\ell} v = 0 \quad \text{für } \ell < k$$

- (iii) $(\Pi_k - \Pi_{k-1})^2 v = (\Pi_k - \Pi_{k-1})v$
 (iv) $(\Pi_k - \Pi_{k-1})(\Pi_\ell - \Pi_{\ell-1})v = 0$ für $\ell \neq k$

BEWEIS:

- (i) Sei $E \in \mathcal{E}_\ell$. Durch (beliebige) Verfeinerung entsteht dann eine Triangulierung $\mathcal{E}_k$, d.h. $k \geq \ell$. Mit $e_i \in \mathcal{E}_k$ sei nun E darstellbar in der Form $E = e_1 \cup \dots \cup e_n$ für ein n abhängig von k . Dann gilt mit $v \in L^2$ und $i \in \{1, \dots, n\}$ beliebig

$$\Pi_k \Pi_\ell v|_{e_i} = \frac{1}{\text{length}(e_i)} \int_{e_i} \Pi_\ell v \, d\Gamma = \Pi_\ell v|_E \frac{1}{\text{length}(e_i)} \int_{e_i} d\Gamma = \Pi_\ell v|_E,$$

für alle $E \in \mathcal{E}_\ell$, d.h. es gilt

$$\Pi_k \Pi_\ell v = \Pi_\ell v \quad \text{auf } \Gamma.$$

Andererseits gilt

$$\begin{aligned} \Pi_\ell \Pi_k v|_E &= \frac{1}{\text{length}(E)} \int_E \Pi_k v \, d\Gamma = \frac{1}{\text{length}(E)} \left(\sum_{i=1}^n \int_{e_i} \Pi_k v \, d\Gamma \right) \\ &= \frac{1}{\text{length}(E)} \left(\sum_{i=1}^n \Pi_k v|_{e_i} \int_{e_i} d\Gamma \right) = \frac{1}{\text{length}(E)} \left(\sum_{i=1}^n \int_{e_i} v \, d\Gamma \right) \\ &= \frac{1}{\text{length}(E)} \int_E v \, d\Gamma = \Pi_\ell v|_E, \end{aligned}$$

womit auch

$$\Pi_\ell \Pi_k v = \Pi_\ell v \quad \text{auf } \Gamma$$

gilt.

- (ii) Für die restlichen Beweise verzichten wir aus Gründen der Übersichtlichkeit auf das Anschreiben der Funktion $v \in L^1(\Gamma)$ und erhalten sodann für $\ell < k$ mit der Eigenschaft (i)

$$(\Pi_k - \Pi_{k-1})\Pi_\ell = \Pi_k \Pi_\ell - \Pi_{k-1} \Pi_\ell = \Pi_\ell - \Pi_\ell = 0$$

- (iii) Anwenden von (i) führt zu

$$\begin{aligned} (\Pi_k - \Pi_{k-1})^2 &= (\Pi_k - \Pi_{k-1})(\Pi_k - \Pi_{k-1}) = \Pi_k \Pi_k - \Pi_{k-1} \Pi_k - \Pi_k \Pi_{k-1} + \Pi_{k-1} \Pi_{k-1} \\ &= \Pi_k - \Pi_{k-1} - \Pi_{k-1} + \Pi_{k-1} = \Pi_k - \Pi_{k-1} \end{aligned}$$

- (iv) Es gilt

$$(\Pi_k - \Pi_{k-1})(\Pi_\ell - \Pi_{\ell-1}) = \Pi_k \Pi_\ell - \Pi_{k-1} \Pi_\ell - \Pi_k \Pi_{\ell-1} + \Pi_{k-1} \Pi_{\ell-1}$$

mit $k \neq \ell$. Im Falle $k > \ell$ gilt mit (i)

$$(\Pi_k - \Pi_{k-1})(\Pi_\ell - \Pi_{\ell-1}) = \Pi_\ell - \Pi_\ell - \Pi_{\ell+1} + \Pi_{\ell+1} = 0$$

und im Falle $k < \ell$

$$(\Pi_k - \Pi_{k-1})(\Pi_\ell - \Pi_{\ell-1}) = \Pi_k - \Pi_k - \Pi_{k+1} + \Pi_{k+1} = 0$$

womit auch die Behauptung (iv) bewiesen ist. $\square$

Mit diesem Lemma können wir eine wichtige Eigenschaft des Multilevel-Operators beweisen:

Lemma 5.20 *Sei die Triangulierung $\mathcal{E}_\ell^{(k)}$ mit $k > 0$ eine k -fache uniforme Verfeinerung von $\mathcal{E}_\ell = \{E_1, \dots, E_N\}$ und $A_\ell^{(k,s)}$ der Multilevel-Operator aus (5.64). Dann gilt für $j \in \{1, \dots, N\}$ und $s \in \mathbb{R}$ beliebig*

$$\langle A_\ell^{(k,s/2)} \Psi_\ell^{(k)}, A_\ell^{(k,s/2)} \Psi_\ell^{(k)} \rangle_{L^2(E_j)} = \langle A_\ell^{(k,s)} \Psi_\ell^{(k)}, \Psi_\ell^{(k)} \rangle_{L^2(E_j)} \quad (5.65)$$

für alle $\Psi_\ell^{(k)} \in \mathcal{P}^0(\mathcal{E}_\ell^{(k)})$.

BEWEIS: Wir zeigen für $u, v \in \mathbb{N}_0$ zuerst die Gültigkeit von

$$\langle \Pi_\ell^{(u)} \Psi_\ell^{(k)}, \Pi_\ell^{(v)} \Psi_\ell^{(k)} \rangle_{L^2(E_j)} = \langle \Pi_\ell^{(u)} \Pi_\ell^{(v)} \Psi_\ell^{(k)}, \Psi_\ell^{(k)} \rangle_{L^2(E_j)} \quad (5.66)$$

für alle $E_j \in \mathcal{E}_\ell$. Sei zuerst $u \geq v$. Da $\mathcal{E}_\ell^{(u)}$ eine u -fache uniforme Verfeinerung von $\mathcal{E}_\ell$ ist, gilt

$$E_j = \bigcup_{m=1}^{2^u} e_{j,m}$$

mit $e_{j,m} \in \mathcal{E}_\ell^{(u)}$. Daher gilt

$$\langle \Pi_\ell^{(u)} \Pi_\ell^{(v)} \Psi_\ell^{(k)}, \Psi_\ell^{(k)} \rangle_{L^2(E_j)} = \int_{E_j} \Pi_\ell^{(u)} \Pi_\ell^{(v)} \Psi_\ell^{(k)} \Psi_\ell^{(k)} d\Gamma = \sum_{m=1}^{2^u} \int_{e_{j,m}} \Pi_\ell^{(u)} \Pi_\ell^{(v)} \Psi_\ell^{(k)} \Psi_\ell^{(k)} d\Gamma.$$

Wegen $u \geq v$ gilt mit Lemma 5.19 (i) $\Pi_\ell^{(u)} \Pi_\ell^{(v)} \Psi_\ell^{(k)} = \Pi_\ell^{(v)} \Psi_\ell^{(k)} = \text{const}$ auf $e_{j,m} \in \mathcal{E}_\ell^{(u)}$. Wir erhalten weiters mit der Definition (3.29) der L^2 -Projektion $\Pi_\ell^{(u)}$

$$\begin{aligned} \langle \Pi_\ell^{(u)} \Pi_\ell^{(v)} \Psi_\ell^{(k)}, \Psi_\ell^{(k)} \rangle_{L^2(E_j)} &= \sum_{m=1}^{2^u} \Pi_\ell^{(v)} \Psi_\ell^{(k)}|_{e_{j,m}} \int_{e_{j,m}} \Psi_\ell^{(k)} d\Gamma \\ &= \sum_{m=1}^{2^u} \Pi_\ell^{(v)} \Psi_\ell^{(k)}|_{e_{j,m}} \Pi_\ell^{(u)} \Psi_\ell^{(k)}|_{e_{j,m}} \text{length}(e_{j,m}) \\ &= \sum_{m=1}^{2^u} \int_{e_{j,m}} \Pi_\ell^{(v)} \Psi_\ell^{(k)} \Pi_\ell^{(u)} \Psi_\ell^{(k)} d\Gamma = \int_{E_j} \Pi_\ell^{(v)} \Psi_\ell^{(k)} \Pi_\ell^{(u)} \Psi_\ell^{(k)} d\Gamma \\ &= \langle \Pi_\ell^{(u)} \Psi_\ell^{(k)}, \Pi_\ell^{(v)} \Psi_\ell^{(k)} \rangle_{L^2(E_j)}. \end{aligned}$$

Analog zeigt man den Fall $v \geq u$, womit (5.66) bewiesen ist. Nun gilt weiter mit $h_\ell^{(u)}|_{e_{j,m}} = 2^{-u} h_\ell|_{E_j}$ für alle $e_{j,m} \subseteq E_j$

$$\begin{aligned} \langle A_\ell^{(k,s/2)} \Psi_\ell^{(k)}, A_\ell^{(k,s/2)} \Psi_\ell^{(k)} \rangle_{L^2(E_j)} &= \int_{E_j} A_\ell^{(k,s/2)} \Psi_\ell^{(k)} A_\ell^{(k,s/2)} \Psi_\ell^{(k)} d\Gamma \\ &= \sum_{u,v=1}^k \left(\int_{E_j} (h_\ell^{(u)})^{-s} (h_\ell^{(v)})^{-s} (\Pi_\ell^{(u)} - \Pi_\ell^{(u-1)}) \Psi_\ell^{(k)} (\Pi_\ell^{(v)} - \Pi_\ell^{(v-1)}) \Psi_\ell^{(k)} d\Gamma \right) \\ &= \sum_{u,v=1}^k \left(2^{s(u+v)} (h_\ell|_{E_j})^2 \int_{E_j} (\Pi_\ell^{(u)} - \Pi_\ell^{(u-1)}) \Psi_\ell^{(k)} (\Pi_\ell^{(v)} - \Pi_\ell^{(v-1)}) \Psi_\ell^{(k)} d\Gamma \right). \end{aligned}$$

Ausmultiplizieren, Anwendung von (5.66) und anschließendem Zusammenfassen der Terme führt auf

$$\begin{aligned} \langle A_\ell^{(k,s/2)} \Psi_\ell^{(k)}, A_\ell^{(k,s/2)} \Psi_\ell^{(k)} \rangle_{L^2(E_j)} &= \sum_{u,v=1}^k \left(2^{s(u+v)} (h_\ell|_{E_j})^2 \right. \\ &\quad \left. \int_{E_j} (\Pi_\ell^{(u)} - \Pi_\ell^{(u-1)}) (\Pi_\ell^{(v)} - \Pi_\ell^{(v-1)}) \Psi_\ell^{(k)} \Psi_\ell^{(k)} d\Gamma \right). \end{aligned}$$

Mit Lemma 5.19 (iii) und (iv) folgt dann

$$\begin{aligned} \langle A_\ell^{(k,s/2)} \Psi_\ell^{(k)}, A_\ell^{(k,s/2)} \Psi_\ell^{(k)} \rangle_{L^2(E_j)} &= \sum_{u=1}^k 2^{2su} (h_\ell|_{E_j})^2 \int_{E_j} (\Pi_\ell^{(u)} - \Pi_\ell^{(u-1)}) \Psi_\ell^{(k)} \Psi_\ell^{(k)} d\Gamma \\ &= \int_{E_j} \sum_{u=1}^k (h_\ell^{(u)})^{-2s} (\Pi_\ell^{(u)} - \Pi_\ell^{(u-1)}) \Psi_\ell^{(k)} \Psi_\ell^{(k)} d\Gamma \\ &= \langle A_\ell^{(k,s)} \Psi_\ell^{(k)}, \Psi_\ell^{(k)} \rangle_{L^2(E_j)} \end{aligned}$$

und damit die Behauptung. $\square$

Aus [13, Theorem 2] zitieren wir die auf uniformen Netzen gültige Äquivalenz der Norm $\|A_\ell^{(k,-1/4)} \Psi_\ell^{(k)}\|_{L^2(\Gamma)}$ auf $H^{-1/2}(\Gamma)$ für alle $\Psi_\ell^{(k)} \in \mathcal{P}^0(\mathcal{E}_\ell^{(k)})$ und $k \in \mathbb{N}$. Insbesondere ist sie daher bei uniformer Netzverfeinerung äquivalent zur Energienorm $\|\cdot\|_V$. Mit $e_\ell^{(k,q)}$ aus (5.50) für $k > 0$, $q \in \mathbb{N}_0$ und Lemma 5.20 definieren wir die lokalen Fehleranteile

$$\sigma_{\ell,A}^{(k,q)}(E_j) := \|A_\ell^{(k,-1/4)} e_\ell^{(k,q)}\|_{L^2(E_j)} = \left(\langle A_\ell^{(k,-1/2)} e_\ell^{(k,q)}, e_\ell^{(k,q)} \rangle_{L^2(E_j)} \right)^{1/2}$$

für den **Steinbach-Schätzer über den Multilevel-Operator**

$$\sigma_{\ell,A}^{(k,q)} := \left(\sum_{j=1}^N (\sigma_{\ell,A}^{(k,q)}(E_j))^2 \right)^{1/2} = \left(\langle A_\ell^{(k,-1/2)} e_\ell^{(k,q)}, e_\ell^{(k,q)} \rangle_{L^2(\Gamma)} \right)^{1/2}. \quad (5.67)$$

Damit sind die Fehlerschätzer $\sigma_{\ell,A}^{(k,q)}$ und $\sigma_{\ell,V}^{(k,q)}$ aus (5.63) auf uniformen Netzen äquivalent. Folglich gilt mit Lemma 5.17

Satz 5.21 Sei $\mathcal{E}_\ell^{(k)}$ mit $k > 0$ eine k -fache uniforme Verfeinerung der Triangulierung $\mathcal{E}_\ell$. Dann ist der Steinbach-Schätzer

$$\sigma_{\ell,A}^{(k,q)} = \left(\langle A_\ell^{(k,-1/2)} e_\ell^{(k,q)}, e_\ell^{(k,q)} \rangle_{L^2(\Gamma)} \right)^{1/2}$$

auf uniformen Netzen effizient und zuverlässig bis auf Datenoszillationen für hinreichend kleines $\|h_\ell^{(k)}\|_{L^\infty(\Gamma)} < \|h_\ell\|_{L^\infty(\Gamma)}$. Insbesondere gilt in diesem Fall mit Netzweite h für hinreichend glatte Dirichlet- und Neumann-Daten

$$\sigma_{\ell,A}^{(k,q)} = \mathcal{O}(h^{3/2})$$

aufgrund von Satz 3.6. □

Analog dem vorigen Abschnitt definieren wir für die Implementierung mit $\tilde{e}_\ell^{(k,q)} \approx e_\ell^{(k,q)}$ aus (5.61) und den approximierten lokalen Fehleranteilen

$$\tilde{\sigma}_{\ell,A}^{(k,q)}(E_j) := \|A_\ell^{(k,-1/4)} \tilde{e}_\ell^{(k,q)}\|_{L^2(E_j)} = \left(\langle A_\ell^{(k,-1/2)} \tilde{e}_\ell^{(k,q)}, \tilde{e}_\ell^{(k,q)} \rangle_{L^2(E_j)} \right)^{1/2},$$

den approximierten Fehlerschätzer

$$\tilde{\sigma}_{\ell,A}^{(k,q)} := \left(\sum_{j=1}^N (\tilde{\sigma}_{\ell,A}^{(k,q)}(E_j))^2 \right)^{1/2}.$$

5.13 Implementierung der Steinbach-Schätzer $\sigma_{\ell,V}$ und $\sigma_{\ell,A}$ in MATLAB

5.13.1 Steinbach-Schätzer $\sigma_{\ell,V}$

Listing 9: computeEstSigmaV

```

1 function [ind,sigmaV] = computeEstSigmaV(coordinates,elements,x,uDh)
2 %*** set parameter for k-times uniform mesh-refinement of \ell-1-mesh
3 k = 1;
4
5 %*** Gaussian quadrature on [-1,1] with 5 nodes and exactness 9
6 quad_nodes = [ -sqrt(245+14*sqrt(70)) ...
7               -sqrt(245-14*sqrt(70)) ...
8               0 ...
9               sqrt(245-14*sqrt(70)) ...
10              sqrt(245+14*sqrt(70)) ] / 21;
11 quad_weights = [ 161/450-13/900*sqrt(70); ...
12                 161/450+13/900*sqrt(70); ...
13                 128/225; ...
14                 161/450+13/900*sqrt(70); ...
15                 161/450-13/900*sqrt(70)];
16 nQ = length(quad_nodes);
17
```

```

18 %*** allocate memory
19 nE = size(elements,1);
20 e = zeros(2^k*nE,1);
21 ind = zeros(nE,1);
22 sons = zeros(nE,2^k);
23 sons(:,1) = [1:nE];
24
25 %*** k-times uniform mesh-refinement
26 coordinates_fine = coordinates;
27 elements_fine = elements;
28
29 for i=1:k
30     [coordinates_fine,elements_fine,father2son] ...
31         = refineBoundaryMesh(coordinates_fine,elements_fine);
32     for m=1:2^(k-i+1):2^k
33         sons(:,m+2^(k-i)) = father2son(sons(:,m),2);
34     end
35 end
36
37 %*** evaluate phih elementwise at evaluation points
38 x_sx = reshape(repmat(x,1,nQ)',nQ*nE,1);
39
40 %*** sx2element(i) returns the element number j such that sx(i,:) lies on Ej
41 sx2element = reshape(repmat((1:nE),nQ,1),nQ*nE,1);
42
43 for m=1:2^k
44     %*** build vector of evaluations points as (nI*nE x 2)-matrix
45     a = coordinates_fine(elements_fine(sons(:,m),1),:);
46     b = coordinates_fine(elements_fine(sons(:,m),2),:);
47     sx = reshape(a,2*nE,1)*(1-quad_nodes) + reshape(b,2*nE,1)*(1+quad_nodes);
48     sx = 0.5*reshape(sx',nQ*nE,2);
49
50     %*** compute vector of (squared) element-widths
51     h = sum((a-b).^2,2);
52
53     %*** compute outer normal vector
54     normal = b-a;
55     normal = [normal(:,2),-normal(:,1)]./repmat(sqrt(h),1,2);
56
57     %*** evaluate p_m = (K'-1/2)*phih + W*gh elementwise at quad_nodes
58     %*** in son-Element e_jm
59     p_m = evaluateKadj(coordinates,elements,x,sx,normal(sx2element,:)) ...
60         - 0.5*x_sx + evaluateW(coordinates,elements,uDh,sx,normal(sx2element,:));
61     p_m = reshape(p_m,nQ,nE)';
62
63     %*** compute coefficients of \widetilde{e}^{(k,0)}
64     e(sons(:,m)) = 0.5*p_m*quad_weights;
65 end
66
67 %*** compute squared local error indicators sigmaV(Ej)^2
68 V = buildV(coordinates_fine,elements_fine);
69 sons = reshape(sons',2^k*nE,1);
70 ind = reshape(e'*V(:,sons),2^k,nE).*reshape(e(sons),2^k,nE);
71 ind = abs(sum(ind,1))';
    
```

```

72
73 %*** compute squared error estimator sigmaV^2
74 sigmaV = e'*V*e;

```

In diesem Abschnitt berechnen wir den Fehlerschätzer

$$\tilde{\sigma}_{\ell,V}^{(k,q)} = \|\tilde{e}_{\ell}^{(k,q)}\|_V^2 = \left(\langle V\tilde{e}_{\ell}^{(k,q)}, \tilde{e}_{\ell}^{(k,q)} \rangle_{\Gamma} \right)^{1/2}$$

und die lokalen Fehlerindikatoren

$$\tilde{\sigma}_{\ell,V}^{(k,q)}(E_j) = \left(|\langle V\tilde{e}_{\ell}^{(k,q)}, \tilde{e}_{\ell}^{(k,q)} \rangle_{E_j}| \right)^{1/2}$$

mit $\tilde{e}_{\ell}^{(k,q)} \in \mathcal{P}^0(\mathcal{E}_{\ell}^{(k)})$ für $q = 0$. Leider erhalten wir durch Lemma 5.17 keine Aussage darüber wie fein das Netz $\mathcal{E}_{\ell}^{(k)}$ gegenüber $\mathcal{E}_{\ell}$ sein muss, damit wir einen effizienten und zuverlässigen Fehlerschätzer erhalten. Wir werden uns mit dieser Fragestellung im Zuge numerischer Simulationen im Abschnitt 6.2 beschäftigen.

Das Netz $\mathcal{E}_{\ell}^{(k)}$ wird durch k -fache uniforme Verfeinerung der Triangulierung $\mathcal{E}_{\ell}$ erzeugt. Wir gehen also wieder von der Darstellung (5.58), d.h.

$$\Gamma = \bigcup_{j=1}^N E_j = \bigcup_{j=1}^N \bigcup_{m=1}^{2^k} e_{j,m}$$

mit $E_j = \bigcup_{m=1}^{2^k} e_{j,m}$, aus. Mit den Basisfunktionen $\chi_{j,m}$ auf den Elementen $e_{j,m} \in \mathcal{E}_{\ell}^{(k)}$ gilt für $\tilde{e}_{\ell}^{(k)}$ aus (5.59) mit den Koeffizienten $y_{j,m} := y_{j,m}^{(k)}$ aus (5.60)

$$\tilde{e}_{\ell}^{(k)} = \sum_{j=1}^N \sum_{m=1}^{2^k} y_{j,m} \chi_{j,m},$$

wobei wir für die Berechnung dieser Koeffizienten eine 5-Punkt Gaußquadratur mit den Knoten

$$\left. \begin{aligned} x_1 &= -\frac{1}{21} \sqrt{245 + 14\sqrt{70}} \\ x_2 &= -\frac{1}{21} \sqrt{245 - 14\sqrt{70}} \\ x_3 &= 0 \\ x_4 &= \frac{1}{21} \sqrt{245 - 14\sqrt{70}} \\ x_5 &= \frac{1}{21} \sqrt{245 + 14\sqrt{70}} \end{aligned} \right\} \quad (5.68)$$

und den zugehörigen Gewichten

$$\left. \begin{aligned} \omega_1 &= \frac{161}{450} - \frac{13}{900}\sqrt{70} \\ \omega_2 &= \frac{161}{450} + \frac{13}{900}\sqrt{70} \\ \omega_3 &= \frac{128}{225} \\ \omega_4 &= \frac{161}{450} + \frac{13}{900}\sqrt{70} \\ \omega_5 &= \frac{161}{450} - \frac{13}{900}\sqrt{70} \end{aligned} \right\} \quad (5.69)$$

verwenden.

Für die Berechnung der Fehlerindikatoren erhalten wir wegen $\chi_{t,n} = 0$ auf E_j für $t \neq j$

$$\begin{aligned} (\tilde{\sigma}_{\ell,V}^{(k,0)}(E_j))^2 &= |\langle V\tilde{e}_\ell^{(k)}, \tilde{e}_\ell^{(k)} \rangle_{E_j}| = \left| \sum_{i,t=1}^N \sum_{m,n=1}^{2^k} y_{i,m} y_{t,n} \langle V\chi_{i,m}, \chi_{t,n} \rangle_{E_j} \right| \\ &= \left| \sum_{i=1}^N \sum_{m,n=1}^{2^k} y_{i,m} y_{j,n} \langle V\chi_{i,m}, \chi_{j,n} \rangle_{E_j} \right|. \end{aligned}$$

Mit der Definition (3.11) der Galerkin-Matrix $\mathbf{V}^{(k)}$ bzgl. der Triangulierung $\mathcal{E}_\ell^{(k)}$ gilt mit einer Indexfunktion $\iota : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$, die für $\iota(j, n)$ den n -ten Sohn $e_{j,n} \in \mathcal{E}_\ell^{(k)}$ des Elementes E_j liefert,

$$(\tilde{\sigma}_{\ell,V}^{(k,0)}(E_j))^2 = \left| \sum_{i=1}^N \sum_{m,n=1}^{2^k} y_{\iota(i,m)} y_{\iota(j,n)} \mathbf{V}_{\iota(j,n), \iota(i,m)}^{(k)} \right| = |\mathbf{y}^T \mathbf{V}^{(k)}(:, \iota(j, :)) \mathbf{y}(\iota(j, :))|, \quad (5.70)$$

wobei der Vektor $\mathbf{y} = (y_1, \dots, y_{2^k N}) \in \mathbb{R}^{2^k N}$ durch $y_{\iota(i,m)} := y_{i,m}$ gebildet wird.

Der Steinbach-Fehlerschätzer lässt sich dann mit obigen Überlegungen durch

$$(\tilde{\sigma}_{\ell,V}^{(k,0)})^2 = \langle V\tilde{e}_\ell^{(k)}, \tilde{e}_\ell^{(k)} \rangle_\Gamma = \sum_{i,j=1}^N \sum_{m,n=1}^{2^k} y_{i,m} y_{j,n} \langle V\chi_{i,m}, \chi_{j,n} \rangle_\Gamma = \mathbf{y}^T \mathbf{V}^{(k)} \mathbf{y} \quad (5.71)$$

berechnen.

Für die Implementierung setzen wir $k := 1$. Wir werden dies, ebenso wie die Wahl der Gaußquadratur, mit Hilfe von numerischen Simulationen in Abschnitt 6.2.1 begründen.

Listing 9 zeigt die Implementierung in MATLAB:

- Die Funktion übernimmt das Netz $\mathcal{E}_\ell$ in Form von **coordinates** und **elements**. Weiters werden auch der Koeffizientenvektor $\mathbf{x}$ der Galerkin-Lösung Φ_ℓ in Form von **x** und die Auswertungen der Dirichlet-Daten g an allen Knoten der Triangulierung als Spaltenvektor **u**dh eingelesen. Als Output werden die quadrierten lokalen Fehlerindikatoren $(\tilde{\sigma}_{\ell,V}^{(k,0)}(E))^2$ für alle $E \in \mathcal{E}_\ell$ als Spaltenvektor **ind** und der quadrierte Steinbach-Schätzer $(\tilde{\sigma}_{\ell,V}^{(k,0)})^2$ als **sigma**v zurückgegeben (Zeile 1).

- Zuerst wird der Parameter k für die k -fache uniforme Netzverfeinerung des Netzes $\mathcal{E}_\ell$ definiert (Zeile 3).
- Anschließend werden die Gaußknoten (5.68) und -gewichte (5.69) initialisiert und deren Anzahl in `nQ` geschrieben (Zeilen 6–16).
- Weiters wird der Speicher für die benötigten Arrays allokiert und die erste Spalte des $(\mathbf{nE} \times 2^k)$ -Arrays `sons` initialisiert. Dabei wird `sons` die oben beschriebene Funktion $\iota : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ realisieren (Zeilen 19–23).
- Es wird das Array `sons` erstellt, sodass alle 2^k -Söhne $e_{j,m}, m = 1, \dots, 2^k$, des Elementes $E_j \in \mathcal{E}_\ell$ durch `sons(j, :)` angesprochen werden können. Beispielsweise wird der m -te Sohn $e_{j,m}$ von E_j durch `elements_fine(sons(j, m), :)` aufgerufen. Weiters wird auch die Anordnung berücksichtigt, sodass der linke Knoten des ersten Sohnes $e_{j,1}$ mit jenem des Vaterelementes E_j übereinstimmt und der rechte Nachbar des Sohnes $e_{j,m}$ fortlaufend durch `sons(j, m + 1)` indiziert werden kann (Zeilen 26–35).
- Werte Φ_ℓ auf allen Söhnen $e_{j,m} \in \mathcal{E}_\ell^{(k)}$ in den in (5.68) definierten Quadraturknoten x_i aus, d.h. es wird $\Phi_\ell \circ \gamma_{j,m}(x_i)$ berechnet (Zeile 38).
- In Zeile 41 wird das $(\mathbf{nQ} \cdot \mathbf{nE} \times 1)$ -Array `sx2element` erstellt, sodass für $j := \mathbf{sx2element}(i)$ dann $\mathbf{sx}(i, :) \in E_j$ liegt, wobei für das $(\mathbf{nQ} \cdot \mathbf{nE} \times 2)$ -Array `sx`

$$\mathbf{sx} = \begin{pmatrix} \text{coordinates}(\gamma_{1,m}(x_1)) \\ \text{coordinates}(\gamma_{1,m}(x_2)) \\ \text{coordinates}(\gamma_{1,m}(x_3)) \\ \text{coordinates}(\gamma_{2,m}(x_1)) \\ \text{coordinates}(\gamma_{2,m}(x_2)) \\ \text{coordinates}(\gamma_{2,m}(x_3)) \\ \vdots \\ \text{coordinates}(\gamma_{\mathbf{nE},m}(x_1)) \\ \text{coordinates}(\gamma_{\mathbf{nE},m}(x_2)) \\ \text{coordinates}(\gamma_{\mathbf{nE},m}(x_3)) \end{pmatrix}$$

gilt, vgl. Zeilen 45–48.

- Berechne alle quadrierten Netzweiten und den äußeren Normalenvektor der Söhne $e_{j,m}$ für $j = 1, \dots, N$ (Zeilen 51–55).
- Anschließend werden mit den Gaußknoten (5.68) und den Gaußgewichten (5.69) die Koeffizienten $y_{j,m}$ auf allen Elementen $e_{j,m}, j = 1, \dots, N$, mittels Formel (5.60), d.h. durch

$$y_{j,m} = \frac{1}{2} \sum_{i=1}^4 \omega_i ((K' - 1/2)\Phi_\ell + WG_\ell) \circ \gamma_{j,m}(x_i),$$

berechnet (Zeilen 59–64).

- Danach wird die Einfachschichtpotentialmatrix $\mathbf{V}^{(k)}$ bezogen auf das Netz $\mathcal{E}_\ell^{(k)}$ berechnet (Zeile 68).
- Mittels (5.70) werden die quadrierten lokalen Fehlerindikatoren berechnet, sodass

$$\mathbf{v} := \left((\tilde{\sigma}_{\ell,V}^{(k,0)}(E_1))^2, \dots, (\tilde{\sigma}_{\ell,V}^{(k,0)}(E_N))^2 \right) \in \mathbb{R}^N$$

in Form des Spaltenvektors `ind` zurückgegeben wird (Zeilen 69–71).

- Abschließend erfolgt laut Formel (5.71) die Berechnung des Steinbach-Schätzers $(\tilde{\sigma}_{\ell,V}^{(k,0)})^2$ (Zeile 74).

5.13.2 Steinbach-Schätzer $\sigma_{\ell,A}$

Listing 10: computeEstSigmaA

```

1 function ind = computeEstSigmaA(coordinates,elements,x,uDh)
2 %*** set parameter for k-times uniform mesh-refinement of \ell-mesh
3 k = 1;
4
5 %*** Gaussian quadrature on [-1,1] with 1 node and exactness 1
6 quad_nodes = 0;
7 quad_weights = 2;
8 nQ = length(quad_nodes);
9
10 %*** allocate memory
11 nE = size(elements,1);
12 e = zeros(2^k*nE,1);
13 Pi_e = zeros(k+1,2^k*nE);
14 tmp = zeros(nE,1);
15 ind = zeros(nE,1);
16 sons = zeros(nE,2^k);
17 sons(:,1) = [1:nE];
18
19 %*** k-times uniform mesh-refinement
20 coordinates_fine = coordinates;
21 elements_fine = elements;
22
23 for i=1:k
24     [coordinates_fine,elements_fine,father2son] ...
25         = refineBoundaryMesh(coordinates_fine,elements_fine);
26     for m=1:2^(k-i+1):2^k
27         sons(:,m+2^(k-i)) = father2son(sons(:,m),2);
28     end
29 end
30
31 %*** evaluate phih elementwise at evaluation points
32 x_sx = reshape(repmat(x,1,nQ)',nQ*nE,1);
33
34 %*** sx2element(i) returns the element number j such that sx(i,:) lies on Ej
35 sx2element = reshape(repmat((1:nE),nQ,1),nQ*nE,1);

```

```
36
37 for m=1:2^k
38     %*** build vector of evaluations points as (nI*nE x 2)-matrix
39     a = coordinates_fine(elements_fine(sons(:,m),1),:);
40     b = coordinates_fine(elements_fine(sons(:,m),2),:);
41     sx = reshape(a,2*nE,1)*(1-quad_nodes) + reshape(b,2*nE,1)*(1+quad_nodes);
42     sx = 0.5*reshape(sx',nQ*nE,2);
43
44     %*** compute vector of (squared) element-widths
45     h = sum((a-b).^2,2);
46
47     %*** compute outer normal vector
48     normal = b-a;
49     normal = [normal(:,2),-normal(:,1)]./repmat(sqrt(h),1,2);
50
51     %*** evaluate p_m = (K'-1/2)*phih + W*gh elementwise at quad_nodes
52     %*** in son-Element e_jm
53     p_m = evaluateKadj(coordinates,elements,x,sx,normal(sx2element,:)) ...
54           - 0.5*x_sx + evaluateW(coordinates,elements,uDh,sx,normal(sx2element,:));
55     p_m = reshape(p_m,nQ,nE)';
56
57     %*** compute coefficients of \widetilde{e}^{(k,0)}
58     e(sons(:,m)) = 0.5*p_m*quad_weights;
59 end
60
61 %*** compute L2-Projection onto i-times uniform refined \ell_1-mesh, i=0...k
62 Pi_e(k+1,:) = e;
63
64 for i=k:-1:1
65     for m=1:2^(k-i+1):2^k-1
66         Pi_e(i,sons(:,m)) = ( Pi_e(i+1,sons(:,m)) + ...
67                               Pi_e(i+1,sons(:,m+2^(k-i)))) )*0.5;
68         Pi_e(i,sons(:,m+1:m+2^(k-i+1)-1)) = ...
69             repmat(Pi_e(i,sons(:,m)),1,2^(k-i+1)-1);
70     end
71 end
72
73 %*** compute (squared) local mesh-size of \ell_1-mesh
74 h = sum((coordinates(elements(:,1),:)-coordinates(elements(:,2),:)).^2,2);
75
76 %*** compute squared local error estimators sigmaA(Ej)^2
77 for i=2:k+1
78     tmp = 0*tmp;
79     for m=1:2^k
80         tmp = tmp + (Pi_e(i,sons(:,m))'-Pi_e(i-1,sons(:,m)))'.^2;
81     end
82     tmp = tmp*2^(-(i+k-1));
83     ind = ind + tmp;
84 end
85 ind = ind.*h;
```

In diesem Abschnitt implementieren wir den Fehlerschätzer

$$\tilde{\sigma}_{\ell,A}^{(k,q)} = \left(\sum_{j=1}^N (\tilde{\sigma}_{\ell,A}^{(k,q)}(E_j))^2 \right)^{1/2}$$

mit den lokalen Anteilen

$$\tilde{\sigma}_{\ell,A}^{(k,q)}(E_j) = \left(\langle A_\ell^{(k,-1/2)} \tilde{e}_\ell^{(k,q)}, \tilde{e}_\ell^{(k,q)} \rangle_{L^2(E_j)} \right)^{1/2}$$

und dem Multilevel-Operator

$$A_\ell^{(k,-1/2)} \tilde{e}_\ell^{(k,q)} = \sum_{i=1}^k h_\ell^{(i)} (\Pi_\ell^{(i)} - \Pi_\ell^{(i-1)}) \tilde{e}_\ell^{(k,q)}.$$

für $q = 0$. Wie bei der Implementierung des Steinbach-Schätzers über das Einfachschichtpotential in Abschnitt 5.13.1 gehen wir von einer Darstellung

$$\tilde{e}_\ell^{(k)} = \sum_{j=1}^N \sum_{m=1}^{2^k} y_{j,m} \chi_{j,m}$$

aus, wobei wir wieder $y_{j,m} := y_{j,m}^{(k)}$ mit den Koeffizienten aus (5.60) setzen. Im Unterschied zur Berechnung des Steinbach-Schätzers $\tilde{\sigma}_{\ell,V}^{(k,0)}$ berechnen wir diese Koeffizienten mit einer 1-Punkt Gaußquadratur mit dem Knoten

$$x_1 = 0 \tag{5.72}$$

und dem zugehörigen Gewicht

$$\omega_1 = 2. \tag{5.73}$$

Mit Lemma 5.19 (iii) und (5.66) erhalten wir mit $h_\ell^{(i)}|_{e_{j,m}} = 2^{-i} h_\ell|_{E_j} = 2^{-i} \text{length}(E_j)$ für alle

Söhne $e_{j,m}$ von E_j

$$\begin{aligned}
(\tilde{\sigma}_{\ell,A}^{(k,0)}(E_j))^2 &= \langle A_\ell^{(k,-1/2)} \tilde{e}_\ell^{(k)}, \tilde{e}_\ell^{(k)} \rangle_{L^2(E_j)} = \sum_{i=1}^k \int_{E_j} h_\ell^{(i)} (\Pi_\ell^{(i)} - \Pi_\ell^{(i-1)}) \tilde{e}_\ell^{(k)} \tilde{e}_\ell^{(k)} d\Gamma \\
&= \sum_{i=1}^k 2^{-i} \text{length}(E_j) \int_{E_j} (\Pi_\ell^{(i)} - \Pi_\ell^{(i-1)}) \tilde{e}_\ell^{(k)} \tilde{e}_\ell^{(k)} d\Gamma \\
&= \sum_{i=1}^k 2^{-i} \text{length}(E_j) \int_{E_j} (\Pi_\ell^{(i)} - \Pi_\ell^{(i-1)})^2 \tilde{e}_\ell^{(k)} \tilde{e}_\ell^{(k)} d\Gamma \\
&= \sum_{i=1}^k 2^{-i} \text{length}(E_j) \int_{E_j} (\Pi_\ell^{(i)} - \Pi_\ell^{(i-1)}) \tilde{e}_\ell^{(k)} (\Pi_\ell^{(i)} - \Pi_\ell^{(i-1)}) \tilde{e}_\ell^{(k)} d\Gamma \\
&= \sum_{i=1}^k 2^{-i} \text{length}(E_j) \int_{E_j} \left((\Pi_\ell^{(i)} - \Pi_\ell^{(i-1)}) \tilde{e}_\ell^{(k)} \right)^2 d\Gamma \\
&= \sum_{i=1}^k 2^{-i} \text{length}(E_j) \sum_{m=1}^{2^k} \int_{e_{j,m}} \left((\Pi_\ell^{(i)} - \Pi_\ell^{(i-1)}) \tilde{e}_\ell^{(k)} \right)^2 d\Gamma.
\end{aligned}$$

Da $\Pi_\ell^{(i)} \tilde{e}_\ell^{(k)}$ laut Lemma 5.19 (i) für alle $i \in \{0, \dots, k\}$ auf allen Söhnen $e_{j,m} \in \mathcal{E}_\ell^{(k)}$ konstant und weiters $\text{length}(e_{j,m}) = 2^{-k} \text{length}(E_j)$ ist, gilt

$$\begin{aligned}
(\tilde{\sigma}_{\ell,A}^{(k,0)}(E_j))^2 &= \sum_{i=1}^k 2^{-i} \text{length}(E_j) \sum_{m=1}^{2^k} \text{length}(e_{j,m}) \left((\Pi_\ell^{(i)} - \Pi_\ell^{(i-1)}) \tilde{e}_\ell^{(k)}|_{e_{j,m}} \right)^2 \\
&= \sum_{i=1}^k 2^{-i} \text{length}(E_j) \sum_{m=1}^{2^k} 2^{-k} \text{length}(E_j) \left((\Pi_\ell^{(i)} - \Pi_\ell^{(i-1)}) \tilde{e}_\ell^{(k)}|_{e_{j,m}} \right)^2.
\end{aligned}$$

Folglich erhalten wir

$$(\tilde{\sigma}_{\ell,A}^{(k,0)}(E_j))^2 = \text{length}(E_j)^2 \sum_{i=1}^k 2^{-(k+i)} \sum_{m=1}^{2^k} \left((\Pi_\ell^{(i)} - \Pi_\ell^{(i-1)}) \tilde{e}_\ell^{(k)}|_{e_{j,m}} \right)^2. \quad (5.74)$$

Wir interessieren uns nun für die Berechnung von $\Pi_\ell^{(i)} \tilde{e}_\ell^{(k)}$ für $i = 0, \dots, k$ bezogen auf das Netz $\mathcal{E}_\ell^{(k)}$. Sei dazu $E^{(i)} \in \mathcal{E}_\ell^{(i)}$ für festes $i \in \{0, \dots, k\}$. Mit $e_{j,m} \in \mathcal{E}_\ell^{(k)}$ lässt sich dieses Element durch

$$E^{(i)} = \bigcup_{m=1}^{2^{k-i}} e_{j,m}.$$

darstellen, vgl. Abbildung 4. Dann gilt für die Projektion $\Pi_\ell^{(i)}$ auf ein solches Element wegen

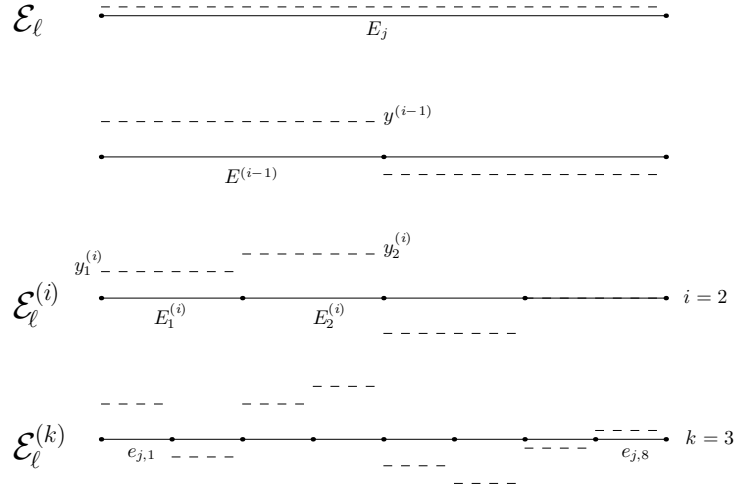


Abbildung 4: Illustration der Elementverfeinerung für $k=3$ inkl. Koeffizienten von $\Pi_\ell^{(i)} \tilde{e}_\ell^{(k)}$, $i=0 \dots k$

$$\text{length}(E^{(i)}) = 2^{k-i} \text{length}(e_{j,m})$$

$$\begin{aligned} y^{(i)} &:= \Pi_\ell^{(i)} \tilde{e}_\ell^{(k)}|_{E^{(i)}} = \frac{1}{\text{length}(E^{(i)})} \int_{E^{(i)}} \tilde{e}_\ell^{(k)} d\Gamma = \frac{1}{\text{length}(E^{(i)})} \sum_{m=1}^{2^{k-i}} \int_{e_{j,m}} y_{j,m} d\Gamma \\ &= \frac{\text{length}(e_{j,m})}{\text{length}(E^{(i)})} \sum_{m=1}^{2^{k-i}} y_{j,m} = 2^{i-k} \sum_{m=1}^{2^{k-i}} y_{j,m}. \end{aligned}$$

Ist die Projektion $\Pi_\ell^{(i)} \tilde{e}_\ell^{(k)}$ auf einem Netz bekannt, so lässt sich daraus die Projektion $\Pi_\ell^{(i-1)} \tilde{e}_\ell^{(k)}$ recht einfach berechnen. Sei dazu $E_1^{(i)}, E_2^{(i)} \in \mathcal{E}_\ell^{(i)}$ mit $E^{(i-1)} = E_1^{(i)} \cup E_2^{(i)} \in \mathcal{E}_\ell^{(i-1)}$. Weiters seien $y_1^{(i)}$ und $y_2^{(i)}$ die Koeffizienten von $\Pi_\ell^{(i)} \tilde{e}_\ell^{(k)}$ auf $E_1^{(i)}$ und $E_2^{(i)}$. Analog ist $y^{(i-1)}$ der Koeffizient von $\Pi_\ell^{(i-1)} \tilde{e}_\ell^{(k)}$ auf $E^{(i)}$. Es gilt wegen obiger Rechnung für die Koeffizienten der Nachbarelemente $E_1^{(i)}, E_2^{(i)}$

$$y_1^{(i)} = 2^{i-k} \sum_{m=1}^{2^{k-i}} y_{j,m}, \quad y_2^{(i)} = 2^{i-k} \sum_{m=2^{k-i}+1}^{2^{k-(i-1)}} y_{j,m}$$

bzw. für den Koeffizient auf dem vereinigten Element $E^{(i-1)}$

$$y^{(i-1)} = 2^{i-1-k} \sum_{m=1}^{2^{k-(i-1)}} y_{j,m}.$$

Dann gilt

$$2^{-(i-1-k)} y^{(i-1)} = 2^{-(i-k)} (y_1^{(i)} + y_2^{(i)})$$

bzw. dazu äquivalent

$$y^{(i-1)} = \frac{1}{2}(y_1^{(i)} + y_1^{(i)}). \quad (5.75)$$

Damit lassen sich die Projektionen $\Pi_\ell^{(i)} \tilde{e}_\ell^{(k)}$ iterativ berechnen.

Für die Implementierung setzen wir $k := 1$. Wir werden diese Wahl des Parameters k , wie auch die der Gaußquadratur, mittels numerischer Simulationen in Abschnitt 6.2.2 begründen.

Wir kommen zur Dokumentation von Listing 10:

- Die Funktion übernimmt das Netz $\mathcal{E}_\ell$ in Form von `coordinates` und `elements`. Weiters werden auch der Koeffizientenvektor $\mathbf{x}$ der Galerkin-Lösung Φ_ℓ in Form von `x` und die Auswertungen der Dirichlet-Daten g an allen Knoten der Triangulierung als Spaltenvektor `uDh` eingelesen. Als Output werden die quadrierten lokalen Fehleranteile $(\tilde{\sigma}_{\ell,A}^{(k,0)}(E))^2$ für alle $E \in \mathcal{E}_\ell$ als Spaltenvektor `sigmaA` zurückgegeben (Zeile 1).
- Wir definieren den Parameter k für die k -fache uniforme Verfeinerung des Netzes $\mathcal{E}_\ell$ (Zeile 3).
- Anschließend werden die Gaußknoten (5.72) und -gewichte (5.73) initialisiert und deren Anzahl in `nQ` geschrieben (Zeilen 6–8).
- Es wird Speicher für die zur Berechnung notwendigen Arrays allokiert und initialisiert (Zeilen 11–17).
- Die Zeilen 20–59 identisch mit den Zeilen 26–65 von Listing 9 und dort bei Bedarf nachzusehen. Sie dienen zur Bestimmung der Koeffizienten $y_{j,m}$ von $\tilde{e}_\ell^{(k)}$ auf $\mathcal{E}_\ell^{(k)}$.
- Mit Formel (5.75) berechnen wir beginnend bei den bekannten Koeffizienten $y_{j,m}$ von $\tilde{e}_\ell^{(k)}$ auf dem feinsten Netz $\mathcal{E}_\ell^{(k)}$ iterativ alle Projektionen $\Pi_\ell^{(i)} \tilde{e}_\ell^{(k)}$ für $i = 0, \dots, k-1$ in Richtung größer werdender Netze. Die Projektion $\Pi_\ell^{(i)} \tilde{e}_\ell^{(k)}$ wird dann in die Zeile `Pi_e(i+1, :)` $\in \mathbb{R}^{1 \times 2^k nE}$ bezogen auf das Netz $\mathcal{E}_\ell^{(k)}$ geschrieben. Wichtig ist hierbei, dass der rechte Nachbar des Sohnes $e_{j,m} \in \mathcal{E}_\ell^{(k)}$ durch `sons(j, m+1)` indiziert werden kann, vgl. auch wieder Abbildung 4 (Zeilen 62–71).
- Weiters werden die quadrierten Netzweiten für alle Elemente der Triangulierung $\mathcal{E}_\ell$ berechnet (Zeile 74).
- Abschließend wird Formel (5.74) realisiert. Zurückgegeben wird der Vektor

$$\mathbf{v} := \left((\tilde{\sigma}_{\ell,A}^{(k,0)}(E_1))^2, \dots, (\tilde{\sigma}_{\ell,A}^{(k,0)}(E_N))^2 \right) \in \mathbb{R}^N,$$

sodass $\tilde{\sigma}_{\ell,A}^{(k,0)} = (\sum_{j=1}^N v_j)^{1/2}$ gilt (Zeilen 77–85).

6 Numerische Resultate

In diesem letzten Kapitel werden wir die vorgestellten Fehlerschätzer bei der Lösung des Dirichlet-Problems (1.6), d.h.

$$\begin{aligned} -\Delta u &= 0 & \text{in } \Omega \\ u &= g & \text{auf } \Gamma, \end{aligned}$$

mit Hilfe von numerischen Simulationen in MATLAB analysieren. Wir verwenden dabei das vorhandene Softwarepaket HILBERT (<http://www.asc.tuwien.ac.at/abem/hilbert/>) und erweitern es um die Steinbach-Schätzer $\sigma_{\ell,V}$, $\sigma_{\ell,A}$.

HILBERT verwendet für die adaptive BEM einen Algorithmus aufbauend auf Algorithmus 4.3. Als Parameter für die adaptive Verfeinerung wählen wir $\theta = 1/4$ und als optionalen Parameter $\rho = 1/4$. Demnach gilt, vgl. Abschnitt 4.1,

$$\frac{1}{4} \sum_{E \in \mathcal{E}_\ell} \iota_\ell(E)^2 \leq \sum_{E \in \mathcal{M}_\ell} \iota_\ell(E)^2$$

mit dem Verfeinerungsindikator ι_ℓ und den markierten Elemente $\mathcal{M}_\ell \subseteq \mathcal{E}_\ell$. Entsprechend wird dann eine Obermenge $\mathcal{M} \subseteq \overline{\mathcal{M}}_\ell \subseteq \mathcal{E}_\ell$ mit

- (a) $\frac{\#\overline{\mathcal{M}}_\ell}{\#\mathcal{E}_\ell} \geq \frac{1}{4}$
- (b) $\iota_\ell(E) \geq \iota_\ell(E')$ für alle $E \in \overline{\mathcal{M}}_\ell$ und $E' \in \mathcal{E}_\ell \setminus \overline{\mathcal{M}}_\ell$

durch die Funktion `markElements` markiert und dann mittels `refineBoundaryMesh` verfeinert. Wir werden diese Wahl der Parameter θ und ρ für alle Simulationen beibehalten.

Ziel in diesem Kapitel wird es zum Einen sein, mehrere adaptive Steuerungen miteinander zu vergleichen, sodass für den Verfeinerungsindikator auf einem Element $E \in \mathcal{E}_\ell$

$$\iota_\ell(E)^2 = \varsigma_\ell(E)^2 + \text{osc}_{D,\ell}(E)^2 \quad \text{für } \varsigma_\ell \in \{\mu_\ell, \tilde{\mu}_\ell, \tau_\ell, \varrho_\ell, \varphi_\ell, \sigma_{\ell,V}, \sigma_{\ell,A}\}$$

mit den $(h - h/2)$ -Schätzern μ_ℓ , $\tilde{\mu}_\ell$, inkl. des Two-Level-Fehlerschätzers τ_ℓ , dem gewichteten Residualschätzer ϱ_ℓ , dem Faermann-Residualschätzer φ_ℓ und den beiden Steinbach-Schätzern $\sigma_{\ell,V}$, $\sigma_{\ell,A}$ gelten wird. Weiters wollen wir diese Fehlerschätzer auch hinsichtlich Stabilität, Konvergenzrate, Genauigkeit und Berechnungsdauer untersuchen, sodass wir am Ende eine vollständige Analyse aller Fehlerschätzer vorliegen haben.

Wir werden die Ergebnisse auf den Beispielgebieten **L-Shape**, **Z-Shape** und beim **Schlitz-Problem** diskutieren. Die Gebiete mit jeweiliger Ausgangstriangulierung für die Simulationen sind in den Abbildungen 5 bis 7 dargestellt.

Auf all diesen Gebieten schreiben wir die exakte Lösung vor, sodass wir den exakten Fehler $\|\phi - \Phi\|_V$ der Galerkin-Approximation Φ_ℓ bzgl. dem Netz $\mathcal{E}_\ell$ als Folgerung von (3.33) mit Hilfe der exakten Lösung ϕ durch

$$\|\phi - \Phi_\ell\|_V \lesssim \|h_\ell^{1/2}(\phi - \Phi_\ell)\|_{L^2(\Gamma)} + \text{osc}_{D,\ell} =: \text{err}_{N,\ell} + \text{osc}_{D,\ell}$$

abschätzen können. Für ein hinreichend glattes Neumann-Datum ϕ führt dies bei uniformer Netzverfeinerung mit Netzweite h zu einer Konvergenzordnung $\text{err}_{N,\ell} = \mathcal{O}(h^{3/2})$. Die Funktion `computeErrNeumann` aus `HILBERT` approximiert $\text{err}_{N,\ell}$ durch eine 3-Punkt Gaußquadratur, sodass in diesem Fall für die Implementierung $|\widetilde{\text{err}}_{N,\ell} - \text{err}_{N,\ell}| = \mathcal{O}(h^{5/2})$ gilt, siehe dazu auch [1, Abschnitt 5.5]. Als obere Schranke (bis auf eine Konstante) für den Fehler $\|\phi - \Phi_\ell\|_V$ werden wir daher

$$\text{error}_\ell := \text{err}_{N,\ell} + \text{osc}_{D,\ell} \tag{6.1}$$

berechnen.

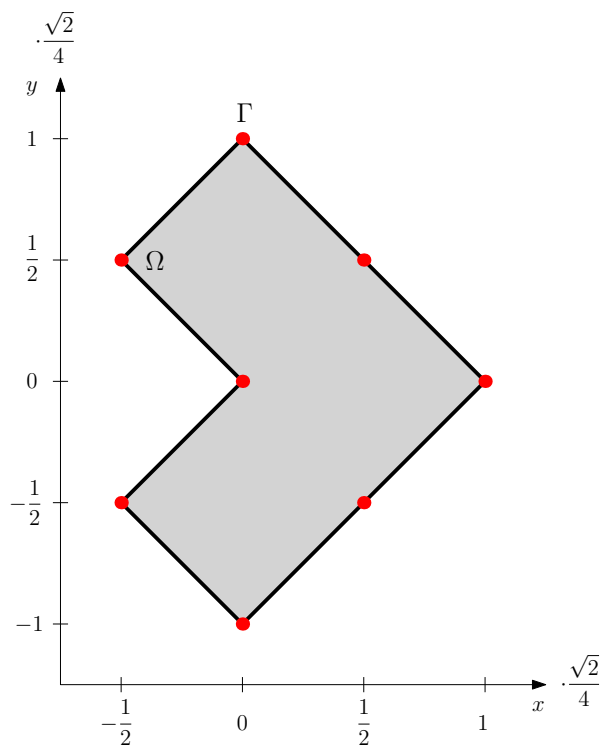


Abbildung 5: L-Shape mit 8 Elementen

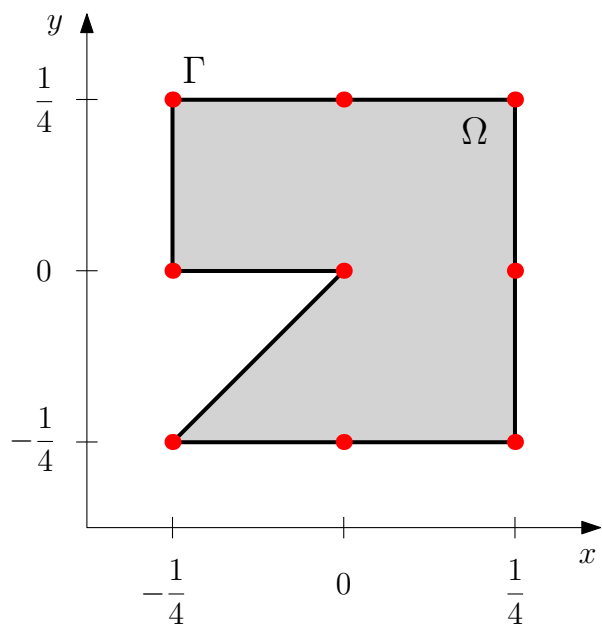


Abbildung 6: Z-Shape mit 9 Elementen

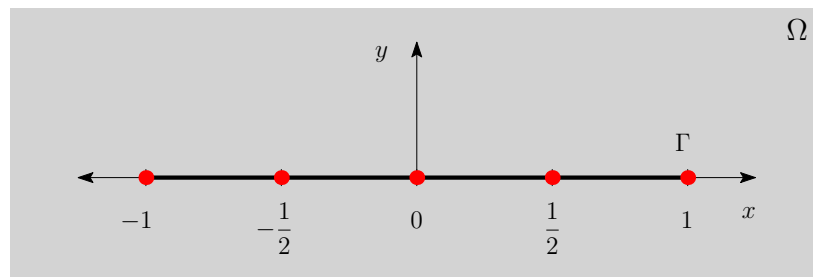


Abbildung 7: Schlitz-Problem mit 4 Elementen

6.1 Vorüberlegungen für die Berechnung auf den einzelnen Gebieten

6.1.1 L-Shape

In Abbildung 5 ist das Gebiet Ω und die Triangulierung $\mathcal{E}_0$ des zugehörigen Randes Γ als Ausgangsnetz für den adaptiven Algorithmus dargestellt. Die speziellen Maße garantieren dabei die Beschränktheit $\text{diam}(\Omega) < 1$. Insbesondere gibt es eine eindeutige Lösung $\phi \in H^{-1/2}(\Gamma)$ der Variationsformulierung

$$\langle V\phi, \Psi_\ell \rangle_\Gamma = \langle (K + 1/2)g, \Psi_\ell \rangle_\Gamma \quad \text{für alle } \Psi_\ell \in \mathcal{P}^0(\mathcal{E}_\ell),$$

vgl. Abschnitt 3.1.

Auf diesem Gebiet schreiben wir als exakte Lösung des Dirichlet-Problems (1.6) in Polarkoordinaten

$$u(r, \theta) = r^\alpha \cos(\alpha\theta) \quad \text{mit } \alpha = \frac{2}{3}$$

für $r > 0$ und $\theta \in [0, 2\pi]$ vor. Für den Gradienten in Zylinderkoordinaten

$$\nabla u = \mathbf{e}_r \frac{\partial}{\partial r} u + \mathbf{e}_\theta \frac{1}{r} \frac{\partial}{\partial \theta} u,$$

wobei

$$\mathbf{e}_r = \cos \theta \mathbf{e}_x + \sin \theta \mathbf{e}_y, \quad \mathbf{e}_\theta = -\sin \theta \mathbf{e}_x + \cos \theta \mathbf{e}_y$$

und $\mathbf{e}_x, \mathbf{e}_y$ die Einheitsvektoren in x - bzw. y -Richtung sind, erhalten wir

$$\begin{aligned} \nabla u &= \alpha r^{\alpha-1} \cos(\alpha\theta) \mathbf{e}_r - \alpha r^{\alpha-1} \sin(\alpha\theta) \mathbf{e}_\theta \\ &= \alpha r^{\alpha-1} \left((\cos(\theta) \cos(\alpha\theta) + \sin(\theta) \sin(\alpha\theta)) \mathbf{e}_x + (\sin(\theta) \cos(\alpha\theta) - \cos(\theta) \sin(\alpha\theta)) \mathbf{e}_y \right), \end{aligned}$$

d.h.

$$\nabla u = \alpha r^{\alpha-1} \begin{pmatrix} \cos(\theta) \cos(\alpha\theta) + \sin(\theta) \sin(\alpha\theta) \\ \sin(\theta) \cos(\alpha\theta) - \cos(\theta) \sin(\alpha\theta) \end{pmatrix} \in \mathbb{R}^2 \quad (6.2)$$

mit $\alpha = \frac{2}{3}$. Dann ist das exakte Neumann-Datum ϕ durch die Normalenableitung von u auf Γ durch

$$\phi = \frac{\partial}{\partial \mathbf{n}} u = \nabla u \cdot \mathbf{n} \quad (6.3)$$

mit dem Einheitsnormalenvektor $\mathbf{n} = n_x \mathbf{e}_x + n_y \mathbf{e}_y$ gegeben. Demnach ist $g = u|_\Gamma$ glatt, hingegen weist die Normalenableitung $\phi = \frac{\partial}{\partial \mathbf{n}} u$ eine Singularität an der einspringenden Ecke, d.h. an $(x, y) = (0, 0)$, und einige Unstetigkeiten entlang des Randes auf, vgl. auch Abbildung 8.

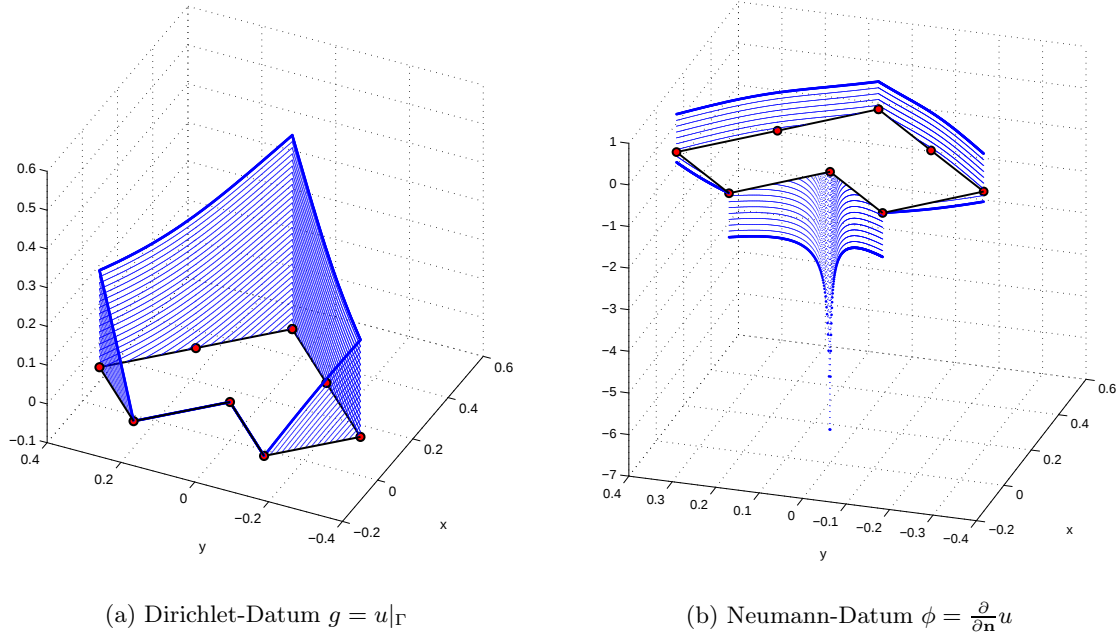


Abbildung 8: Dirichlet- und Neumann-Daten auf L-Shape

6.1.2 Z-Shape

In Abbildung 6 sehen wir das Z-Shape mit der Ausgangstriangulierung $\mathcal{E}_0$ für die Simulation. Aufgrund der Maße des Gebietes gilt $\text{diam}(\Omega) < 1$, was die eindeutige Lösbarkeit der Variationsformulierung impliziert. Als Lösung des Dirichlet-Randwertproblems (1.6) auf dem Gebiet Ω definieren wir, ähnlich dem L-Shape, die Lösung durch

$$u(r, \theta) = r^\alpha \cos(\alpha\theta) \quad \text{mit } \alpha = \frac{4}{7}$$

für $r > 0$ und $\theta \in [0, 2\pi]$. Wie im vorigen Abschnitt lässt sich dann die Normalenableitung $\phi = \frac{\partial}{\partial \mathbf{n}} u$ mit (6.2) durch (6.3) berechnen. Im Vergleich zur Lösung auf dem L-Shape ($\alpha = 2/3$) weist das Neumann-Datum ϕ eine ausgeprägtere Singularität als beim L-Shape an der einspringenden Ecke $(x, y) = (0, 0)$ auf. Weiters verlieren wir auch beim Dirichlet-Datum g an Regularität. Beide Funktionen sind in Abbildung 9 dargestellt.

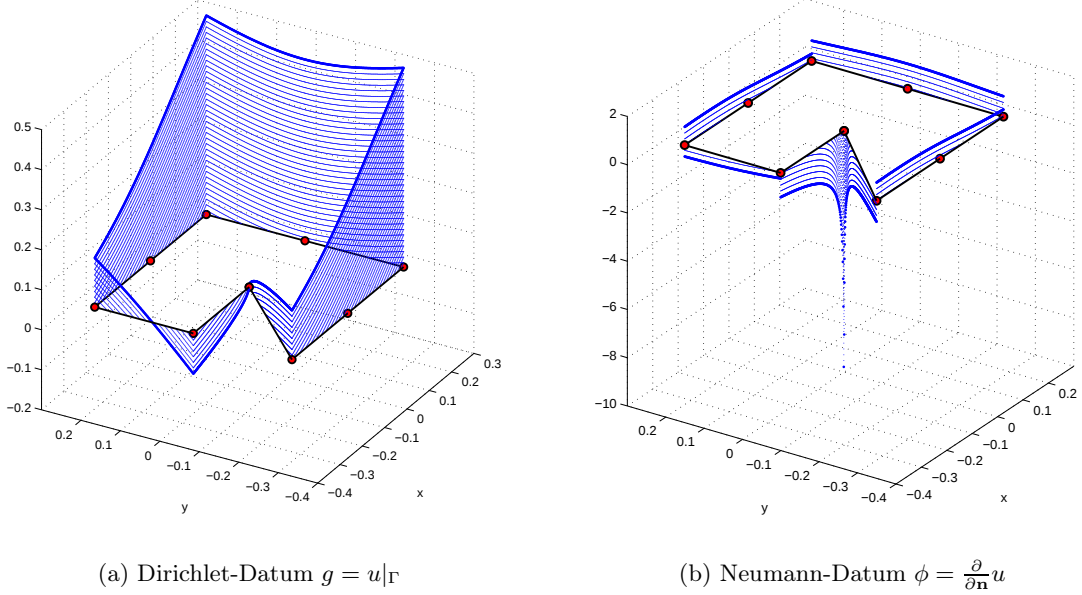


Abbildung 9: Dirichlet- und Neumann-Daten auf Z-Shape

6.1.3 Schlitz

Für das dritte und letzte numerische Experiment untersuchen wir das Schlitz-Problem. Das Gebiet Ω ist hierbei $\mathbb{R}^2 \setminus (-1, 1) \times \{0\}$, vgl. Abbildung 7. Beim Schlitz-Problem geben wir die Symm'sche Integralgleichung durch

$$V\phi = -x \quad \text{auf } \Gamma = (-1, 1) \times \{0\} \quad (6.4)$$

vor. Demnach gilt laut (3.4) $-x = (K + 1/2)g$. Obwohl das Gebiet Ω nicht beschränkt ist, garantiert uns beispielsweise [18, Satz 6.5] trotzdem die Elliptizität von V auf Γ und damit die eindeutige Lösbarkeit der Variationsformulierung von (6.4), d.h. von

$$\langle V\phi, \Psi_{\ell} \rangle_{\Gamma} = -\langle x, \Psi_{\ell} \rangle_{\Gamma} \quad \text{für alle } \Psi_{\ell} \in \mathcal{P}^0(\mathcal{E}_{\ell}).$$

Die exakte Lösung der Normalenableitung lautet dann

$$\phi(x, y) = \frac{-2x}{\sqrt{1-x^2}}$$

für alle $x \in (-1, 1)$ und $y = 0$. Die spezielle Gestalt des Gebietes und die Veränderung der rechten Seite müssen nun in der Problemlösung berücksichtigt werden. Da der Rand Γ nicht mehr geschlossen ist, gilt die Beziehung $\#\mathcal{K}_{\ell} = \#\mathcal{E}_{\ell} + 1$ zwischen der Menge der Knoten $\mathcal{K}_{\ell}$ und der Elementanzahl der Triangulierung $\mathcal{E}_{\ell}$. Sei $N := \#\mathcal{E}_{\ell}$, dann erhalten wir mit den Knoten

$z_j = (z_j^{(1)}, z_j^{(2)}) \in \mathcal{K}_\ell$ und den zugehörigen Hutfunktionen ζ_j für $x \in (-1, 1)$ auf dem Netz $\mathcal{E}_\ell$

$$x = \sum_{j=1}^{N+1} z_j^{(1)} \zeta_j(x)$$

für alle $\ell \in \mathbb{N}_0$. Insbesondere wird die rechte Seite von (6.4) exakt dargestellt, was zum Verschwinden der Dirichlet-Datenoszillationen, d.h. $\text{osc}_{D,\ell} \equiv 0$, führt. Mit der Darstellung

$$\Phi_\ell = \sum_{j=1}^N x_j \chi_j$$

folgt dann für die Variationsformulierung auf $\mathcal{P}^0(\mathcal{E}_\ell)$

$$\sum_{j=1}^N x_j \langle V \chi_j, \chi_k \rangle_\Gamma = \langle V \Phi_\ell, \chi_k \rangle_\Gamma = -\langle x, \chi_k \rangle_\Gamma = -\sum_{j=1}^{N+1} z_j^{(1)} \langle \zeta_j, \chi_k \rangle_\Gamma$$

für alle $k = 1, \dots, N$. Sei nun $\mathbf{z} := (z_1^{(1)}, \dots, z_{N+1}^{(1)})$ der Vektor der x -Koordinaten der Knoten $z_j = (z_j^{(1)}, z_j^{(2)}) \in \mathcal{K}_\ell$ und $\mathbf{x} := (x_1, \dots, x_N)$ der Koeffizientenvektor von Φ_ℓ . Dann implizieren die Definitionen (3.11) und (3.13)

$$\mathbf{V}\mathbf{x} = -\mathbf{M}\mathbf{z} =: \mathbf{b}$$

für das zu lösende lineare Gleichungssystem, wobei nun $\mathbf{V} \in \mathbb{R}^{N \times N}$ und $\mathbf{M} \in \mathbb{R}^{N \times (N+1)}$ gilt.

Weiters ist das Residuum für die Berechnung des gewichteten Residualschätzers ϱ_ℓ und des Faermann-Residualschätzers φ_ℓ nun durch

$$R_\ell = V(\Phi_\ell - \phi) = V\Phi_\ell + x$$

gegeben.

Der nicht geschlossene Rand führt auf eine Einschränkung bei der Wahl der Fehlerschätzer, sodass wir auf die Steinbach-Schätzer $\sigma_{\ell,V}$ und $\sigma_{\ell,A}$ gänzlich verzichten müssen. Für den Faermann-Residualschätzer φ_ℓ sind über die Adaptierung der Berechnung des Residuums R_ℓ hinaus Veränderungen vorzunehmen, da er für die Berechnung auf einem Element zumindest auch ein Nachbarelement - in unserer Implementierung sogar beide - benötigt. Wir fügen also ein virtuelles Element $\tilde{E}$ hinzu, indem wir den Punkt $(1, 0)$ mit dem Punkt $(-1, 0)$ verbinden und für das Residuum $R_\ell|_{\tilde{E}} \equiv 0$ setzen.

Im Anhang ist mit Listing A.1 der Programmcode für den Faermann-Residualschätzer φ_ℓ für das Schlitz-Problem zu finden.

6.2 Parameterwahl der Steinbach-Schätzer

Da uns Lemma 5.17 zu den beiden Steinbach-Schätzern keine Aussage über eine hinreichend kleine Netzweite $\|h_\ell^{(k)}\|_{L^\infty(\Gamma)} < \|h_\ell\|_{L^\infty(\Gamma)}$ der k -fachen uniformen Verfeinerung $\mathcal{E}_\ell^{(k)}$ des Netzes $\mathcal{E}_\ell$ liefert, werden wir diesen Parameter mittels Simulationen schätzen. D.h. wir wollen für beide Steinbach-Schätzer jeweils einen (sinnvollen) Parameter $k \in \mathbb{N}$ bestimmen, sodass die beiden Steinbach-Schätzer

$$\sigma_{\ell,V}^{(k,0)} = \left(\langle V e_\ell^{(k)}, e_\ell^{(k)} \rangle_\Gamma \right)^{1/2}$$

und

$$\sigma_{\ell,A}^{(k,0)} = \left(\langle A_\ell^{(k,-1/2)} e_\ell^{(k)}, e_\ell^{(k)} \rangle_{L^2(\Gamma)} \right)^{1/2}$$

mit

$$e_\ell^{(k,0)} = e_\ell^{(k)} = \Pi_\ell^{(k)}(\tilde{\phi}_\ell - \Phi_\ell)$$

effizient und zuverlässig sind.

Wir werden die dafür notwendigen Simulationen auf dem *symmetrisch rotierten L-Shape* mit dem $\tilde{\mu}_\ell$ -adaptiv gesteuerten Algorithmus durchführen und weiters auch auf dem *Z-Shape* diese Parameterwahl überprüfen.

6.2.1 L-Shape: Parameterschätzung für Steinbach-Schätzer $\sigma_{\ell,V}$

Mit den Vorüberlegungen aus Abschnitt 6.1.1 erhalten wir mit den Codes für den Fehlerschätzer $\tilde{\mu}_\ell$ (Listing 5) und jenem des Steinbach-Schätzers $\sigma_{\ell,V}$ (Listing 9) bei Berechnung der L^2 -Orthogonalprojektion mittels 3-Punkt Gaußquadratur die in Abbildung 10 dargestellten Simulationsergebnisse auf dem L-Shape. Da für jede Parameterwahl die $\tilde{\mu}_\ell$ -adaptive Steuerung verwendet wird, werden immer dieselben Triangulierungen erzeugt. Im oberen Plot sehen wir den Fehlerschätzer $\sigma_{\ell,V}^{(k,0)}$ über der Elementanzahl bei k -facher uniformer Verfeinerung von $\mathcal{E}_\ell$ mit $k \in \{0, 1\}$. Auch wenn der Fall $k = 0$ in der Analysis nicht behandelt wird, wollen wir ihn in den Simulationen untersuchen. Als weiteren Vergleich sind auch die Fehlerschätzer $\sigma_{\ell,V}^{(k=2,q)}$ für $q \in \{1, 5\}$ Auswertungen der Neumann-Summe, implementiert durch die Iterationsvorschrift (5.61), aufgetragen. Weiters sehen wir die obere Schranke error_ℓ für den Fehler $\|\phi - \Phi_\ell\|_V$ bei $\tilde{\mu}_\ell$ -adaptiver Steuerung. Im Falle $k = q = 0$ approximieren laut Lemma 5.15 und der L^2 -Orthogonalprojektion als Mittelwertbildung in grober Näherung auf dem Netz $\mathcal{E}_\ell$

$$e_\ell^{(0,0)} = e_\ell^{(0)} = \Pi_\ell(\tilde{\phi}_\ell - \Phi_\ell) \approx \phi_\ell - \Phi_\ell$$

mit der zum Fehler $\phi - \Phi_\ell$ bis auf Datenoszillationen bzgl. der Energienorm äquivalenten rechten Seite. Wir erkennen, dass der Kurvenverlauf einen instabilen Eindruck macht und daher nicht sehr vielversprechend aussieht. Der Fehlerschätzer $\sigma_{\ell,V}^{(k,0)}$ mit $k = 1$ und damit bei einer uniformen Verfeinerung von $\mathcal{E}_\ell$ weist bereits einen $\mathcal{O}(N^{-3/2})$ -Konvergenzverlauf auf. Folglich vermuten wir bei dieser Parameterwahl bereits die Effizienz und Zuverlässigkeit des Fehlerschätzers. Die weiteren Terme der Neumann-Reihe scheinen keine nennenswerte Verbesserung hinsichtlich Stabilität

und Konvergenzverlauf beizutragen. In Abbildung 10 unten ist die Berechnungsdauer der Funktion `computeEstSigmaV` über der Anzahl der Elemente aufgetragen. Jede zusätzliche Verfeinerung und Auswertung von Neumann-Terme führt zu einer erwarteten Erhöhung der Berechnungsdauer des Steinbach-Schätzers, die es (wenn möglich) zu vermeiden gilt.

Wir erhoffen uns bei einer höheren Anzahl von Quadraturpunkten einen stabileren Konvergenzverlauf. In Abbildung 11 ist das Ergebnis derselben Simulation mit einer 5-Punkt Gaußquadratur dargestellt. Bis auf einen stabileren Verlauf von $\sigma_{\ell,V}^{(0,0)}$ ist auf den ersten Blick kein Unterschied zur 3-Punkt-Gaußquadratur-Variante zu erkennen. Bei genauerer Betrachtung in Abbildung 12 fällt die leicht abflachende Tendenz bei der Berechnung von $\sigma_{\ell,V}^{(1,0)}$ mittels 3-Punkt Gaußquadratur bei hoher Elementanzahl auf. Die Kurven von $\sigma_{\ell,V}^{(5,2)}$ sind dagegen identisch. Mehrerer Neumann-Summanden und größeres $k > 0$ führen also zu einem stabileren Konvergenzverlauf. Hinsichtlich auf die Berechnungsdauer ist dagegen ein Fehlerschätzer mit möglichst geringer Anzahl an Neumann-Summanden wünschenswert. In Abbildung 12 unten sehen wir, dass der durch die Quadraturknotenerhöhung entstehende größere Zeitaufwand ist minimal.

Für die weiteren Simulationen wählen wir daher eine 5-Punkt Gaußquadratur mit den Parametern $k := 1$ und $q := 0$ für die Berechnung des Steinbach-Schätzers $\sigma_{\ell,V}^{(k,q)}$ und schreiben vereinfachend $\sigma_{\ell,V}$.

6.2.2 L-Shape: Parameterschätzung für Steinbach-Schätzer $\sigma_{\ell,A}$

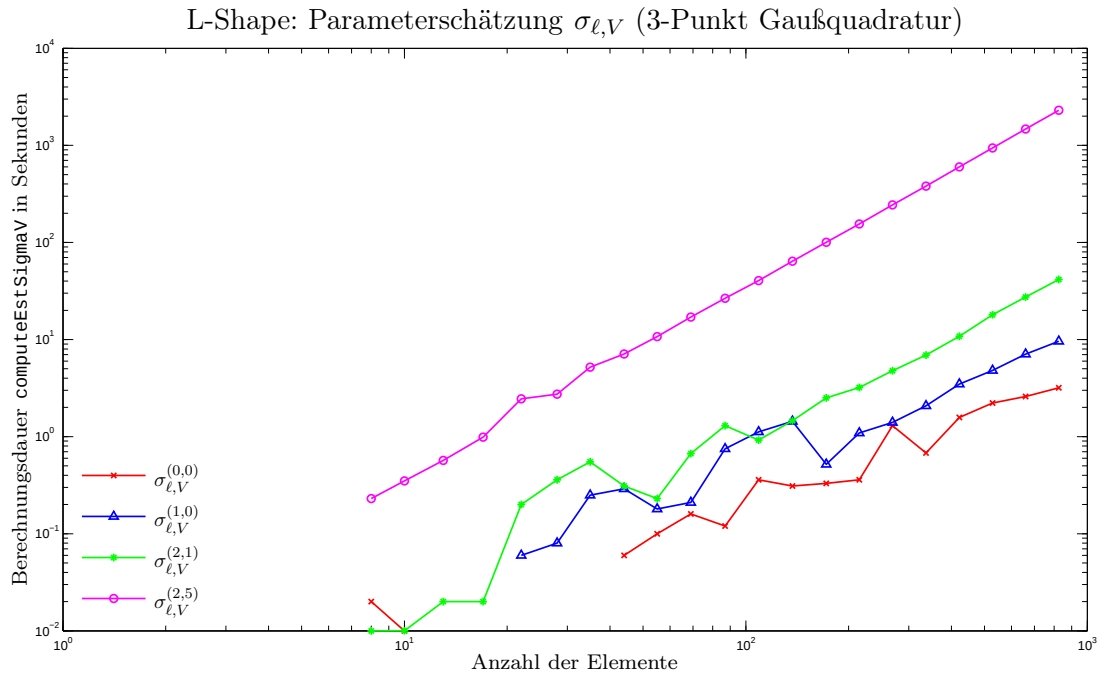
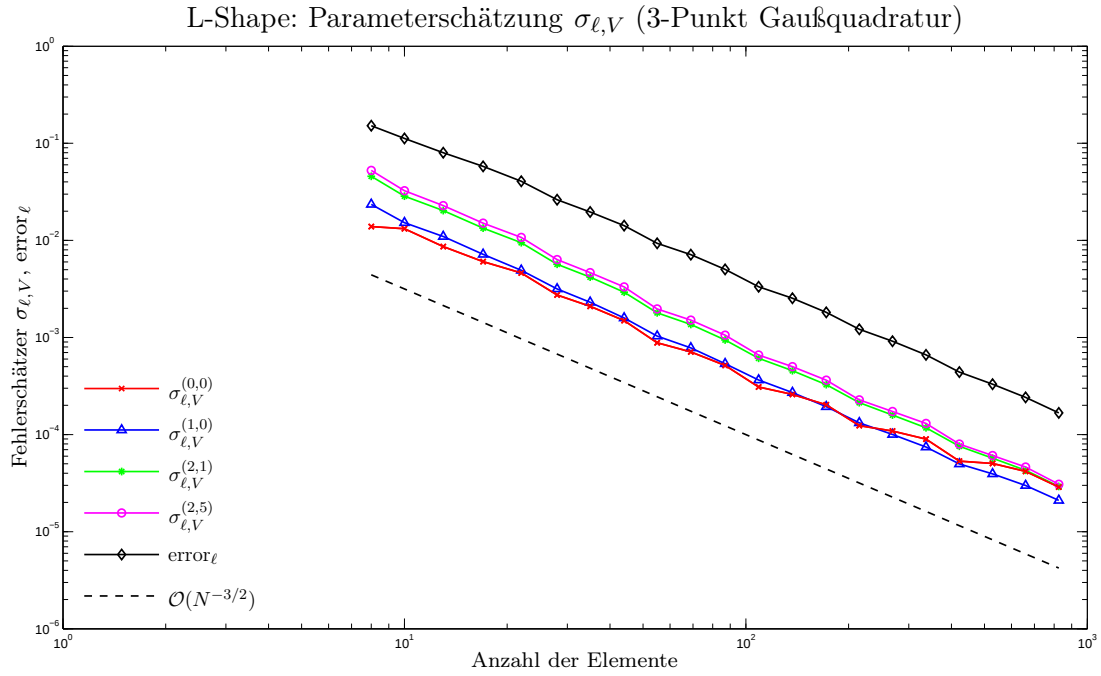
Mit dem Code für den Steinbach-Schätzer $\sigma_{\ell,A}$ (Listing 10) erhalten wir bei $\tilde{\mu}_\ell$ -adaptiver Steuerung und einer 3-Punkt Gaußquadratur die in Abbildung 13 dargestellten Simulationsergebnisse.

Der obere Plot von Abbildung 13 zeigt, dass die Fehlerschätzer $\sigma_{\ell,A}^{(k,0)}$ für alle $k \in \{1, 2, 3, 5\}$ ⁴ beinahe deckungsgleich sind und alle die gleiche Konvergenzrate aufweisen. Damit scheinen die Fehlerschätzer mit allen berechneten Parametern bei einer Implementierung mit einer 3-Punkt Gaußquadratur effizient und zuverlässig zu sein. Weiters zeigt die Zeitmessung im unteren Plot von Abbildung 13, dass sich die Berechnungsdauer des Fehlerschätzers von `computeEstSigmaA` bei jeder Vergrößerung von $k > 0$ erhöht.

Wir wollen nun die Berechnung der Projektion $e_\ell^{(k)} = \Pi_\ell^{(k)}(\tilde{\phi}_\ell - \Phi_\ell)$ mit einer geringeren Quadraturordnung durchführen und die Resultate des daraus resultierenden Steinbach-Schätzers $\sigma_{\ell,A}^{(k,0)}$ bewerten. In Abbildung 14 sehen wir, dass bereits eine 1-Punkt Gaußquadratur für die Berechnung von $e_\ell^{(k)}$ zu einer $\mathcal{O}(N^{-3/2})$ -Konvergenzrate des Fehlerschätzers führt. Auch die Gegenüberstellung mit der 3-Punkt-Gaußquadratur-Berechnung in Abbildung 15 zeigt ein gutes Ergebnis. Dies ist insofern bemerkenswert, da der Steinbach-Schätzer $\sigma_{\ell,V}$ bei der Berechnung der gleichen Größe $e_\ell^{(k)}$ für $k = 1$ und $q = 0$ erst mit einer 5-Punkt Gaußquadratur einen vergleichbar stabilen Konvergenzverlauf aufweist.

Demnach werden wir für den Steinbach-Schätzer $\sigma_{\ell,A}$ eine 1-Punkt Gaußquadratur verwenden und als Parameter $k := 1$ wählen. Vereinfachend werden wir für die weiteren Simulationen $\sigma_{\ell,A}$ im fortlaufenden Dokument schreiben.

⁴Der Fall $k = 0$ ist aufgrund der Definition des Multilevel-Operators ausgeschlossen.


 Abbildung 10: Parameterschätzung $\sigma_{\ell,V}^{(k,q)}$ auf L-Shape (3-Punkt Gaußquadratur)

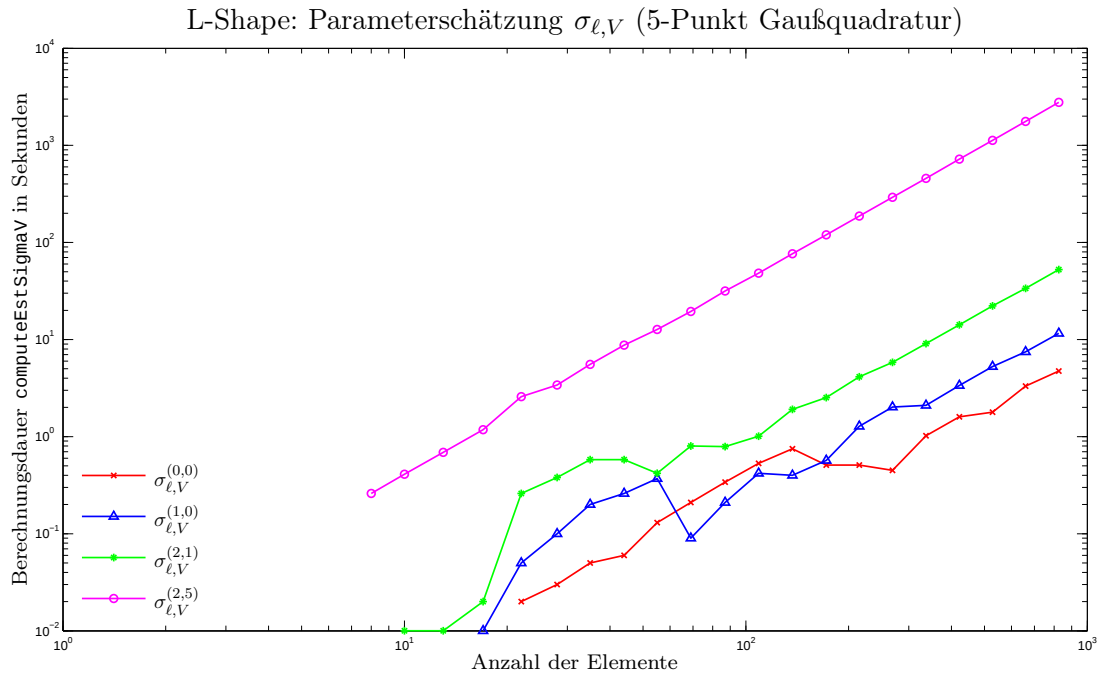
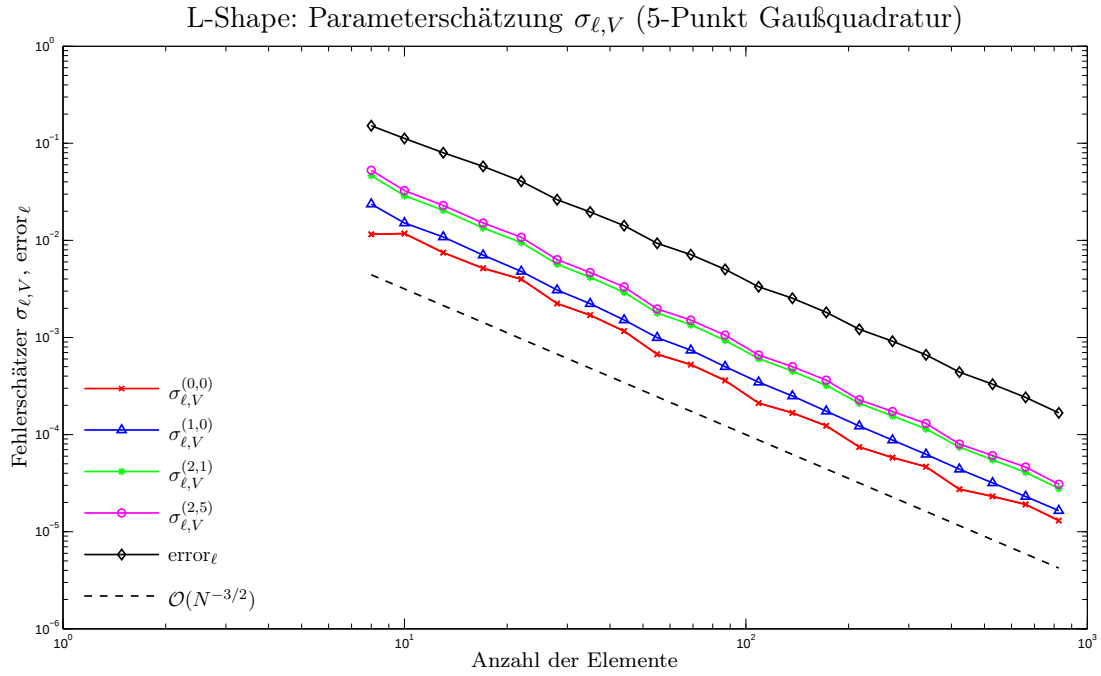


Abbildung 11: Parameterschätzung $\sigma_{\ell,V}^{(k,q)}$ auf L-Shape (5-Punkt Gaußquadratur)

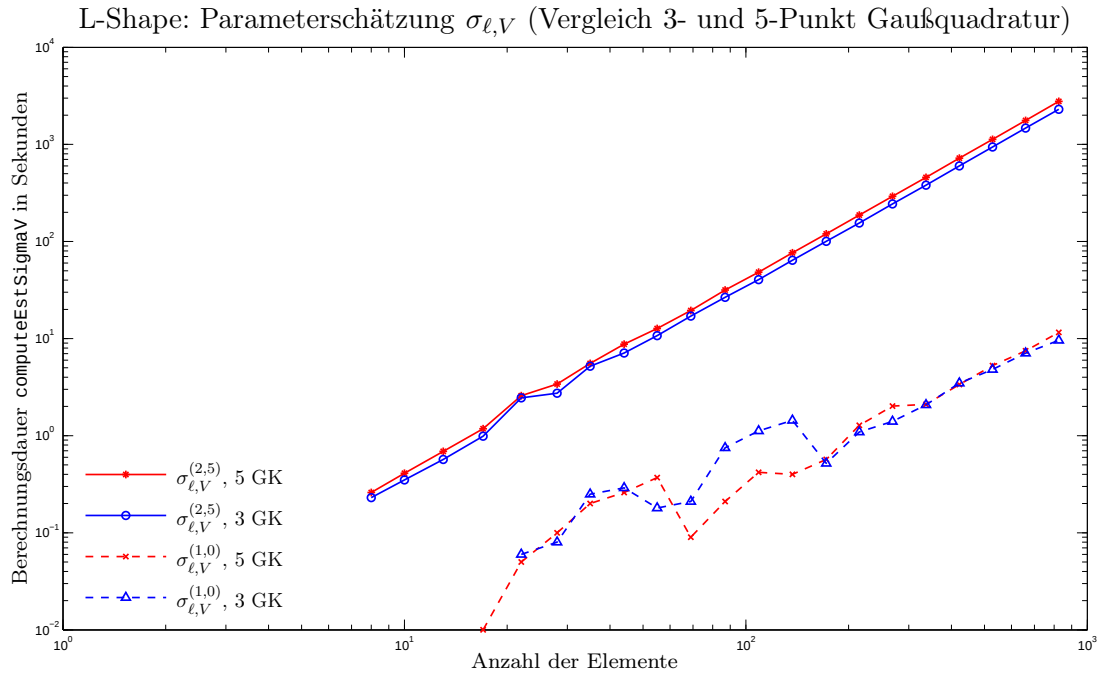
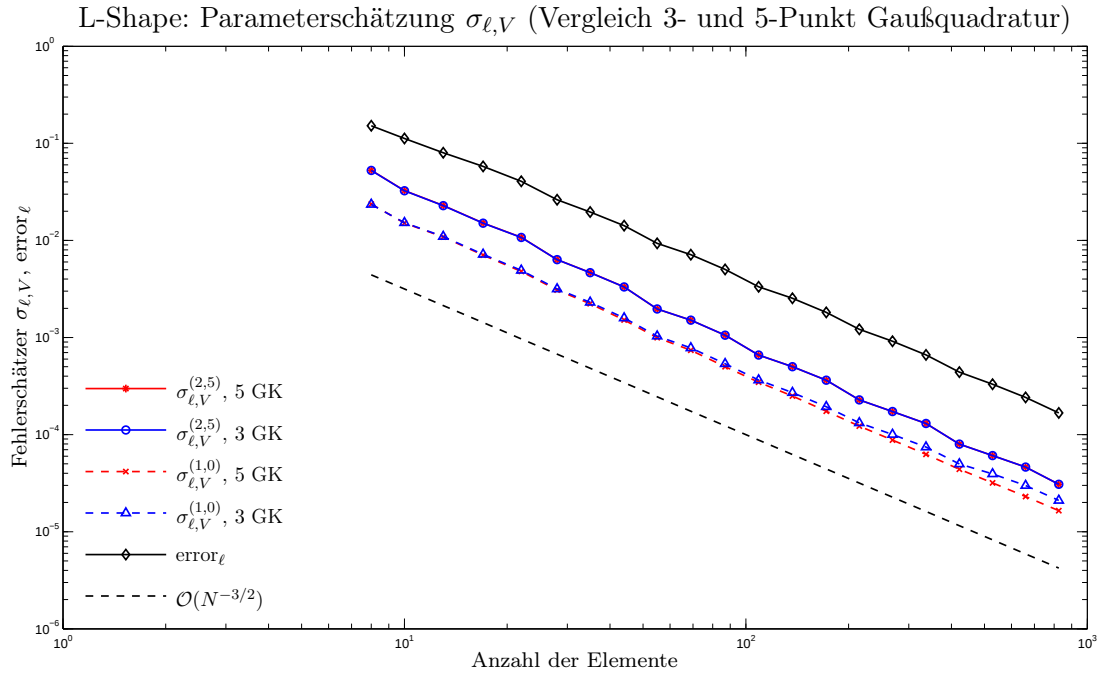


Abbildung 12: Parameterschätzung $\sigma_{\ell,V}^{(k,q)}$ auf L-Shape (Vergleich 3- und 5-Punkt Gaußquadratur)

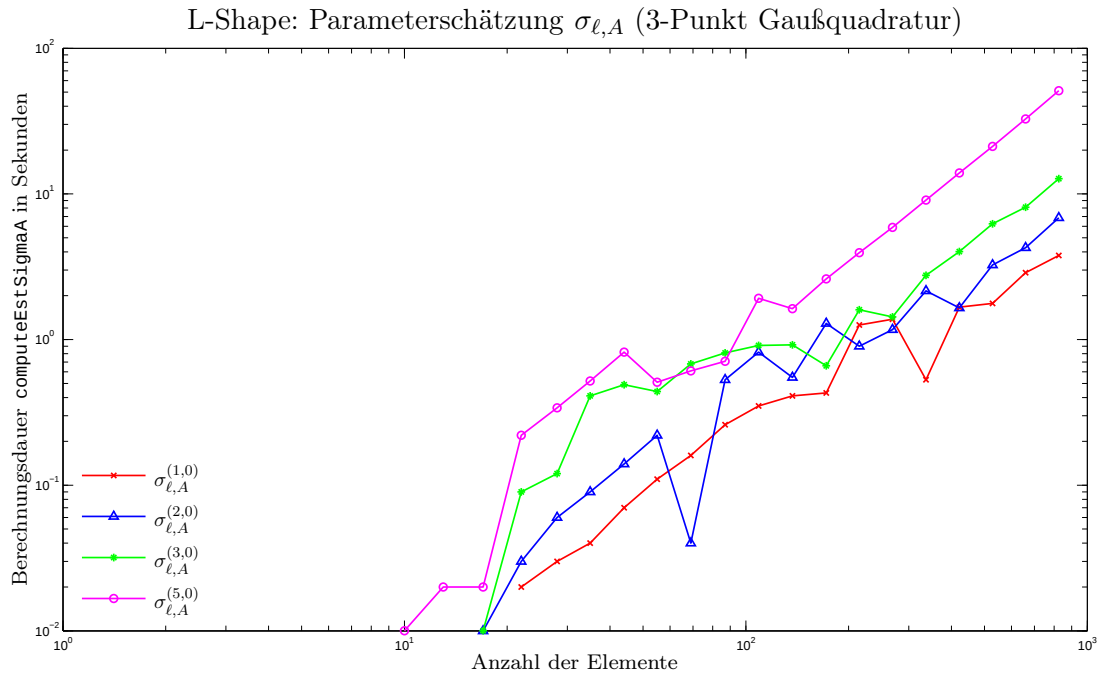
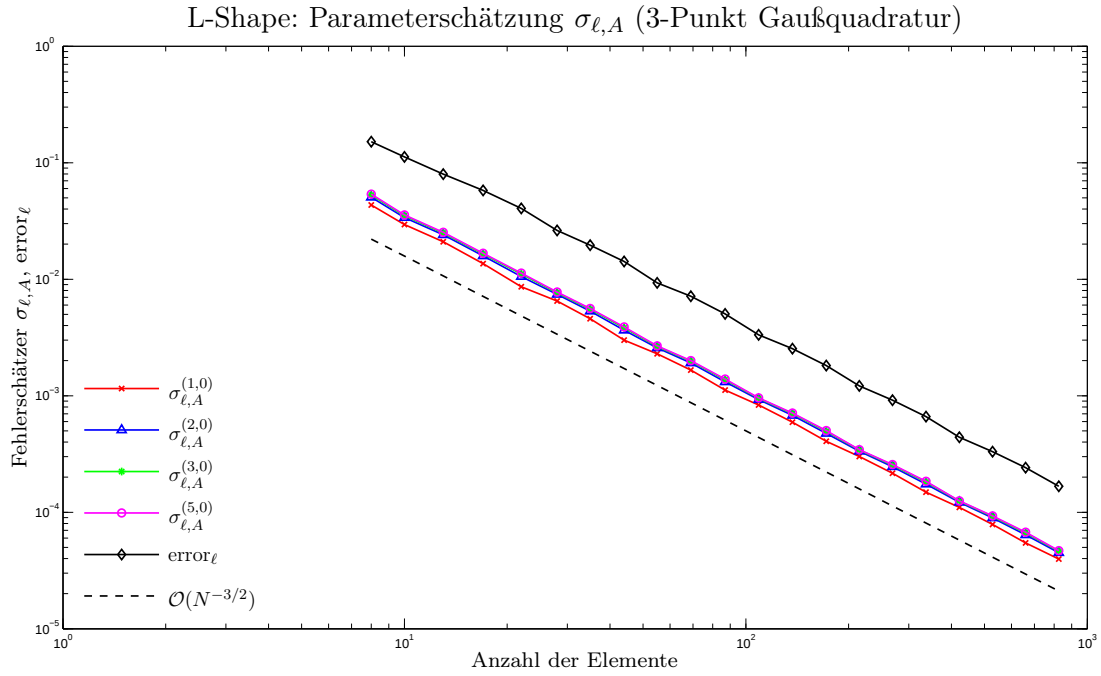
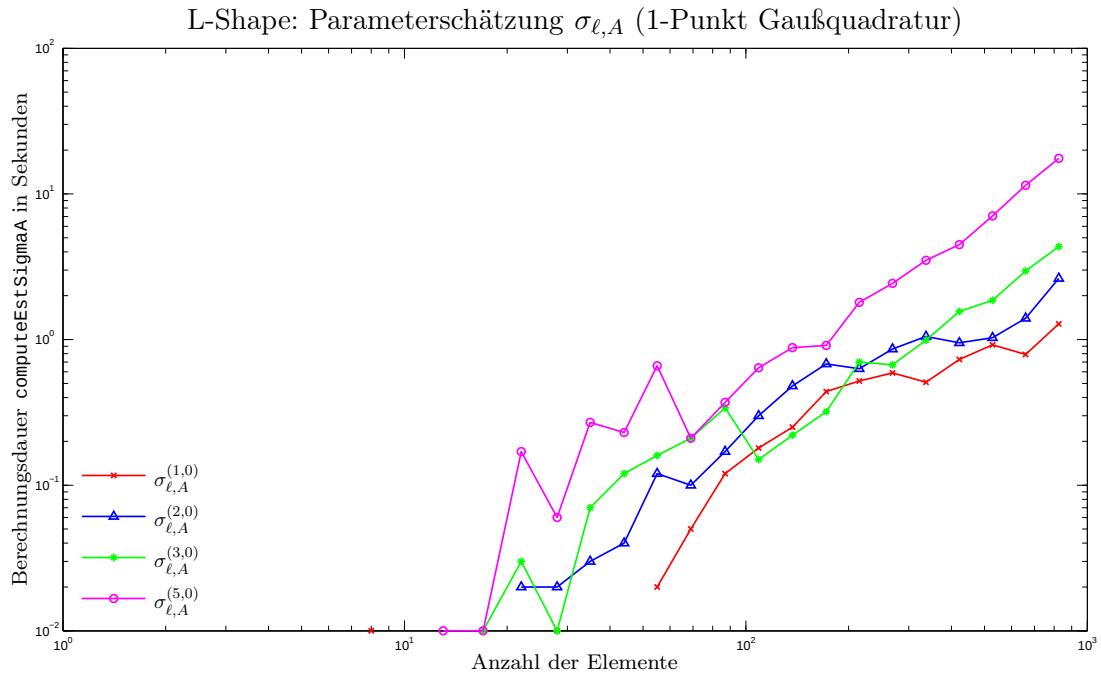
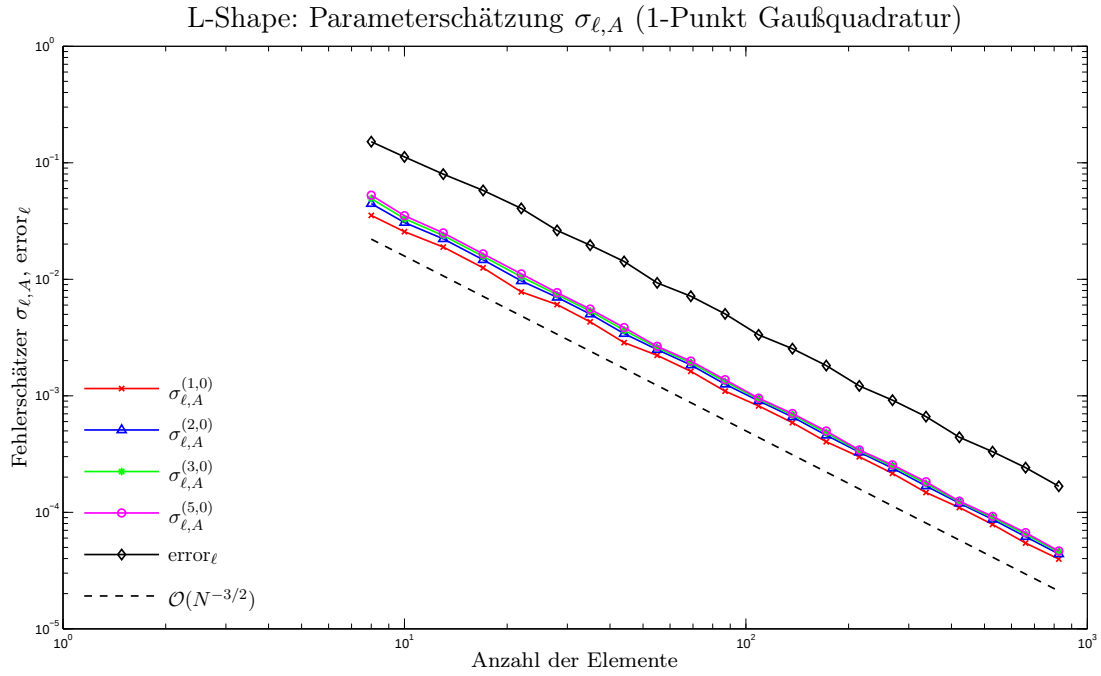


Abbildung 13: Parameterschätzung $\sigma_{\ell,A}^{(k,q)}$ auf L-Shape (3-Punkt Gaußquadratur)


 Abbildung 14: Parameterschätzung $\sigma_{\ell,A}^{(k,q)}$ auf L-Shape (1-Punkt Gaußquadratur)

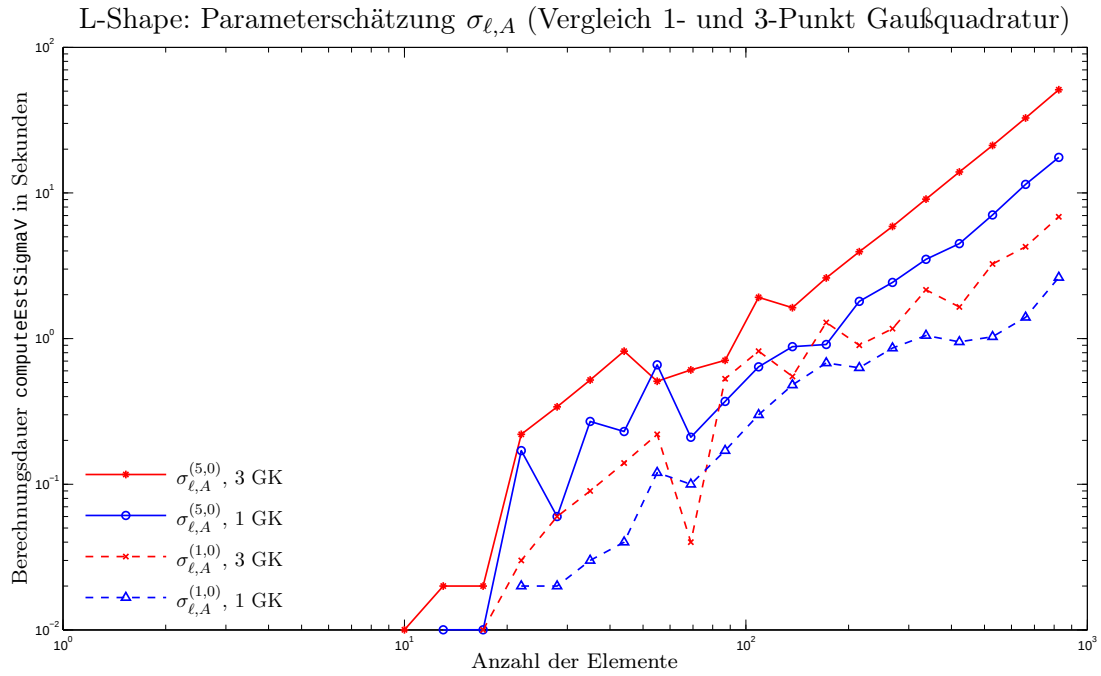
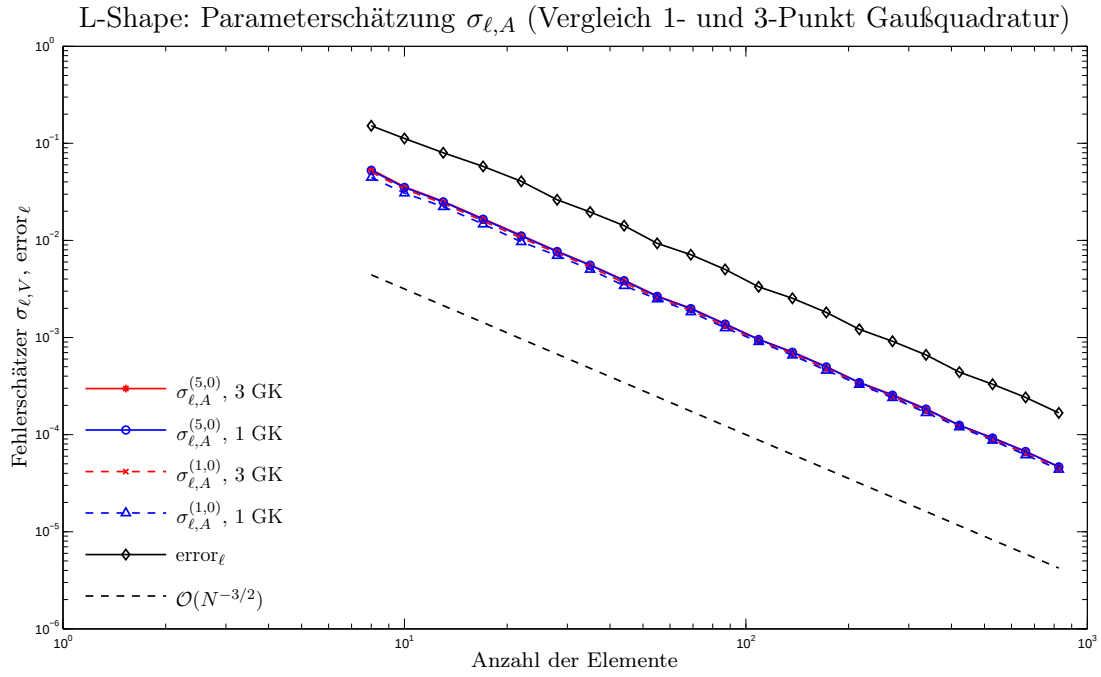


Abbildung 15: Parameterschätzung $\sigma_{\ell,A}^{(k,q)}$ auf L-Shape (Vergleich 1- und 3-Punkt Gaußquadratur)

6.2.3 Z-Shape: Parameterüberprüfung der Steinbach-Schätzer $\sigma_{\ell,V}, \sigma_{\ell,A}$

In diesem Abschnitt möchten wir die eben auf dem L-Shape gewählten Parameter der Steinbach-Schätzer auf ihre Brauchbarkeit bei Simulationen auf dem Z-Shape testen. Alle Plots wurden wieder bei $\tilde{\mu}_\ell$ -adaptiver Steuerung generiert:

In Abbildung 16 sehen wir, dass dies beim Steinbach-Schätzer über das Einfachschichtpotential $\sigma_{\ell,V}$ bei keiner gewählten Parametereinstellung der Fall ist. Diese schlagartig eintretende Ungenauigkeit tritt bei vermeintlich genaueren Berechnungen, d.h. mehr Neumann-Terme, höhere Verfeinerung der Triangulierung $\mathcal{E}_\ell$ und bei höherer Quadraturordnung, früher und stärker auf. Beispielsweise befindet sich der Fehlerschätzer $\sigma_{\ell,V}^{(2,5)}$ bei einer 3-Punkt Gaußquadratur, verglichen mit der 5-Punkt Gaußquadratur, einen Verfeinerungsschritt länger auf der „ $\mathcal{O}(N^{-3/2})$ -Konvergenzkurve“. Selbiges gilt für $\sigma_{\ell,V}^{(1,0)}$. Möglicherweise ist dieses Fehlverhalten auf eine instabile Berechnung des hypersingulären Integraloperators W und/oder des adjungierten Doppelschichtpotentialoperators K' bei singuläreren Daten zurückzuführen. Zum Einen erfolgt mit diesen Operatoren die Berechnung von

$$e_\ell^{(k,0)} = e_\ell^{(k)} = \Pi_\ell^{(k)}(\tilde{\phi}_\ell - \Phi_\ell) = \Pi_\ell^{(k)}((K' - 1/2)\Phi_\ell + WG_\ell)$$

und zum Anderen für $q > 0$

$$e_\ell^{(k,q)} = \sum_{\nu=0}^q (\Pi_\ell^{(k)}(1/2 + K'))^\nu e_\ell^{(k)}.$$

Je öfter die Integraloperatoren ausgewertet werden, desto unzuverlässiger scheint das Ergebnis. Eine oder mehrere Instabilitäten wären also eine mögliche Erklärung für die beobachteten Resultate.

Beim Steinbach-Schätzer über den Multilevel-Operator $\sigma_{\ell,A}^{(k,q)}$ zeigt sich ein ähnliches Bild in Abbildung 17, wenngleich solche Fehlberechnungen nur in abgeschwächter Form existieren – der Multilevel-Operator scheint diese Ungenauigkeiten zu glätten. Auch hier treten diese Sprünge bei niedrigerer Quadraturordnung tendentiell später auf.

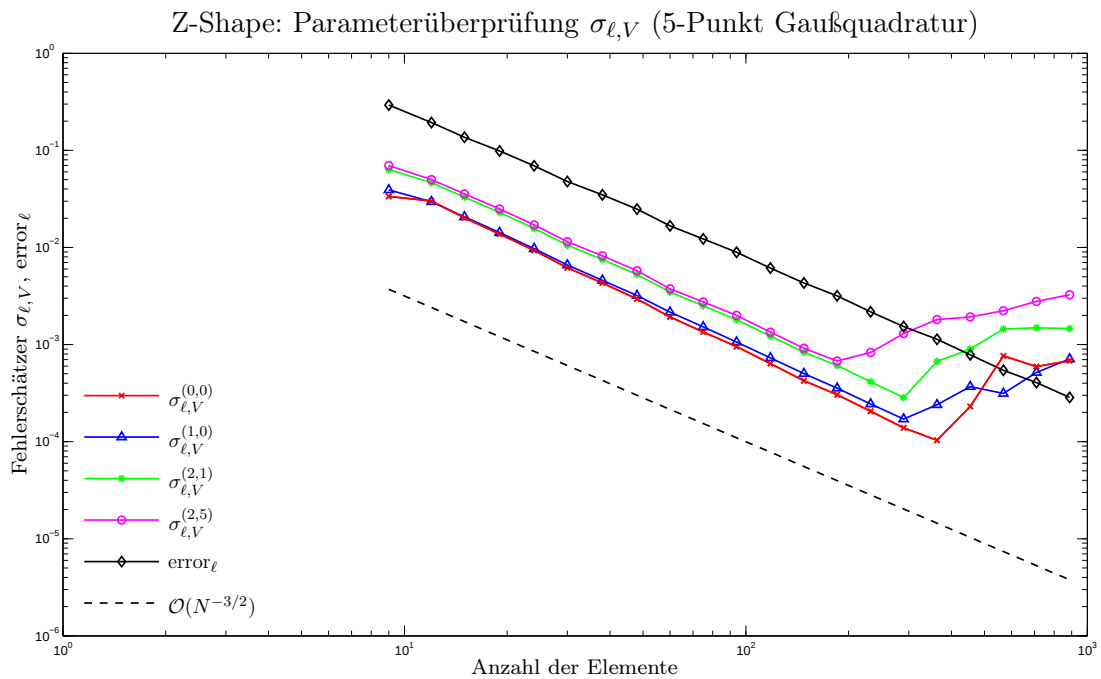
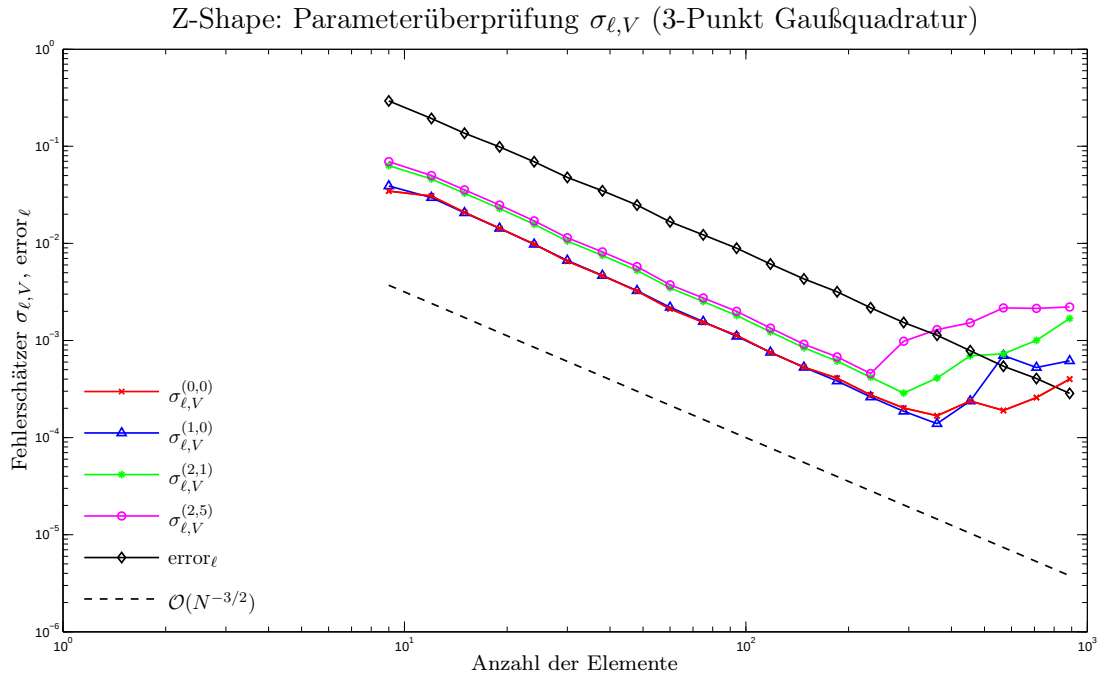
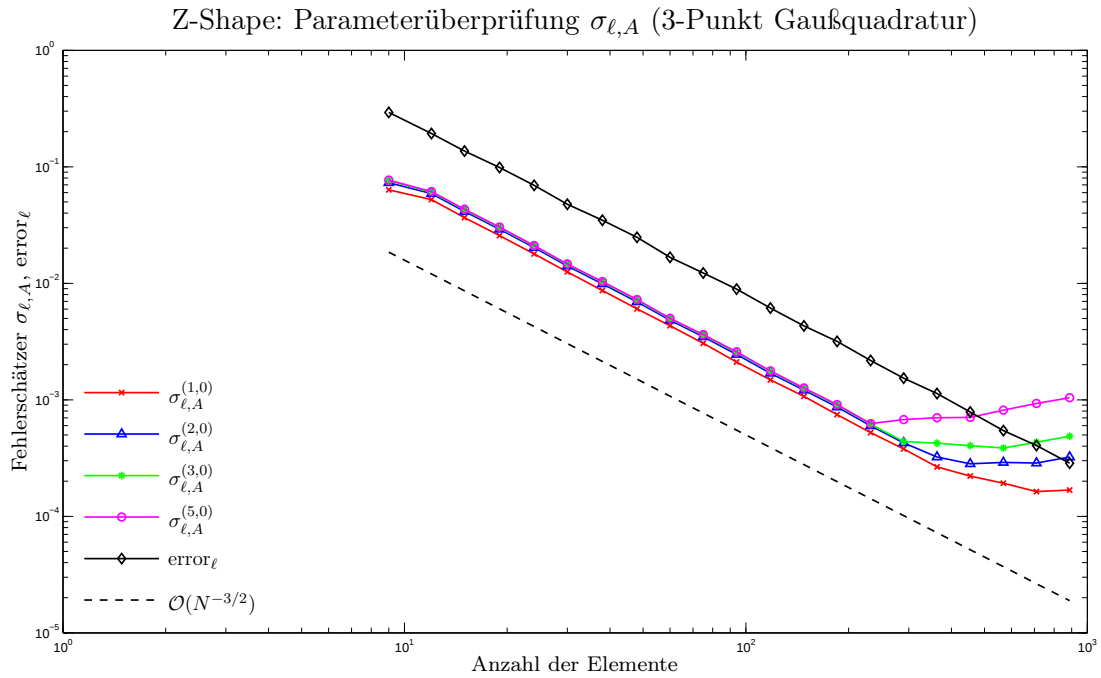
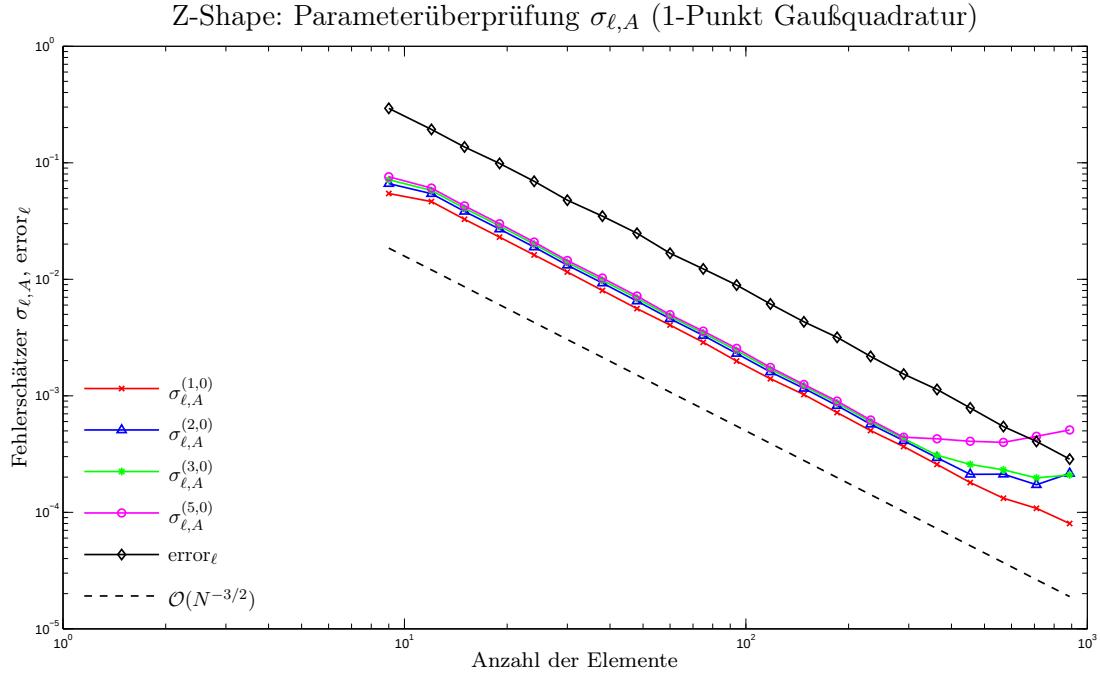


Abbildung 16: Parameterüberprüfung $\sigma_{\ell,V}^{(k,q)}$ auf Z-Shape (3- und 5-Punkt Gaußquadratur)


 Abbildung 17: Parameterüberprüfung $\sigma_{\ell,A}^{(k,q)}$ auf Z-Shape (1- und 3-Punkt Gaußquadratur)

6.2.4 Die Steinbach-Fehlerindikatoren und der Steinbach-Schätzer $\sigma_{\ell,V}$

Bevor wir uns mit der genauen Analyse der einzelnen Fehlerschätzer beschäftigen, noch ein Test mit dem Steinbach-Schätzer $\sigma_{\ell,V}$: Im Unterschied zu all den anderen Fehlerschätzern sind die Steinbach-Fehlerindikatoren

$$\sigma_{\ell,V}^{(k,q)}(E) = (|\langle V e_\ell^{(k,q)}, e_\ell^{(k,q)} \rangle_E|)^{1/2} = (|\langle e_\ell^{(k,q)}, e_\ell^{(k,q)} \rangle_E|)^{1/2} \quad \text{für } E \in \mathcal{E}_\ell$$

und der Steinbach-Schätzer

$$\sigma_{\ell,V}^{(k,q)} = (\langle V e_\ell^{(k,q)}, e_\ell^{(k,q)} \rangle_\Gamma)^{1/2} (\langle e_\ell^{(k,q)}, e_\ell^{(k,q)} \rangle_\Gamma)^{1/2}$$

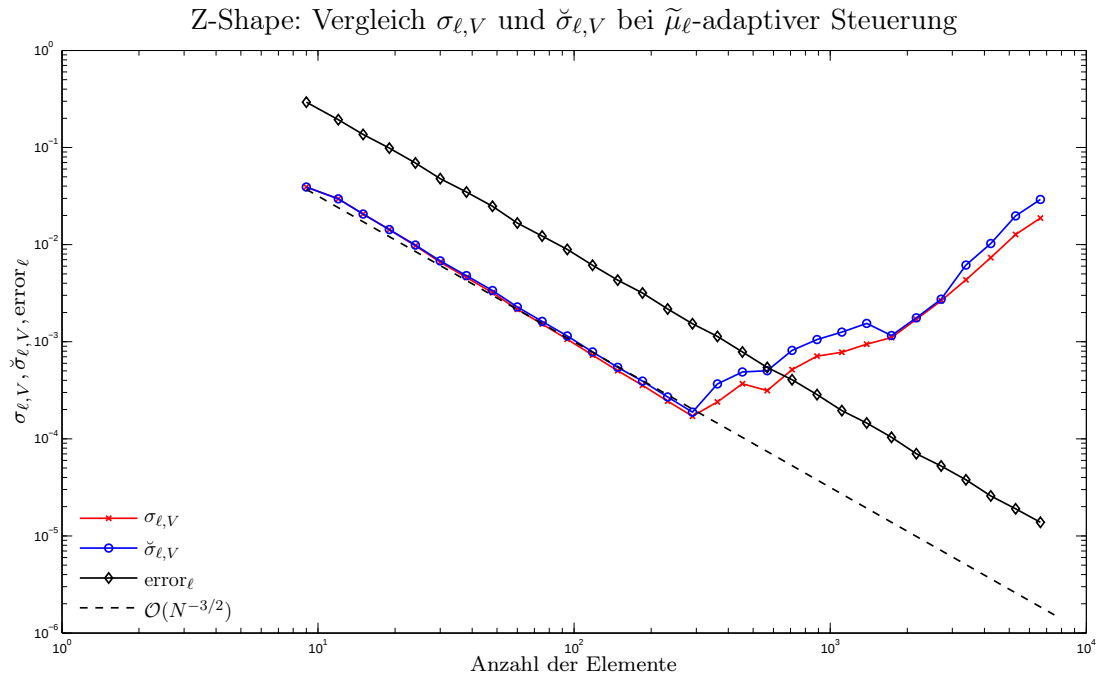
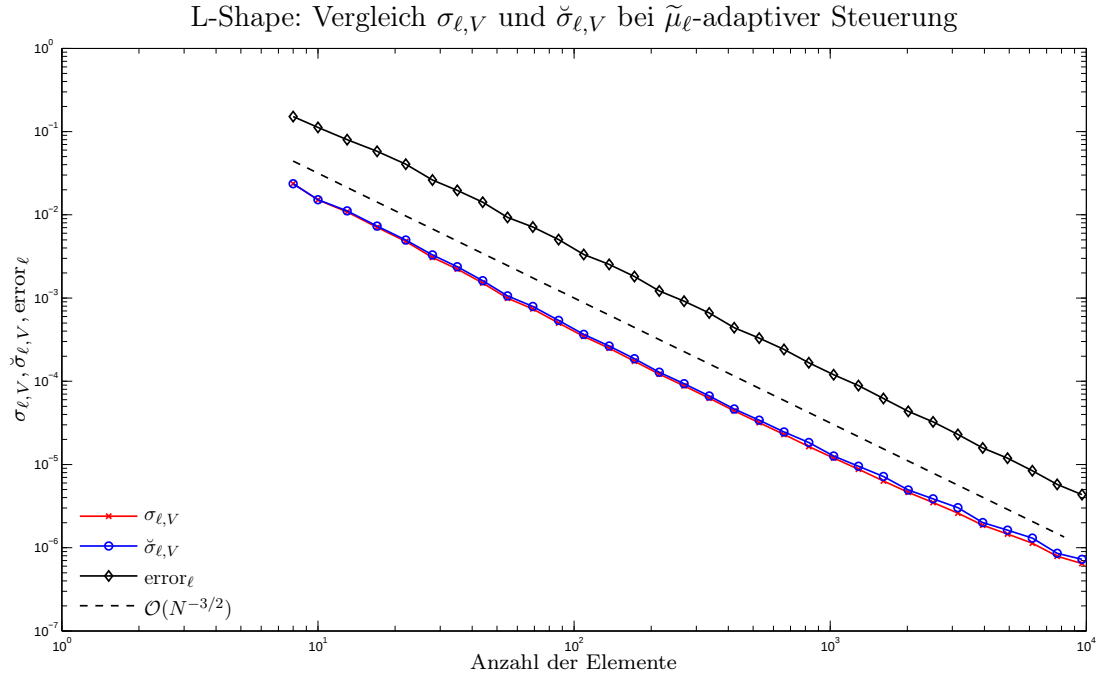
über das Einfachschichtpotential separat zu berechnen. Wie in Abschnitt 5.12.1 gezeigt wurde, gilt

$$\sigma_{\ell,V}^{(k,q)} \leq \left(\sum_{j=1}^N (\sigma_{\ell,V}^{(k,q)}(E_j))^2 \right)^{1/2} =: \check{\sigma}_{\ell,V}^{(k,q)}.$$

Die Beträge bei den Indikatoren garantieren dabei die positive Definitheit der symmetrischen Bilinearform $\langle \cdot, \cdot \rangle_E$ für alle $E \in \mathcal{E}_\ell$. Wir wollen nun untersuchen inwieweit sich die Ergebnisse von $\check{\sigma}_{\ell,V}$ zu jenen von $\sigma_{\ell,V}$ bei den oben gewählten Parametern, d.h. $q = 0$, $k = 1$ und 5-Punkt Gaußquadratur, unterscheiden.

Wir erhalten auf dem L-Shape mit dem $\tilde{\mu}_\ell$ -adaptiv gesteuerten BEM-Algorithmus die in Abbildung 18 oben dargestellten Simulationsergebnisse. Wie erwartet liegt $\sigma_{\ell,V}$ immer unter $\check{\sigma}_{\ell,V}$. Interessant ist allerdings, dass auch $\check{\sigma}_{\ell,V}$ einen effizienten und zuverlässigen Fehlerschätzer abzugeben scheint.

Die Ergebnisse auf dem Z-Shape sind allerdings bescheiden, vgl. Abbildung 18 unten. Nach obigen Erkenntnissen ist dies nicht überraschend.


 Abbildung 18: Vergleich des Steinbach-Schätzers $\sigma_{\ell,V}$ mit $\check{\sigma}_{\ell,V}$

6.3 Analyse der lokalen a-posteriori Fehlerschätzer

Nachdem nun alle Fehlerschätzer vollständig spezifiziert sind, können wir mit deren Analyse beginnen. Wir werden dabei alle lokalen a-posteriori Fehlerschätzer $\mu_\ell, \tilde{\mu}_\ell, \tau_\ell, \varrho_\ell, \varphi_\ell, \sigma_{\ell,V}, \sigma_{\ell,A}$ auf den Beispiel-Gebieten L-Shape, Z-Shape und auf dem Schlitz-Problem (hier ohne Steinbach-Schätzer) testen.

Auf jedem Gebiet werden wir zuerst die Regularität der Lösung $\phi \in H^{-1/2}(\Gamma)$ mittels Simulation bei uniformer Verfeinerung bestimmen. Für eine Funktion $u : \mathbb{R}^2 \rightarrow \mathbb{R}$ gilt $u \in H^s(\Gamma)$, $s \in \mathbb{R}_{\geq 0}$ auf dem Rand Γ genau dann, wenn die Sobolev-Slobodeckij-Norm

$$\|u\|_{H^s(\Gamma)}^2 := \|u\|_{H^k(\Gamma)}^2 + |u|_{H^\kappa(\Gamma)}^2$$

mit $s = k + \kappa$, wobei $k = \lfloor s \rfloor$ und

$$|u|_{H^\kappa(\Gamma)}^2 := \int_{\Gamma} \int_{\Gamma} \frac{|D^k u(x) - D^k u(y)|^2}{|x - y|^{1+2\kappa}} d\Gamma(x) d\Gamma(y),$$

endlich ist. Folglich gilt $u \in H^s(\Gamma)$ für den für uns interessanten Fall $s \in (0, 1)$ genau dann, wenn

$$\|u\|_{H^s(\Gamma)}^2 = \|u\|_{L^2(\Gamma)}^2 + |u|_{H^s(\Gamma)}^2 < \infty.$$

Wir zitieren nun Korollar 4.1.33 aus [15] zu einem Lemma über die Konvergenzordnung des Fehlers $\|\phi - \Phi_\ell\|_V$ für uniforme Netze.

Lemma 6.1 (Konvergenzordnung). *Sei $\phi \in H^s(\Gamma) \subseteq H^{-1/2}(\Gamma)$ für ein $s \geq 0$ die exakte Lösung der Variationsformulierung (3.8) und $\Phi_\ell \in \mathcal{P}^0(\mathcal{E}_\ell)$ die zur Triangulierung $\mathcal{E}_\ell$ gehörige Galerkin-Lösung. Dann gilt mit der uniformen Netzweite h auf $\mathcal{E}_\ell$ die Fehlerabschätzung*

$$\|\phi - \Phi_\ell\|_V \leq C h^{1/2+\min(s,1)} \|\phi\|_{H^s(\Gamma)} \quad (6.5)$$

für eine Konstante $C > 0$ abhängig von der K -Netzkonstante $\kappa(\mathcal{E}_\ell)$. $\square$

Bevor wir mit der Analyse beginnen, zeigen wir den schematischen Ablauf einer jeden adaptiven Verfeinerungsstrategie mit den Programmen aus HILBERT:

Algorithmus 6.2 (BEM-Algorithmus mit μ_ℓ -adaptiver Netzverfeinerung).

INPUT: Ausgangsnetz $\mathcal{E}_0$, $\theta = \rho = 1/4$, maximale Anzahl an Elementen $N_{max} \in \mathbb{N}$, Cauchy-Daten g und ϕ (ϕ für Fehlerberechnung), Zähler $\ell := 0$

- (1) Berechne uniforme Verfeinerung $\hat{\mathcal{E}}_\ell$ mittels `refineBoundaryMesh`.
- (2) Ermittle mit Hilfe von `computeOscDirichlet` die Dirichlet-Datenoszillationen $\widehat{\text{osc}}_{D,\ell}$ auf dem Netz $\hat{\mathcal{E}}_\ell$.
- (3) Bestimme die Dirichlet-Datenoszillationen $\text{osc}_{D,\ell}$ auf $\mathcal{E}_\ell$ mittels

$$\text{osc}_{D,\ell} = \widehat{\text{osc}}_{D,\ell}(\text{father2son}(:, 1)) + \widehat{\text{osc}}_{D,\ell}(\text{father2son}(:, 2)).$$

- (4) Berechne Einfachschichtpotentialmatrix $\widehat{\mathbf{V}}$ auf $\widehat{\mathcal{E}}_\ell$ durch `buildV`.
- (5) Ermittle rechte Seite $\widehat{\mathbf{b}}$ auf $\widehat{\mathcal{E}}_\ell$ durch `buildSymmRHS`.
- (6) Bestimme Koeffizientenvektor $\widehat{\mathbf{x}}$ für die Galerkin-Approximation $\widehat{\Phi}_\ell \in \mathcal{P}^0(\widehat{\mathcal{E}}_\ell)$ aus dem linearen Gleichungssystem $\widehat{\mathbf{V}}\widehat{\mathbf{x}} = \widehat{\mathbf{b}}$.
- (7) Berechne Einfachschichtpotentialmatrix $\mathbf{V}$ und rechte Seite $\mathbf{b}$, wobei $\mathbf{b}$ durch

$$\mathbf{b} = \widehat{\mathbf{b}}(\text{father2son}(:, 1)) + \widehat{\mathbf{b}}(\text{father2son}(:, 2)) \quad (6.6)$$

berechnet wird, für die Triangulierung $\mathcal{E}_\ell$ und bestimme den Koeffizientenvektor $\mathbf{x}$ für die Galerkin-Approximation $\Phi_\ell \in \mathcal{P}^0(\mathcal{E}_\ell)$.

- (8) Berechne mittels `computeEstSlpMu` den lokalen Fehlerschätzer μ_ℓ auf der Triangulierung $\mathcal{E}_\ell$.
- (9) Markiere die zu verfeinernden Elemente $\mathcal{M}_\ell$ von $\mathcal{E}_\ell$ mittels Verfeinerungsindikator $\iota_\ell^2 = \mu_\ell^2 + \text{osc}_{D,\ell}^2$ mit Hilfe von `markElements`.
- (10) Verfeinere markierte Elemente $\mathcal{M}_\ell$ mit `refineBoundaryMesh` und erhalte Netz $\mathcal{E}_{\ell+1}$.
- (11) Berechne alle anderen Fehlerschätzer und bestimme mittels `computeErrNeumann` den Neumann-Fehler $\text{err}_{N,\ell}$ auf $\mathcal{E}_\ell$.
- (12) Falls $\#\mathcal{E}_{\ell+1} < N_{\max}$, setze $\ell := \ell + 1$ und gehe zu (1).

OUTPUT: Fehlerschätzer $\mu_\ell, \widetilde{\mu}_\ell, \tau_\ell, \varrho_\ell, \varphi_\ell, \sigma_{\ell,V}, \sigma_{\ell,A}$, Summe $\check{\sigma}_{\ell,V} = (\sum_{j=1}^N \sigma_{\ell,V}(E_j)^2)^{1/2}$ der lokalen $\sigma_{\ell,V}$ -Fehleranteile, Dirichlet-Datenoszillationen $\text{osc}_{D,\ell}$, Neumann-Fehler $\text{err}_{N,\ell}$, Anzahl der Elemente $\#\mathcal{E}_\ell$ und kumulierte Berechnungsdauern aller Anteile der μ_ℓ -adaptiv gesteuerten Verfeinerungsstrategie.

Algorithmus 6.3 (BEM-Algorithmus mit $\widetilde{\mu}_\ell$ -adaptiver Netzverfeinerung).

INPUT: Ausgangsnetz $\mathcal{E}_0$, $\theta = 1/4$, maximale Anzahl an Elementen $N_{\max} \in \mathbb{N}$, Cauchy-Daten g und ϕ (ϕ für Fehlerberechnung), Zähler $\ell := 0$

- (1) Berechne uniforme Verfeinerung $\widehat{\mathcal{E}}_\ell$ mittels `refineBoundaryMesh`.
- (2) Ermittle mit Hilfe von `computeOscDirichlet` die Dirichlet-Datenoszillationen $\widehat{\text{osc}}_{D,\ell}$ auf dem Netz $\widehat{\mathcal{E}}_\ell$.
- (3) Bestimme die Dirichlet-Datenoszillationen $\text{osc}_{D,\ell}$ auf $\mathcal{E}_\ell$ mittels

$$\text{osc}_{D,\ell} = \widehat{\text{osc}}_{D,\ell}(\text{father2son}(:, 1)) + \widehat{\text{osc}}_{D,\ell}(\text{father2son}(:, 2)).$$

- (4) Berechne Einfachschichtpotentialmatrix $\widehat{\mathbf{V}}$ auf $\widehat{\mathcal{E}}_\ell$ durch `buildV`.
- (5) Ermittle rechte Seite $\widehat{\mathbf{b}}$ auf $\widehat{\mathcal{E}}_\ell$ durch `buildSymmRHS`.

- (6) Bestimme Koeffizientenvektor $\hat{\mathbf{x}}$ für die Galerkin-Approximation $\hat{\Phi}_\ell \in \mathcal{P}^0(\hat{\mathcal{E}}_\ell)$ aus dem linearen Gleichungssystem $\hat{\mathbf{V}}\hat{\mathbf{x}} = \hat{\mathbf{b}}$.
- (7) Berechne mittels `computeEstSlpMuTilde` den lokalen Fehlerschätzer $\tilde{\mu}_\ell$ auf der Triangulierung $\mathcal{E}_\ell$.
- (8) Markiere die zu verfeinernden Elemente $\mathcal{M}_\ell$ von $\mathcal{E}_\ell$ mittels Verfeinerungsindikator $\iota_\ell^2 = \tilde{\mu}_\ell^2 + \text{osc}_{D,\ell}^2$ mit Hilfe von `markElements`.
- (9) Verfeinere markierte Elemente $\mathcal{M}_\ell$ mit `refineBoundaryMesh` und erhalte Netz $\mathcal{E}_{\ell+1}$.
- (10) Berechne Einfachschichtpotentialmatrix $\mathbf{V}$ und rechte Seite $\mathbf{b}$, wobei $\mathbf{b}$ durch

$$\mathbf{b} = \hat{\mathbf{b}}(\text{father2son}(:, 1)) + \hat{\mathbf{b}}(\text{father2son}(:, 2))$$

berechnet wird, für die Triangulierung $\mathcal{E}_\ell$ und bestimme den Koeffizientenvektor $\mathbf{x}$ für die Galerkin-Approximation $\Phi_\ell \in \mathcal{P}^0(\mathcal{E}_\ell)$.

- (11) Berechne alle anderen Fehlerschätzer und bestimme mittels `computeErrNeumann` den Neumann-Fehler $\text{err}_{N,\ell}$ auf $\mathcal{E}_\ell$.
- (12) Falls $\#\mathcal{E}_{\ell+1} < N_{\max}$, setze $\ell := \ell + 1$ und gehe zu (1).

OUTPUT: Fehlerschätzer $\mu_\ell, \tilde{\mu}_\ell, \tau_\ell, \varrho_\ell, \varphi_\ell, \sigma_{\ell,V}, \sigma_{\ell,A}$, Summe $\check{\sigma}_{\ell,V}$ der lokalen $\sigma_{\ell,V}$ -Fehleranteile, Dirichlet-Datenoszillationen $\text{osc}_{D,\ell}$, Neumann-Fehler $\text{err}_{N,\ell}$, Anzahl der Elemente $\#\mathcal{E}_\ell$ und kumulierte Berechnungsdauern aller Anteile der $\tilde{\mu}_\ell$ -adaptiv gesteuerten Verfeinerungsstrategie.

Algorithmus 6.4 (BEM-Algorithmus mit τ_ℓ -adaptiver Netzverfeinerung).

INPUT: Ausgangsnetz $\mathcal{E}_0$, $\theta = \rho = 1/4$, maximale Anzahl an Elementen $N_{\max} \in \mathbb{N}$, Cauchy-Daten g und ϕ (ϕ für Fehlerberechnung), Zähler $\ell := 0$

- (1) Berechne uniforme Verfeinerung $\hat{\mathcal{E}}_\ell$ mittels `refineBoundaryMesh`.
- (2) Ermittle mit Hilfe von `computeOscDirichlet` die Dirichlet-Datenoszillationen $\widehat{\text{osc}}_{D,\ell}$ auf dem Netz $\hat{\mathcal{E}}_\ell$.
- (3) Bestimme die Dirichlet-Datenoszillationen $\text{osc}_{D,\ell}$ auf $\mathcal{E}_\ell$ mittels

$$\text{osc}_{D,\ell} = \widehat{\text{osc}}_{D,\ell}(\text{father2son}(:, 1)) + \widehat{\text{osc}}_{D,\ell}(\text{father2son}(:, 2)).$$

- (4) Berechne Einfachschichtpotentialmatrix $\hat{\mathbf{V}}$ auf $\hat{\mathcal{E}}_\ell$ durch `buildV`.
- (5) Ermittle rechte Seite $\hat{\mathbf{b}}$ auf $\hat{\mathcal{E}}_\ell$ durch `buildSymmRHS`.
- (6) Berechne Einfachschichtpotentialmatrix $\mathbf{V}$ und rechte Seite $\mathbf{b}$, wobei $\mathbf{b}$ durch

$$\mathbf{b} = \hat{\mathbf{b}}(\text{father2son}(:, 1)) + \hat{\mathbf{b}}(\text{father2son}(:, 2))$$

berechnet wird, für die Triangulierung $\mathcal{E}_\ell$ und bestimme den Koeffizientenvektor $\mathbf{x}$ für die Galerkin-Approximation $\Phi_\ell \in \mathcal{P}^0(\mathcal{E}_\ell)$.

- (7) Berechne mittels `computeEstSlpTau` den lokalen Fehlerschätzer τ_ℓ auf der Triangulierung $\mathcal{E}_\ell$.
- (8) Markiere die zu verfeinernden Elemente $\mathcal{M}_\ell$ von $\mathcal{E}_\ell$ mittels Verfeinerungsindikator $\iota_\ell^2 = \tau_\ell^2 + \text{osc}_{D,\ell}^2$ mit Hilfe von `markElements`.
- (9) Verfeinere markierte Elemente $\mathcal{M}_\ell$ mit `refineBoundaryMesh` und erhalte Netz $\mathcal{E}_{\ell+1}$.
- (10) Bestimme Koeffizientenvektor $\hat{\mathbf{x}}$ für die Galerkin-Approximation $\hat{\Phi}_\ell \in \mathcal{P}^0(\hat{\mathcal{E}}_\ell)$ aus dem linearen Gleichungssystem $\hat{\mathbf{V}}\hat{\mathbf{x}} = \hat{\mathbf{b}}$.
- (11) Berechne alle anderen Fehlerschätzer und bestimme mittels `computeErrNeumann` den Neumann-Fehler $\text{err}_{N,\ell}$ auf $\mathcal{E}_\ell$.
- (12) Falls $\#\mathcal{E}_{\ell+1} < N_{\max}$, setze $\ell := \ell + 1$ und gehe zu (1).

OUTPUT: Fehlerschätzer $\mu_\ell, \tilde{\mu}_\ell, \tau_\ell, \varrho_\ell, \varphi_\ell, \sigma_{\ell,V}, \sigma_{\ell,A}$, Summe $\check{\sigma}_{\ell,V}$ der lokalen $\sigma_{\ell,V}$ -Fehleranteile, Dirichlet-Datenoszillationen $\text{osc}_{D,\ell}$, Neumann-Fehler $\text{err}_{N,\ell}$, Anzahl der Elemente $\#\mathcal{E}_\ell$ und kumulierte Berechnungsdauern aller Anteile der τ_ℓ -adaptiv gesteuerten Verfeinerungsstrategie.

Abschließend gilt für die restlichen Fehlerschätzer mit $\varsigma_\ell \in \{\varrho_\ell, \varphi_\ell, \sigma_{\ell,A}, \sigma_{\ell,V}\}$:

Algorithmus 6.5 (BEM-Algorithmus mit ς_ℓ -adaptiver Netzverfeinerung).

INPUT: Ausgangsnetz $\mathcal{E}_0$, $\theta = \rho = 1/4$, maximale Anzahl an Elementen $N_{\max} \in \mathbb{N}$, Cauchy-Daten g und ϕ (ϕ für Fehlerberechnung), Zähler $\ell := 0$

- (1) Ermittle mit Hilfe von `computeOscDirichlet` die Dirichlet-Datenoszillationen $\text{osc}_{D,\ell}$ auf dem Netz $\mathcal{E}_\ell$.
- (2) Berechne $\mathbf{b}, \mathbf{V}$ auf $\mathcal{E}_\ell$ und ermittle den Koeffizientenvektor $\mathbf{x}$ für Galerkin-Lösung $\Phi_\ell \in \mathcal{P}^0(\mathcal{E}_\ell)$.
- (3) Bestimme mittels `computeEstSlpRho`, `computeEstSlpFaermann`, `computeEstSigmaA` bzw. `computeEstSigmaV` den lokalen Fehlerschätzer ς_ℓ auf der Triangulierung $\mathcal{E}_\ell$.
- (4) Markiere die zu verfeinernden Elemente $\mathcal{M}_\ell$ von $\mathcal{E}_\ell$ mittels Verfeinerungsindikator $\iota_\ell^2 = \varsigma_\ell^2 + \text{osc}_{D,\ell}^2$ mit Hilfe von `markElements`.
- (5) Verfeinere markierte Elemente $\mathcal{M}_\ell$ mit `refineBoundaryMesh` und erhalte Netz $\mathcal{E}_{\ell+1}$.
- (6) Berechne uniforme Verfeinerung $\hat{\mathcal{E}}_\ell$.
- (7) Bestimme $\hat{\mathbf{b}}, \hat{\mathbf{V}}$ auf $\hat{\mathcal{E}}_\ell$ und ermittle den Koeffizientenvektor $\hat{\mathbf{x}}$ für Galerkin-Lösung $\hat{\Phi}_\ell \in \mathcal{P}^0(\hat{\mathcal{E}}_\ell)$.
- (8) Berechne alle anderen Fehlerschätzer und bestimme mittels `computeErrNeumann` den Neumann-Fehler $\text{err}_{N,\ell}$ auf $\mathcal{E}_\ell$.
- (9) Falls $\#\mathcal{E}_{\ell+1} < N_{\max}$, setze $\ell := \ell + 1$ und gehe zu (1).

OUTPUT: Fehlerschätzer $\mu_\ell, \tilde{\mu}_\ell, \tau_\ell, \varrho_\ell, \varphi_\ell, \sigma_{\ell,V}, \sigma_{\ell,A}$, Summe $\check{\sigma}_{\ell,V}$ der lokalen $\sigma_{\ell,V}$ -Fehleranteile, Dirichlet-Datenoszillationen $\text{osc}_{D,\ell}$, Neumann-Fehler $\text{err}_{N,\ell}$, Anzahl der Elemente $\#\mathcal{E}_\ell$ und kumulierte Berechnungsdauern aller Anteile der ς_ℓ -adaptiv gesteuerten Verfeinerungsstrategie.

Im Anhang ist mit Listing A.2 der Programmcode für den $\sigma_{\ell,V}$ -adaptiv gesteuerten BEM-Algorithmus für L- und Z-Shape zu finden. Alle anderen Fehlerschätzer wurden analog und nach den Algorithmen 6.2 bis 6.5 implementiert. Beim Schlitz-Problem sind die in Abschnitt 6.1.3 erläuterten Änderungen zu beachten.

6.3.1 L-Shape

Für die Bestimmung der Regularität des Neumann-Datums ϕ , gegeben durch (6.3) mit (6.2) auf dem L-Shape aus Abbildung 5, berechnen wir die Galerkin-Lösung Φ_ℓ bei uniformer Verfeinerung und erhalten für error_ℓ und den Fehlerschätzern das in Abbildung 19 dargestellte Ergebnis. Die Konvergenzordnung $\mathcal{O}(N^{-\text{order}})$ des Fehlers $\|\phi - \Phi_\ell\|_V$ lässt sich durch

$$\text{order} = \frac{\log(\text{error}_{\ell+1}) - \log(\text{error}_\ell)}{\log(N_\ell) - \log(N_{\ell+1})} (> 0) \quad (6.7)$$

berechnen, wobei für uniforme Netze $h = N^{-1}$ gilt. Demnach sehen wir eine Fehler-Konvergenzrate von $\mathcal{O}(h^{2/3})$ und erhalten mit (6.5) $s = 1/6$, d.h. laut Simulation gilt $\phi \in H^{1/6}(\Gamma)$.

In den Abbildungen 20 bis 26 sehen wir alle Fehlerschätzer und die obere Schranke error_ℓ des Fehlers $\|\phi - \Phi_\ell\|_V$ bei je einer adaptiven Steuerung. Dabei weist error_ℓ und damit $\|\phi - \Phi_\ell\|_V$ unabhängig von der Wahl des Verfeinerungsindikators eine Konvergenzordnung von $\mathcal{O}(N^{-3/2})$ auf. Folglich gilt aufgrund der Definition (6.1) von error_ℓ auch für die Dirichlet-Datenoszillationen $\text{osc}_{D,\ell} = \mathcal{O}(N^{-3/2})$. Aus der in dieser Arbeit herausgearbeiteten Analysis war dies nur für uniforme Netze und hinreichend glatten Cauchy-Daten garantiert, vgl. die Sätze 3.3 und 3.6.

Aufgrund der Parallelität aller Kurven zu error_ℓ und damit zum Fehler $\|\phi - \Phi_\ell\|_V$ scheinen alle Fehlerschätzer effizient und zuverlässig zu sein. Einzig der Fehlerschätzer $\sigma_{\ell,V}$ erweckt den Eindruck einer abflachenden Tendenz bei μ_ℓ - und $\tilde{\mu}_\ell$ -adaptiver Steuerung im Bereich großer Elementanzahl. Betrachten wir den Steinbach-Schätzer $\sigma_{\ell,V}$ in Abbildung 33 bei allen adaptiven Verfeinerungsstrategien, so lässt sich dies nicht (mit Gewissheit) bestätigen. Damit weisen alle Fehlerschätzer einen zu $\|\phi - \Phi_\ell\|_V$ parallelen $\mathcal{O}(N^{-3/2})$ -Konvergenzverlauf auf. Aus den übrigen Abbildungen 27 bis 32 schließen wir auf die stabile Implementierung sämtlicher Fehlerschätzer.

Für die $(h - h/2)$ -Fehlerschätzer $\mu_\ell, \tilde{\mu}_\ell$ und τ_ℓ bedeutet deren beobachtete Effizienz und Zuverlässigkeit (bis auf Datenoszillationen), dass die Simulation die Saturationsannahme (3.34) numerisch bestätigt, vgl. Satz 5.7. Weiters verspricht Satz 5.9 die Effizienz des gewichteten Residualschätzers ϱ_ℓ nur bei quasi-uniformen Netzen. Im Abschnitt 5.10 konnten wir zwar mit Satz 5.13 die Effizienz und Zuverlässigkeit für den Faermann-Residualschätzer φ_ℓ nachweisen, doch fehlt die für die Implementierung notwendige Quadraturfehler-Analyse als Beweis für eine brauchbare Approximation $\tilde{\varphi}_\ell \approx \varphi_\ell$ der Realisierung. Mit der Simulation wird die Implementierung durch das numerische Experiment gestützt. Satz 5.18 garantiert die Effizienz und Zuverlässigkeit des Steinbach-Schätzers $\sigma_{\ell,V}$ nur bei (nicht spezifiziertes) hinreichend kleines $\|h_\ell^{(k)}\|_{L^\infty(\Gamma)} < \|h_\ell\|_{L^\infty(\Gamma)}$. Scheinbar reicht die in Abschnitt 6.2 begründete Parameterwahl dafür aus. Beim

Steinbach-Schätzer über den Multilevel-Operator $\sigma_{\ell,A}$ musste für Satz 5.21 noch zusätzlich von uniformen Netzen ausgegangen werden.

Damit ist die Effizienz und Zuverlässigkeit aller Fehlerschätzer, zumindest auf dem L-Shape, experimentell nachgewiesen.

Aufgrund der beobachteten Stabilität aller Implementierungen reicht es, die Fehlerschätzer bei eigener Adaptivität zu untersuchen. Abbildung 34 kann dann als Zusammenfassung der Abbildungen 20 bis 33 angesehen werden, wobei auch eine Gegenüberstellung hinsichtlich der kumulierten Berechnungszeiten des jeweiligen adaptiven-Algorithmus gegeben ist.

Die Abbildung 35 oben zeigt die Schranke error_ℓ bei allen adaptiven Steuerungen. Abgesehen von $\sigma_{\ell,V}$, der kontinuierlich einen etwas höheren Wert bezogen auf eine feste Elementanzahl aufweist, ist zwischen den Implementierungen hinsichtlich der Fehlerschranke kaum ein Unterschied feststellbar. Die (vermeintlich) genauesten Ergebnisse bzgl. der Energienorm liefern bei näherer Betrachtung die $(h - h/2)$ -Fehlerschätzer $\mu_\ell, \tilde{\mu}_\ell, \tau_\ell$, dicht gefolgt von den Residualschätzern $\varrho_\ell, \varphi_\ell$ und dem Steinbach-Schätzer über den Multilevel-Operator $\sigma_{\ell,A}$. Bei der Wahl eines Residualschätzer-gesteuerten oder $\sigma_{\ell,A}$ -gesteuerten adaptiven Algorithmus kann demnach innerhalb geringerer Zeit eine höhere Genauigkeit garantiert werden, vgl. Abbildung 35 unten. Die Berechnungszeit der Steinbach-Schätzer ist allerdings sehr stark von den gewählten Parametern abhängig, wie wir in Abschnitt 6.2 gesehen haben.

Zum Schluss wollen wir eine genaue Rechenzeit-Analyse der verschiedenen gesteuerten adaptiven Algorithmen durchführen und deren anteilige Zusammensetzung erörtern. In Abbildung 36 sind die kumulierten Berechnungszeiten des jeweiligen adaptiven BEM-Algorithmus über der Anzahl der Elemente aufgetragen. Wir sehen, dass sich der Zeitaufwand für die Berechnung der Algorithmen der verschiedenen Verfeinerungsstrategien asymptotisch wie $\mathcal{O}(N^2)$ zur Anzahl der Elemente verhält. Der $\sigma_{\ell,A}$ -gesteuerte Algorithmus ist etwas vor den beiden Residualschätzer-adaptiven Verfeinerungsstrategien und liefert daher die schnellsten Resultate, während die $(h - h/2)$ -Fehlerschätzer $\mu_\ell, \tilde{\mu}_\ell$ und τ_ℓ die längste Zeit für die Berechnung je Triangulierung brauchen. Der Vergleich der absoluten Zeit bei 5000 Elementen von etwa 2000 Sekunden bei den $(h - h/2)$ -Algorithmen gegenüber knapp 800 Sekunden bei jenen der Residualschätzern macht dies deutlich.

Die Abbildungen 37 bis 43 zeigen jeweils die absoluten und relativen Berechnungszeiten der verschieden gesteuerten BEM-Algorithmen, wobei alle Zeiten wieder als kumulierte Berechnungsdauern zu verstehen sind.

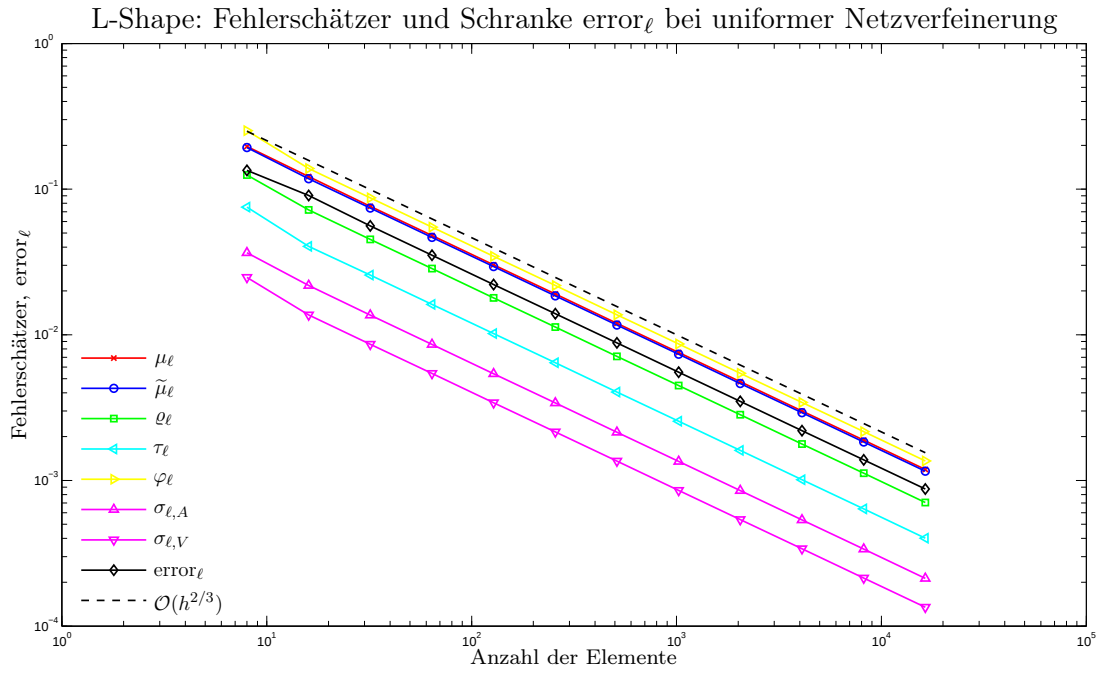
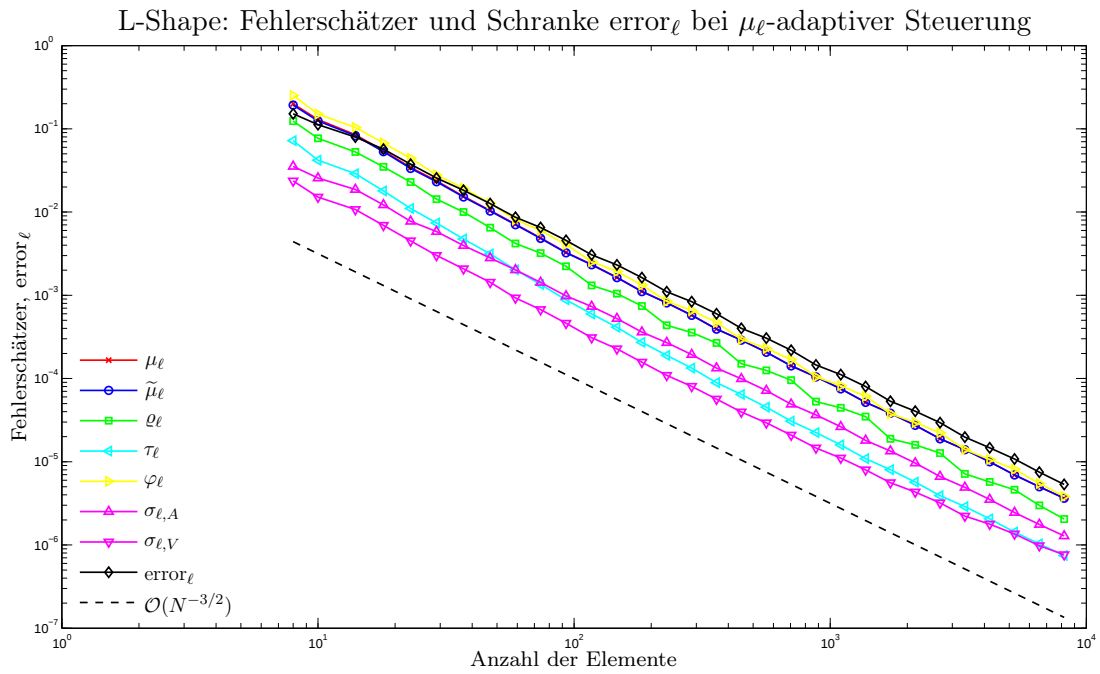
Bei den $(h - h/2)$ -adaptiven Algorithmen fällt der Hauptteil bei der μ_ℓ -Steuerung auf die Berechnung der rechten Seite $\hat{\mathbf{b}}$, da $\mathbf{b}$ aus $\hat{\mathbf{b}}$ mittels (6.6) gewonnen wird und der damit verbundene Zeitaufwand vernachlässigbar gering ist. Weiters ist auch das Aufstellen der Einfachschichtpotentialmatrizen $\mathbf{V}, \hat{\mathbf{V}}$ und das Lösen der Gleichungssysteme $\mathbf{V}\mathbf{x} = \mathbf{b}$ bzw. $\hat{\mathbf{V}}\hat{\mathbf{x}} = \hat{\mathbf{b}}$ teuer. Analog verhält es sich bei der $\tilde{\mu}_\ell$ -adaptiven Steuerung, wobei die Berechnung der Einfachschichtpotential $\mathbf{V}$ und das Lösen nach $\mathbf{x}$ entfällt. Folglich ist die $\tilde{\mu}_\ell$ -adaptive Steuerung etwas schneller. Bei der τ_ℓ -Steuerung werden zwar beide Einfachschichtpotentialmatrizen $\mathbf{V}, \hat{\mathbf{V}}$ und beide rechten Seiten $\mathbf{b}, \hat{\mathbf{b}}$ benötigt, doch muss nur mehr das Gleichungssystem $\mathbf{V}\mathbf{x} = \mathbf{b}$ gelöst werden, sodass insgesamt wieder etwas Zeit eingespart wird. Bei allen $(h - h/2)$ -Fehlerschätzern sind die Berechnungszeiten der Funktionen `computeEstSlpMu`, `computeEstSlpMuTilde` und `computeEstSlpTau` nicht der Rede wert.

Die Zeitanalyse der Residualschätzer-gesteuerten Algorithmen ist nahezu identisch. Es werden

nur die Galerkin-Daten $\mathbf{b}, \mathbf{V}$ benötigt um den Koeffizientenvektor $\mathbf{x}$ der Galerkin-Lösung Φ_ℓ zu bestimmen. Im Gegensatz zu den $(h - h/2)$ -Strategien sind die Berechnungen der Fehlerschätzer ϱ_ℓ und φ_ℓ in Form von `computeEstSlpRho` und `computeEstSlpFaermann` komplexer und damit auch deutlich zeitintensiver. Besonders bei einer geringeren Elementanzahl fällt dies stärker ins Gewicht.

BEM-Algorithmen mit einer Steinbach-Schätzer-Steuerung lassen sich ähnlich den Residualschätzern charakterisieren. Auch hier werden nur die Galerkin-Daten $\mathbf{b}, \mathbf{V}$ für die Bestimmung des Koeffizientenvektors $\mathbf{x}$ benötigt. Allerdings ist die Funktionsauswertung `computeEstSigmaA` im Vergleich zu jener von `computeEstSigmaV` sehr schnell, woraus sich dann auch die insgesamt schnellere Berechnung erklärt.

Zusammenfassend lässt sich sagen, dass mit den $(h - h/2)$ -adaptiven BEM-Algorithmen die genauesten Galerkin-Approximationen bezogen auf die Elementanzahl bzgl. der Energienorm garantiert werden können, dicht gefolgt von den Residualschätzern $\varrho_\ell, \varphi_\ell$ und dem Steinbach-Schätzer über den Multilevel-Operator $\sigma_{\ell,A}$. Der Vorteil der Residualschätzer liegt dann klar bei der schnelleren Berechnungszeit. Die Steinbach-Schätzer und deren Berechnungsdauer sind sehr stark von den zugehörigen Parametern abhängig und folglich gut überlegt zu wählen. Bei unserer sehr einfachen Wahl der Parameter, d.h. $q = 0$, $k = 1$ und 1-Punkt Gaußquadratur, ist der $\sigma_{\ell,A}$ -gesteuerte Algorithmus sogar der schnellste. Weiters liefert der adaptive BEM-Algorithmus bei allen untersuchten adaptiven Verfeinerungsstrategien die maximal mögliche Konvergenzrate des Fehlers $\|\phi - \Phi_\ell\|_V$ von $\mathcal{O}(N^{-3/2})$.


 Abbildung 19: L-Shape: Fehlerschätzer und Schranke error_ℓ bei uniformer Netzverfeinerung

 Abbildung 20: L-Shape: Fehlerschätzer und Schranke error_ℓ bei μ_ℓ -adaptiver Steuerung

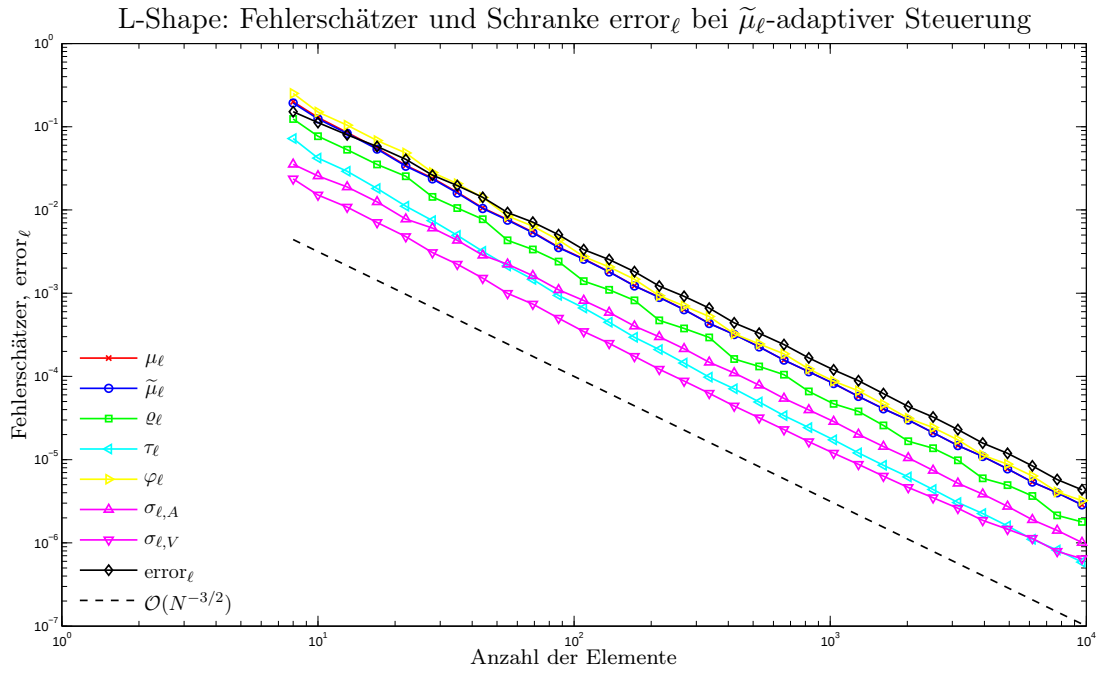


Abbildung 21: L-Shape: Fehlerschätzer und Schranke error_ℓ bei $\tilde{\mu}_\ell$ -adaptiver Steuerung

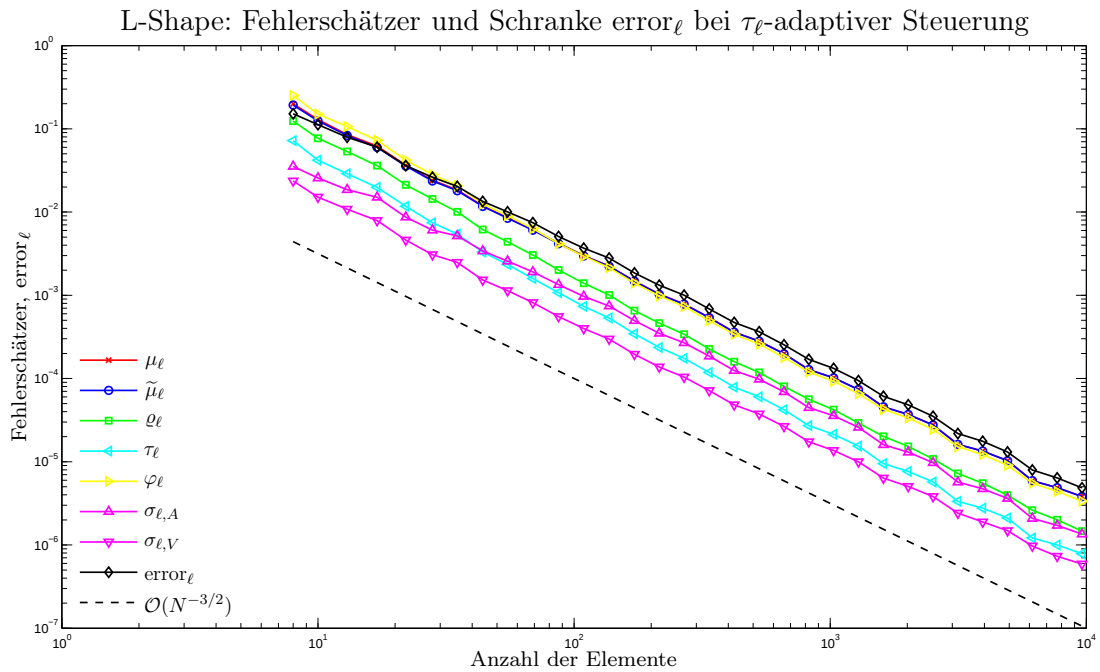
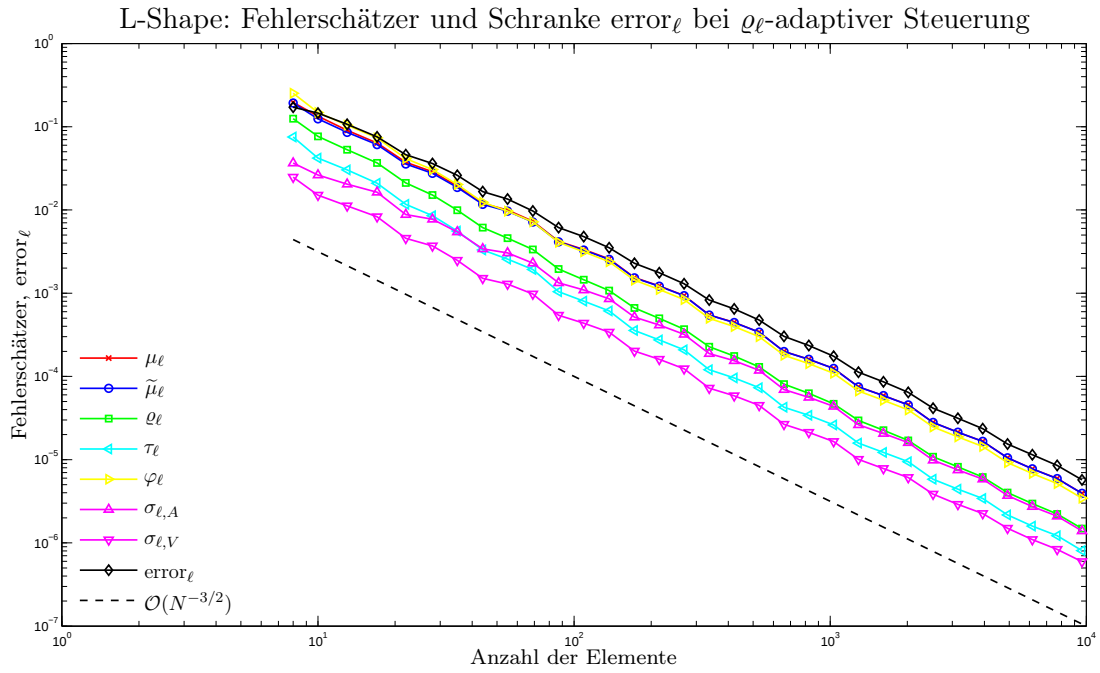
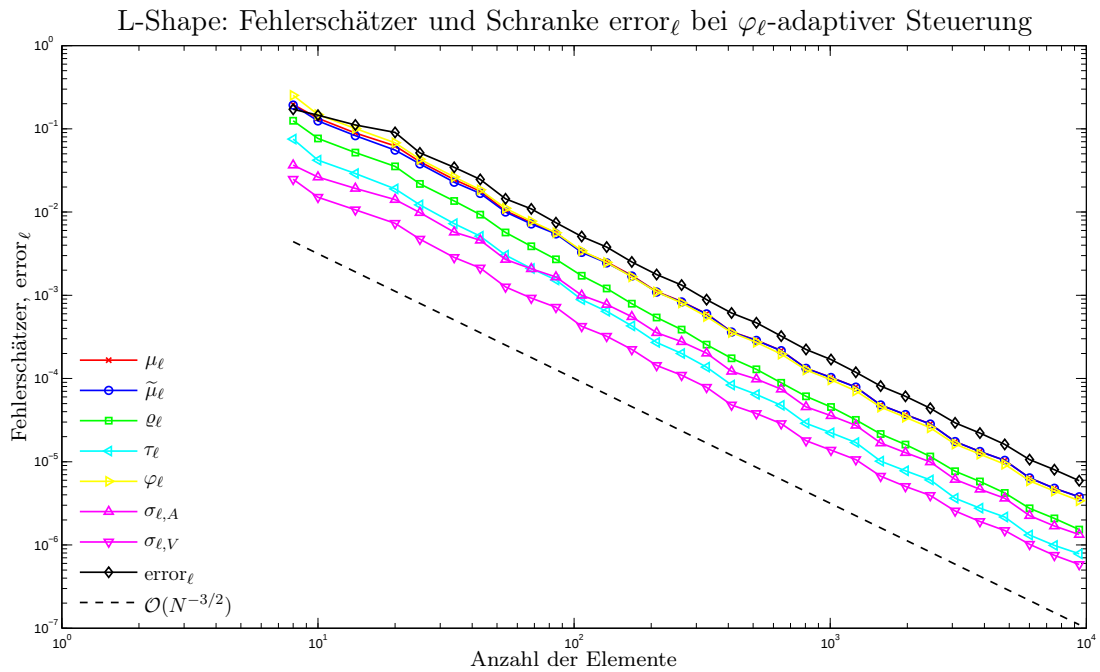


Abbildung 22: L-Shape: Fehlerschätzer und Schranke error_ℓ bei τ_ℓ -adaptiver Steuerung


 Abbildung 23: L-Shape: Fehlerschätzer und Schranke error_ℓ bei ϱ_ℓ -adaptiver Steuerung

 Abbildung 24: L-Shape: Fehlerschätzer und Schranke error_ℓ bei φ_ℓ -adaptiver Steuerung

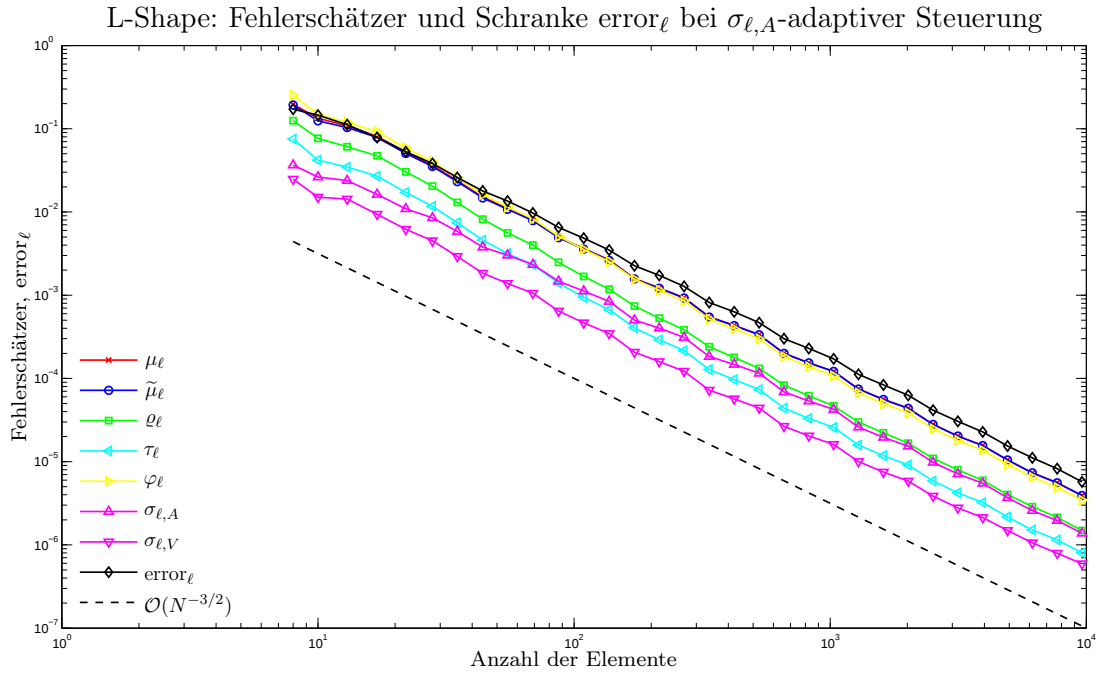


Abbildung 25: L-Shape: Fehlerschätzer und Schranke error_ℓ bei $\sigma_{\ell,A}$ -adaptiver Steuerung

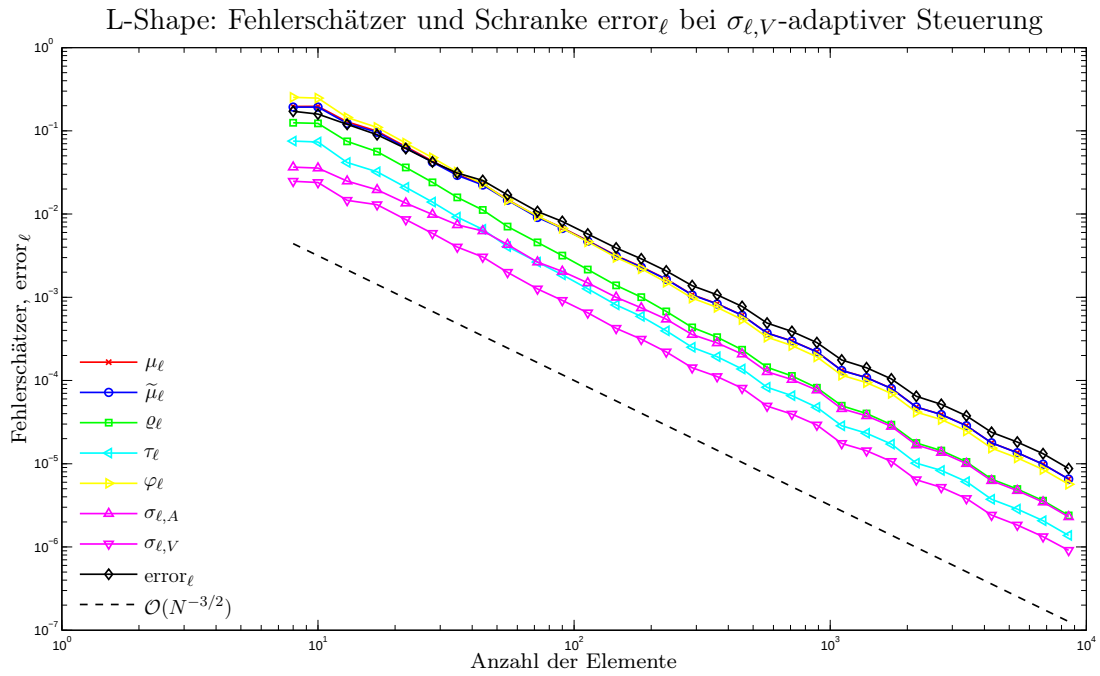


Abbildung 26: L-Shape: Fehlerschätzer und Schranke error_ℓ bei $\sigma_{\ell,V}$ -adaptiver Steuerung

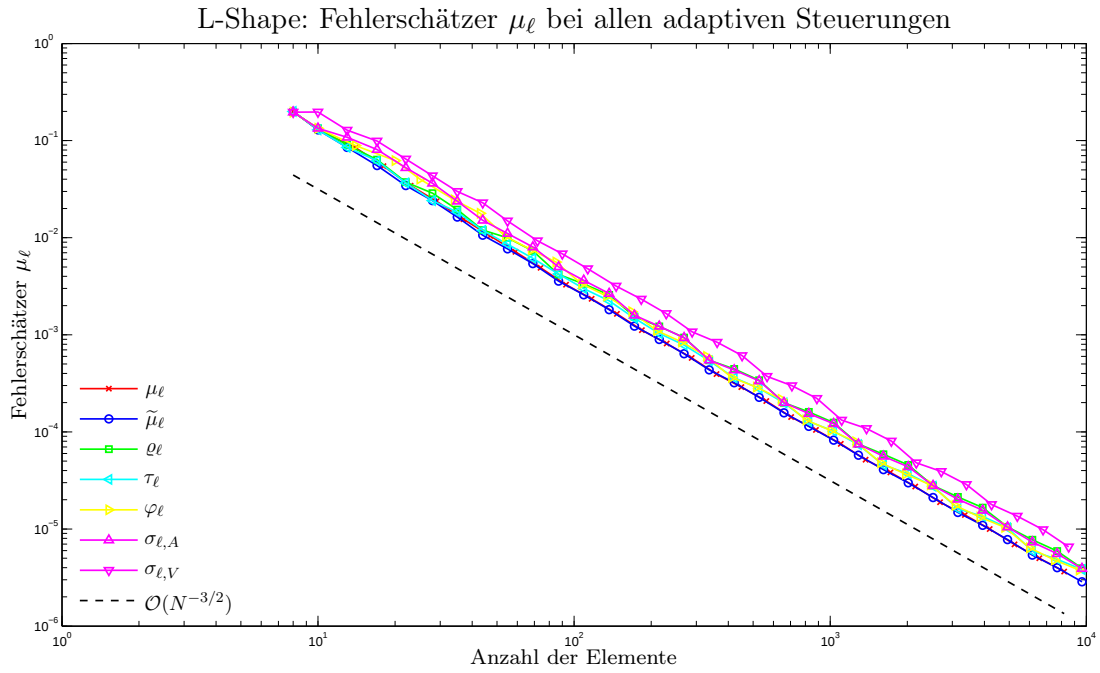


Abbildung 27: L-Shape: Fehlerschätzer μ_ℓ bei allen adaptiven Steuerungen

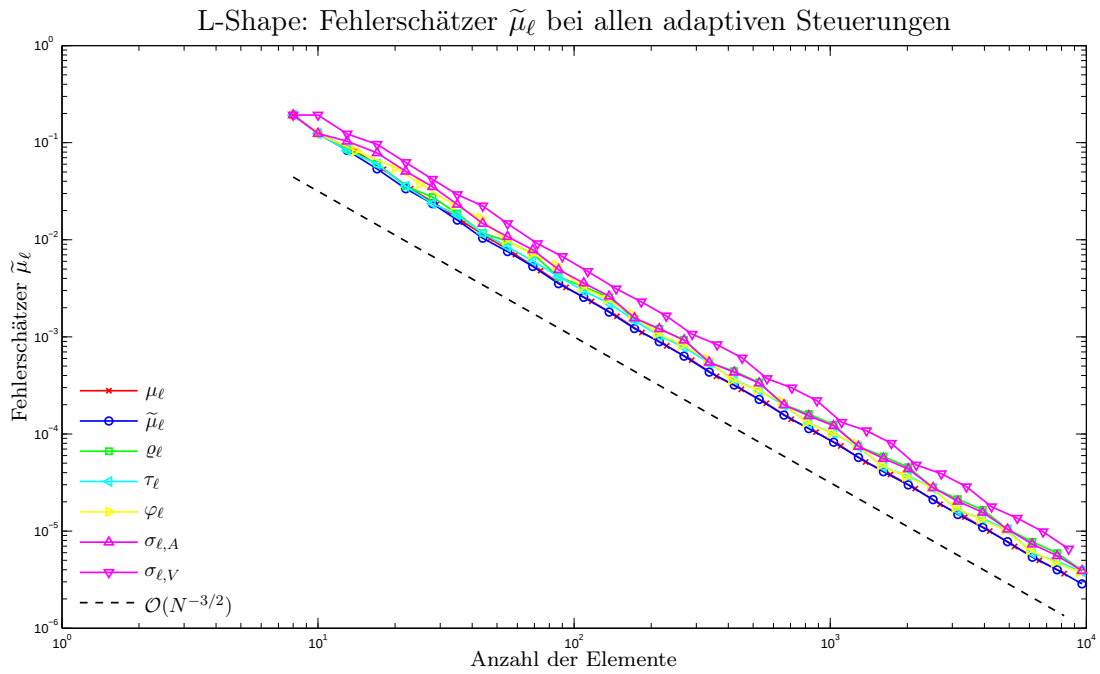


Abbildung 28: L-Shape: Fehlerschätzer $\tilde{\mu}_\ell$ bei allen adaptiven Steuerungen

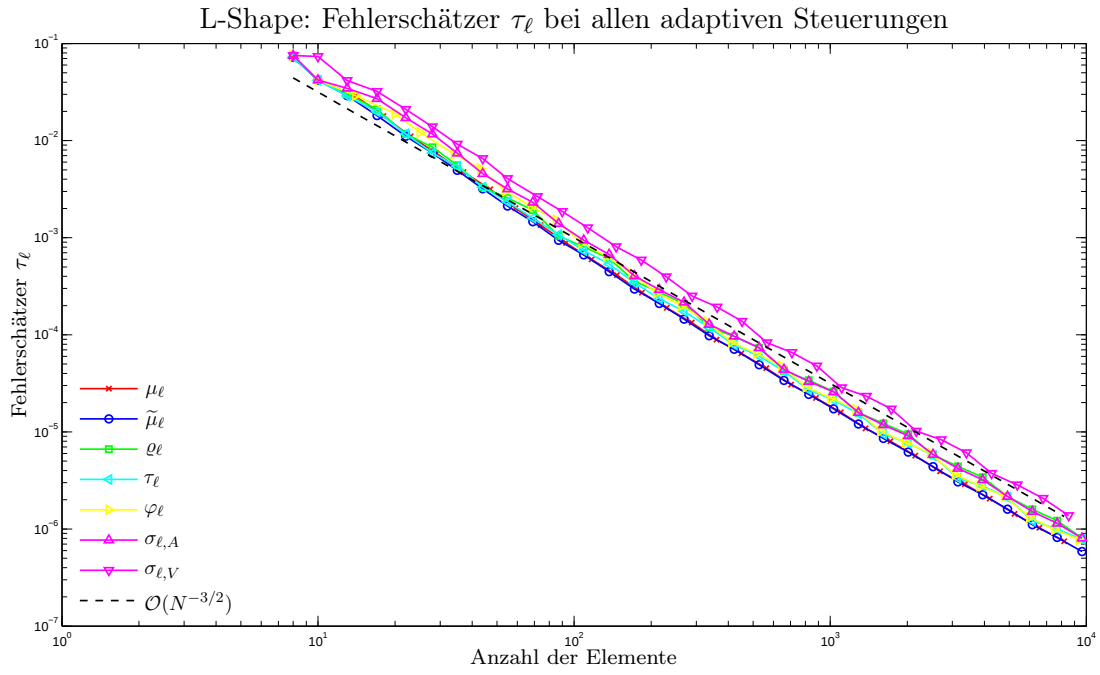


Abbildung 29: L-Shape: Fehlerschätzer τ_ℓ bei allen adaptiven Steuerungen

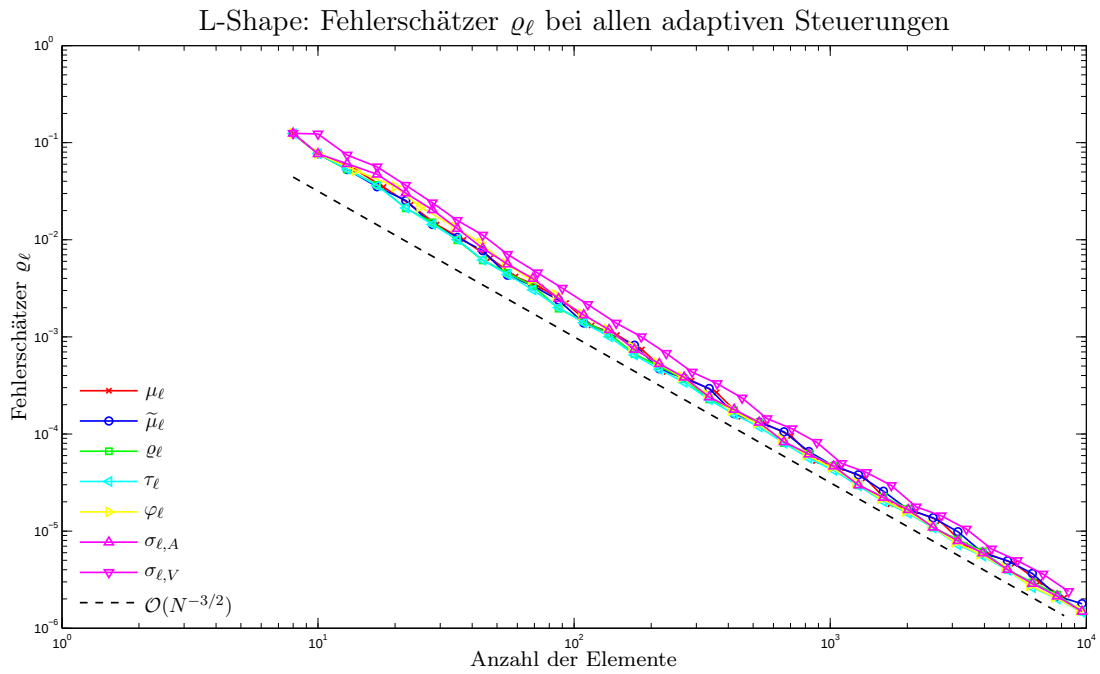


Abbildung 30: L-Shape: Fehlerschätzer ϱ_ℓ bei allen adaptiven Steuerungen

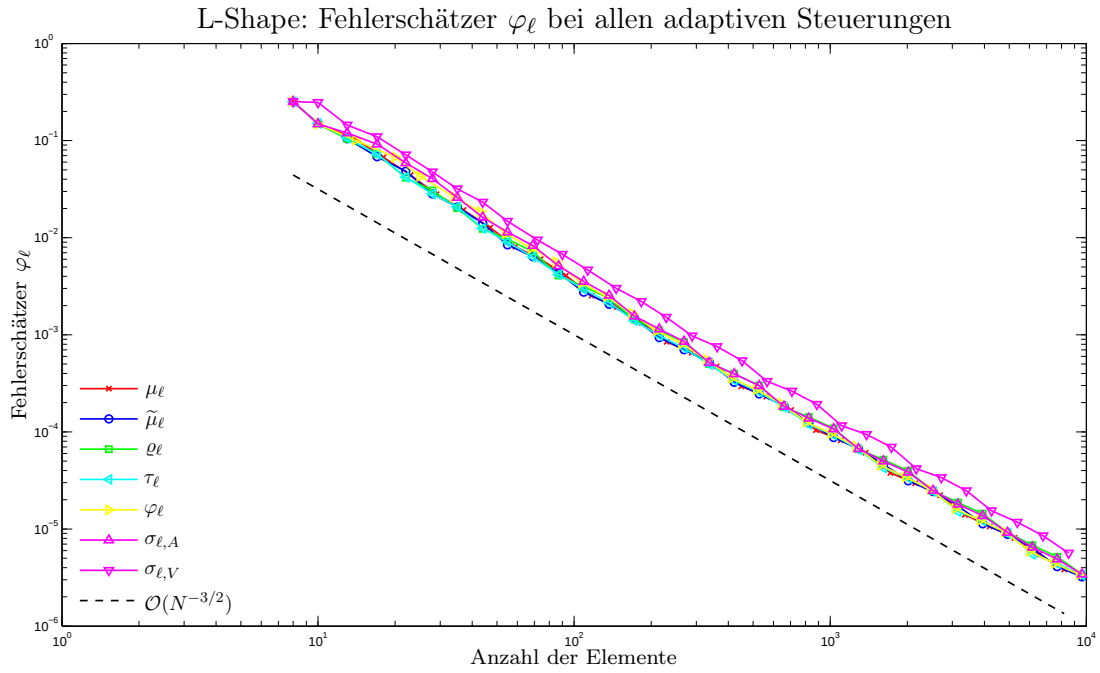


Abbildung 31: L-Shape: Fehlerschätzer φ_ℓ bei allen adaptiven Steuerungen

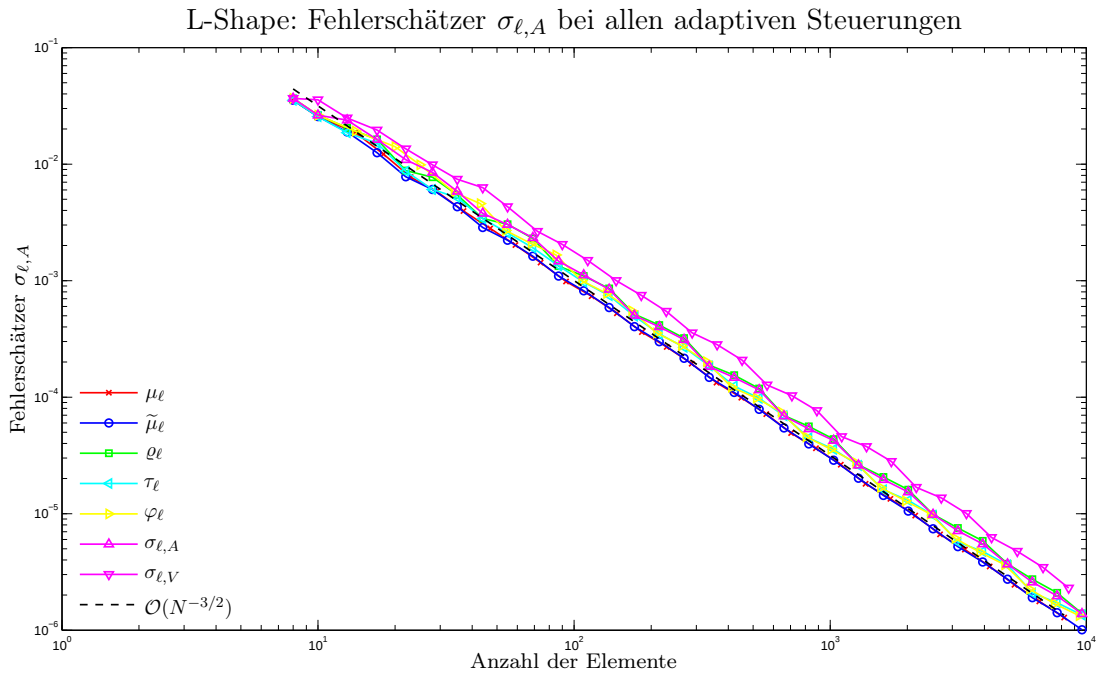


Abbildung 32: L-Shape: Fehlerschätzer $\sigma_{\ell,A}$ bei allen adaptiven Steuerungen

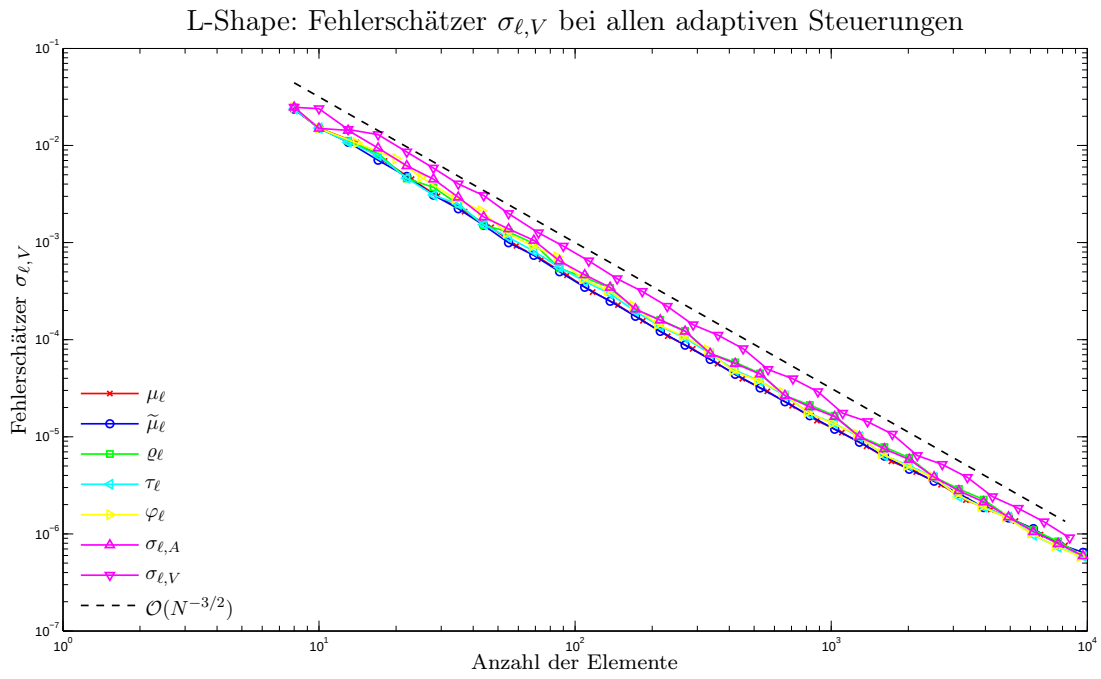


Abbildung 33: L-Shape: Fehlerschätzer $\sigma_{\ell,V}$ bei allen adaptiven Steuerungen

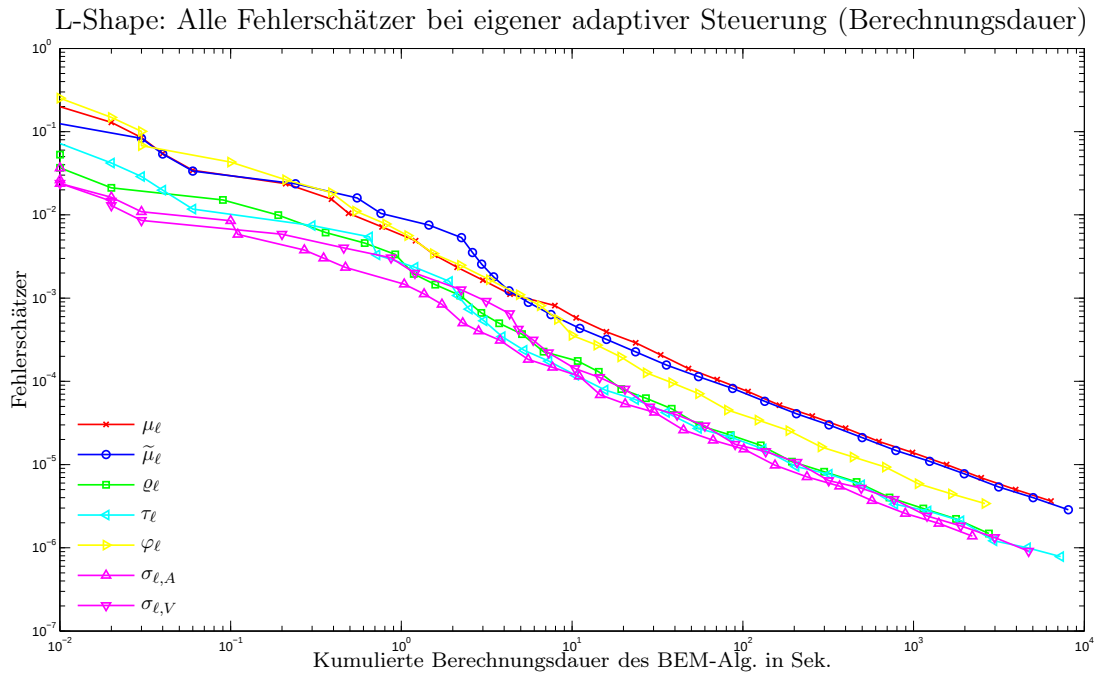
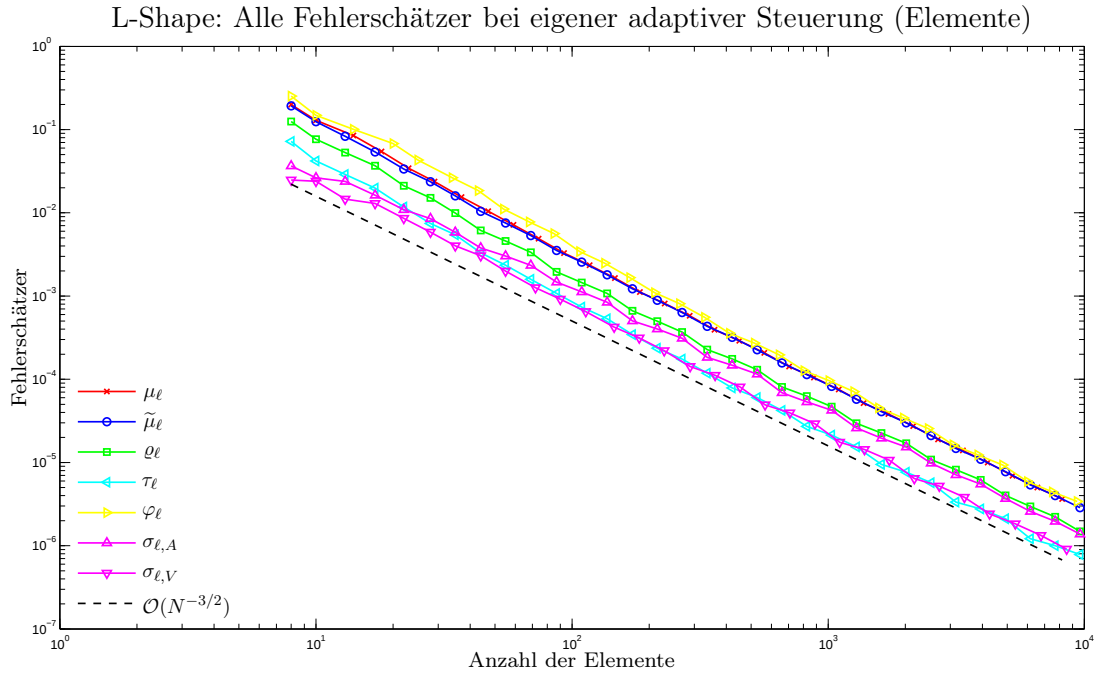


Abbildung 34: L-Shape: Alle Fehlerschätzer bei eigener adaptiver Steuerung

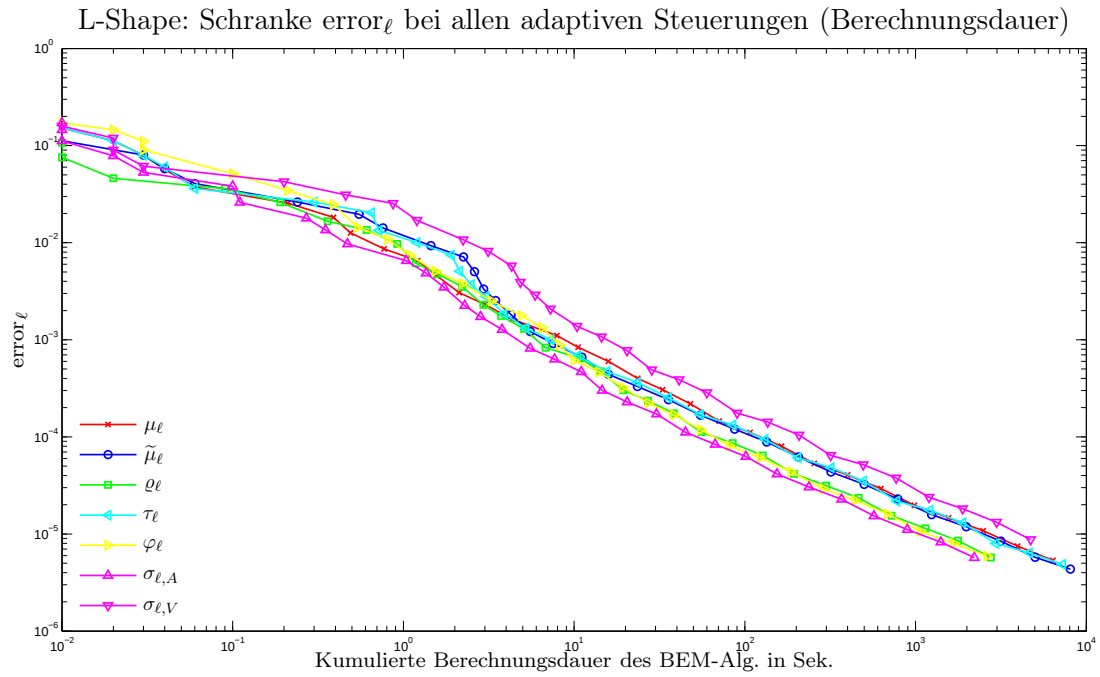
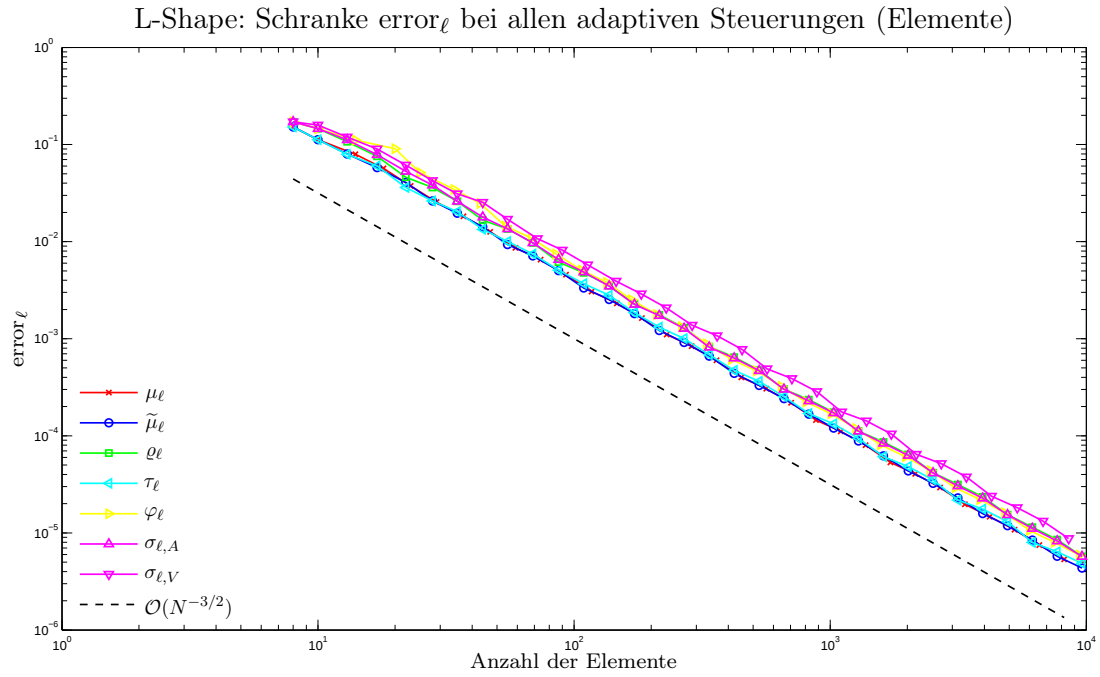
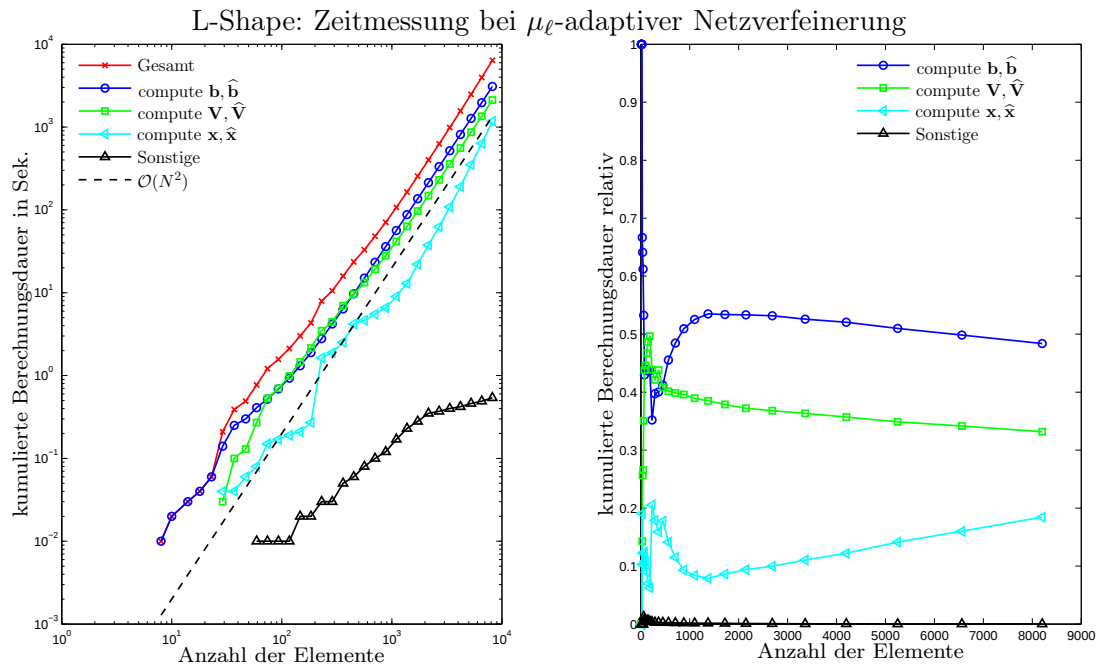
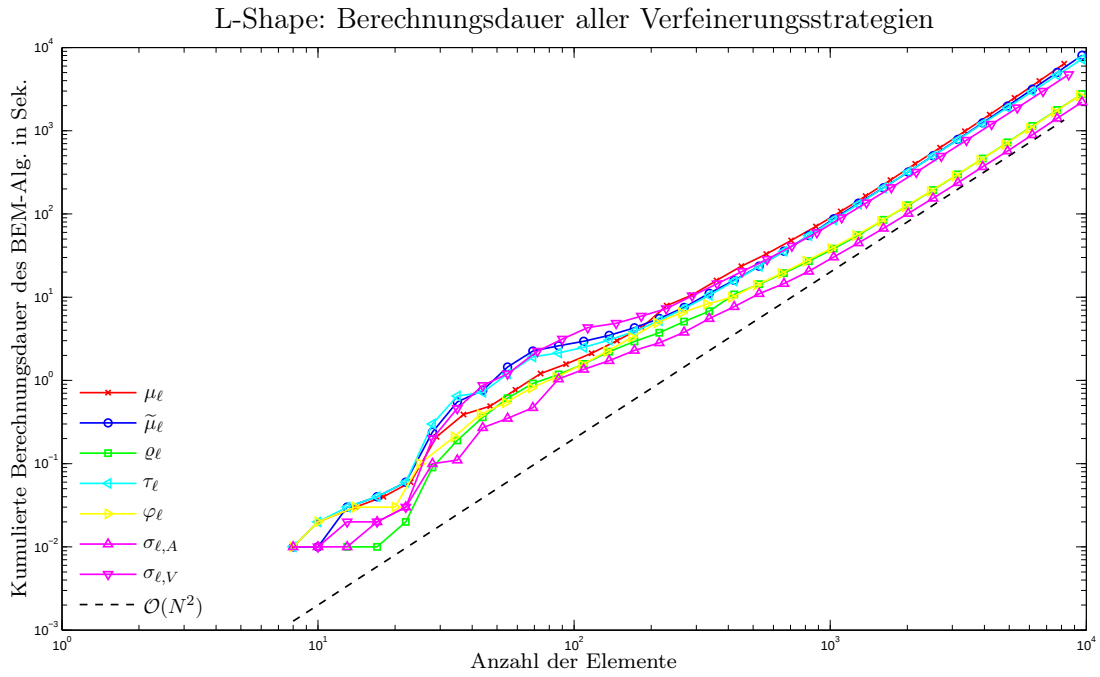


Abbildung 35: L-Shape: Schranke error_ℓ bei allen adaptiven Steuerungen über Berechnungsdauer und Anzahl der Elemente



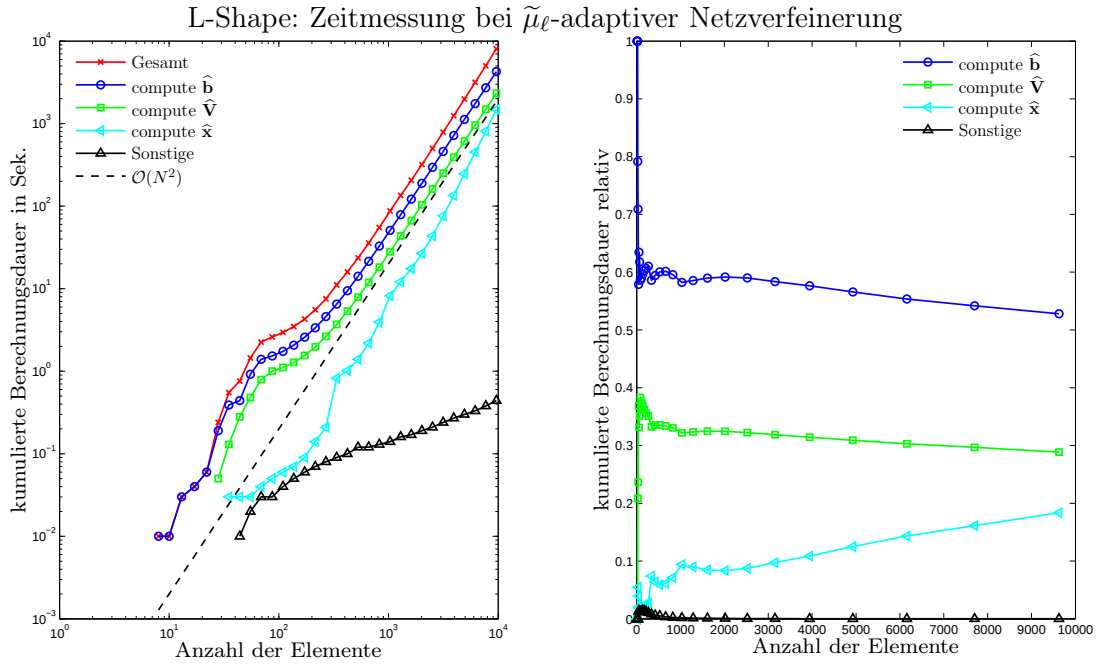


Abbildung 38: L-Shape: Zeitmessung bei $\tilde{\mu}_\ell$ -adaptiver Netzverfeinerung

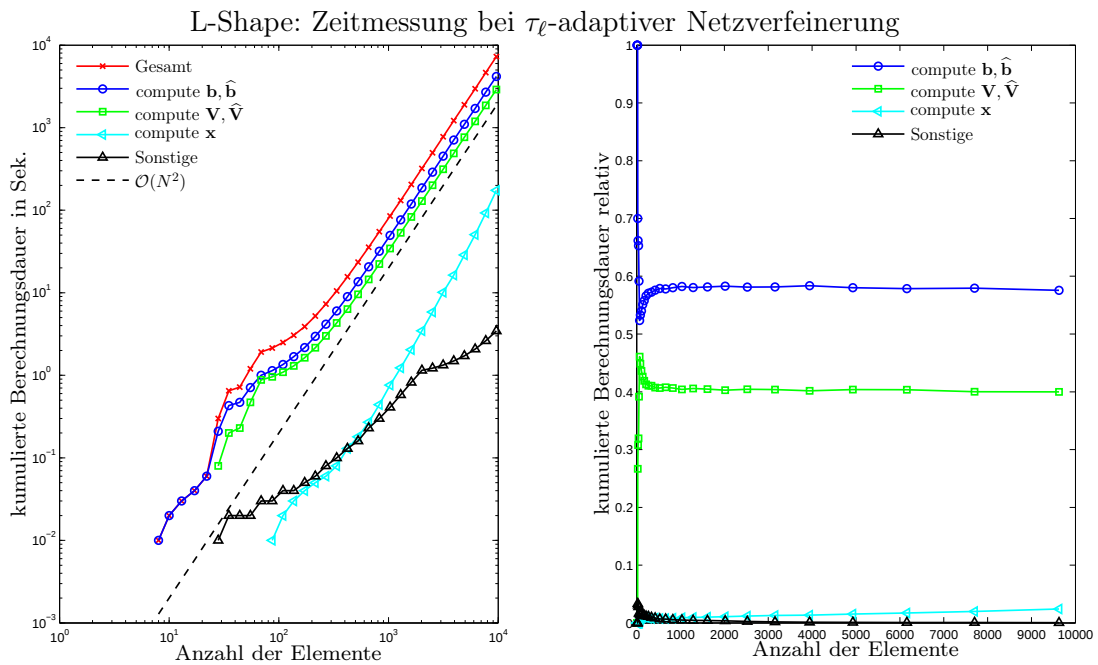
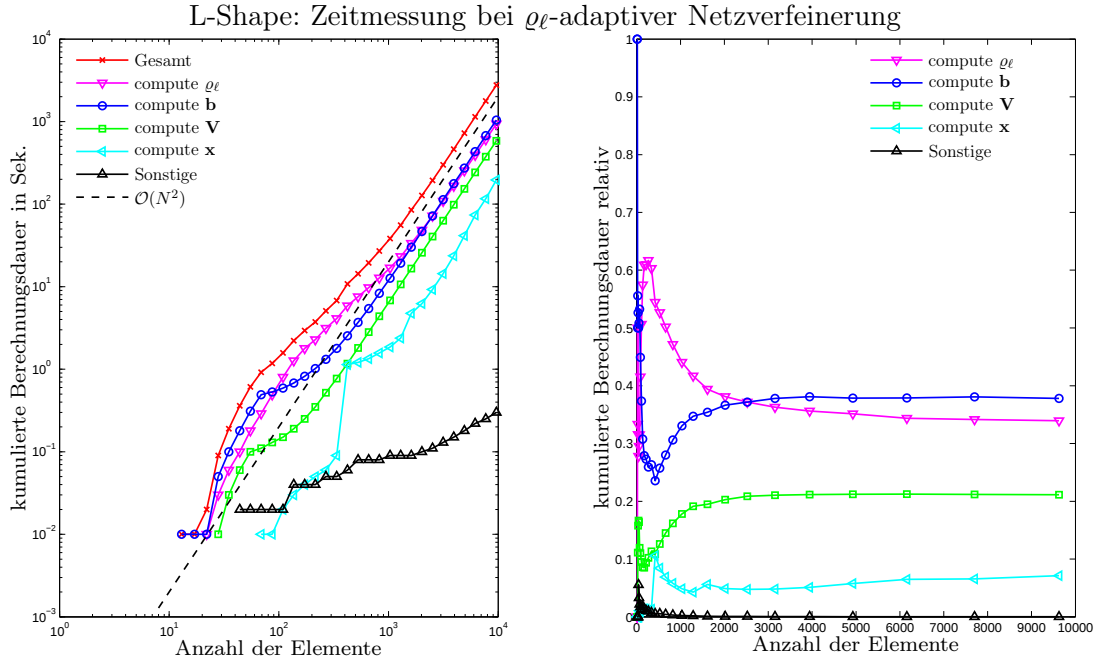
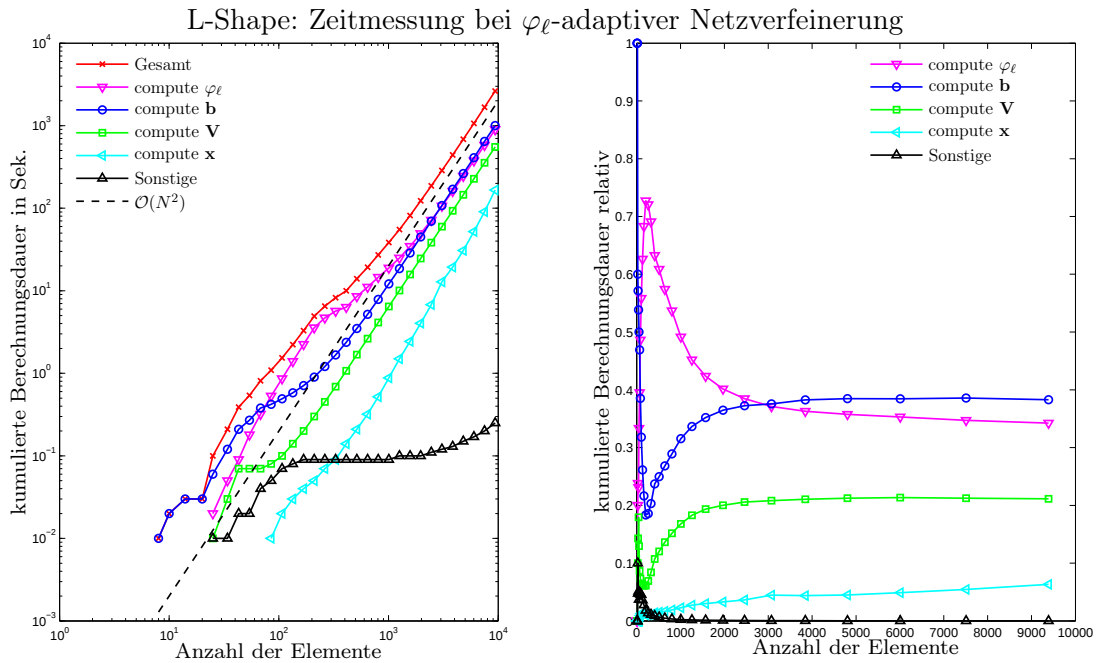


Abbildung 39: L-Shape: Zeitmessung bei τ_ℓ -adaptiver Netzverfeinerung


 Abbildung 40: L-Shape: Zeitmessung bei ϱ_ℓ -adaptiver Netzverfeinerung

 Abbildung 41: L-Shape: Zeitmessung bei φ_ℓ -adaptiver Netzverfeinerung

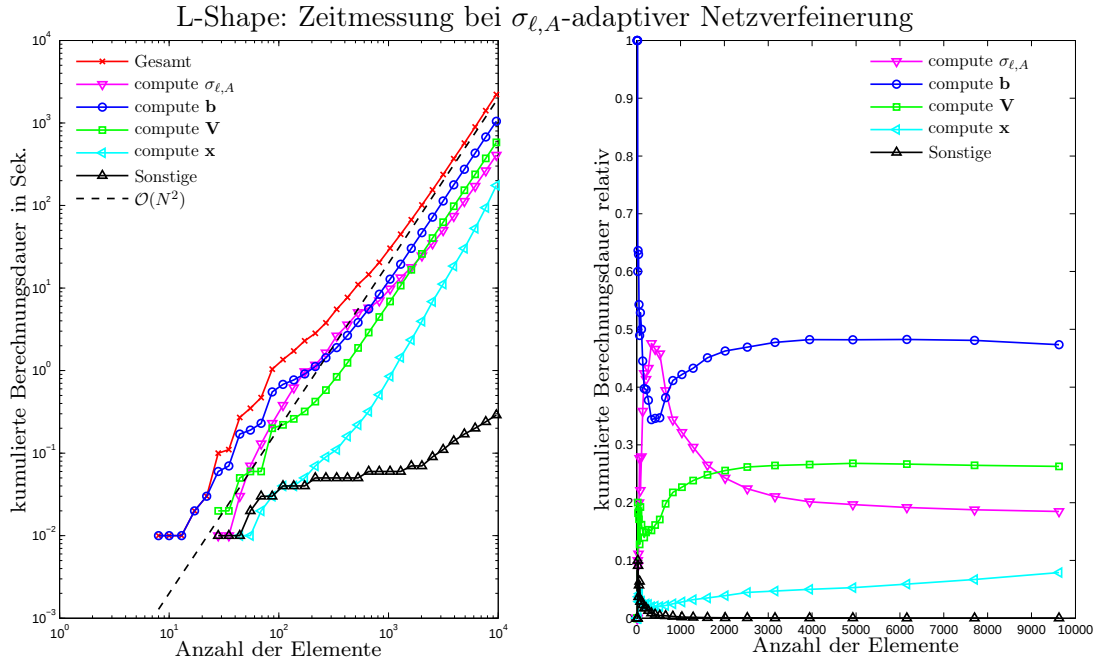


Abbildung 42: L-Shape: Zeitmessung bei $\sigma_{\ell,A}$ -adaptiver Netzverfeinerung

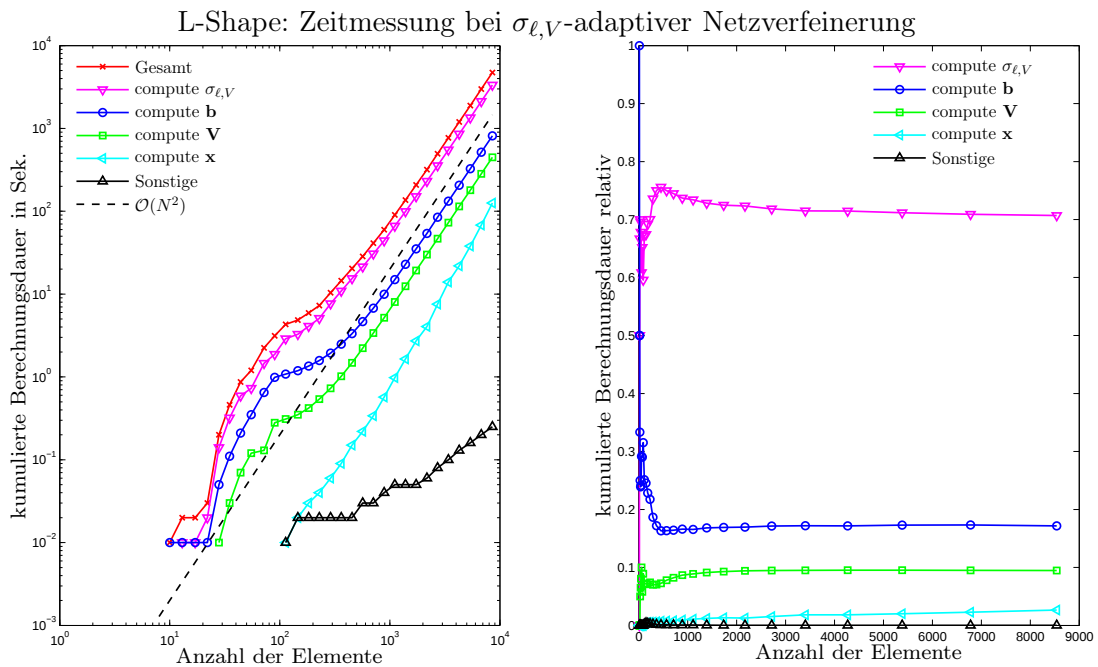


Abbildung 43: L-Shape: Zeitmessung bei $\sigma_{\ell,V}$ -adaptiver Netzverfeinerung

6.3.2 Z-Shape

In diesem Abschnitt führen wir nun die numerischen Simulationen auf dem Z-Shape, dargestellt in Abbildung 6 mit der Ausgangspartitionierung $\mathcal{E}_0$ des Randes Γ für den adaptiven Algorithmus, durch. Zuerst schätzen wir wieder die Regularität der Lösung ϕ und berechnen die Galerkin-Lösung Φ_ℓ mit uniformer Netzverfeinerung. In Abbildung 44 sehen wir die obere Schranke error_ℓ des Fehlers $\|\phi - \Phi_\ell\|_V$ und alle Fehlerschätzer über der Anzahl der Elemente aufgetragen. Mit Gleichung (6.7) erhalten wir eine Konvergenzverhalten von ca. $\|\phi - \Phi_\ell\|_V \approx \mathcal{O}(h^{4/7})$, was mit (6.5) zu einer Schätzung von $\phi \in H^s(\Gamma)$ mit $s \leq 1/14$ führt.

In den Abbildungen 45 bis 51 fällt sofort das schlechte Abschneiden beider Steinbach-Schätzer $\sigma_{\ell,V}, \sigma_{\ell,A}$ auf, was nach den Ausführungen in Abschnitt 6.2.3 zu erwarten war. Überraschend ist allerdings, dass obwohl der Steinbach-Schätzer $\sigma_{\ell,A}$ unglaubliche Ergebnisse liefert, dessen adaptiv-gesteuerter Algorithmus aber hinsichtlich Fehlerschranke und übriger Fehlerschätzer gute Ergebnisse erzielt, vgl. Abbildung 50 – ganz im Gegensatz zum Schätzer $\sigma_{\ell,V}$, vgl. Abbildung 51. Nehmen wir die Steinbach-Schätzer nun aus, so konvergiert die Fehlerschranke error_ℓ bei allen anderen Schätzern mit $\mathcal{O}(N^{-3/2})$ und folglich auch der Fehler $\|\phi - \Phi_\ell\|_V$. Die Fehlerschätzer sind dazu parallel, was die Effizienz und Zuverlässigkeit sämtlicher Fehlerschätzer (bis auf Steinbach-Schätzer) auch auf dem Z-Shape experimentell nachweist. Wieder wird hier ausdrücklich auf die nicht vorhandenen Voraussetzungen der Randdaten und der Netze hingewiesen, die für die Beweise obiger Beobachtungen in dieser Arbeit notwendig sind, siehe die Ausführungen beim L-Shape. Die Abbildungen 52 bis 58 zeigen, dass bis auf die Steuerungen mit den Steinbach-Schätzern, alle Ergebnisse beinahe deckungsgleich sind, woraus wir wieder auf die Stabilität der Implementierungen schließen. Aus den Abbildungen 57 und 58 entnehmen wir, dass die Ergebnisse der Steinbach-Schätzer aber einer Anzahl von etwa 400 Elementen zu unbrauchbaren Ergebnissen führen, wobei der Steinbach-Schätzer $\sigma_{\ell,A}$ etwas länger durchhält. Bezugnehmend auf die Resultate in Abschnitt 6.2.3 erwarten wir auch mit keiner anderen Parameterwahl eine signifikante Verbesserung.

In Abbildung 59 sind alle Fehlerschätzer bei eigener adaptiver Steuerung über der Elementanzahl und der Berechnungsdauer aufgetragen. Sehr schön sieht man die Konvergenzrate von $\mathcal{O}(N^{-3/2})$ aller Fehlerschätzer (wieder bis auf Steinbach-Schätzer).

Betrachten wir weiters die obere Schranke error_ℓ des Fehlers $\|\phi - \Phi_\ell\|_V$ in Abbildung 60: Ausgenommen von dem Steinbach-Schätzer $\sigma_{\ell,V}$ liefern alle Fehlerschätzer einen in etwa gleichen Kurvenverlauf über der Anzahl der Elemente. Die Genauigkeit $\|\phi - \Phi_\ell\|_V$ der Galerkin-Lösung kann bei einem $\sigma_{\ell,V}$ -adaptiv gesteuerten Algorithmus nicht über einen gewissen Level hinaus garantiert werden. Nach der bereits auf dem L-Shape durchgeführten Rechenzeit-Analyse liefern die Residualschätzer $\varrho_\ell, \varphi_\ell$ und der Steinbach-Schätzer $\sigma_{\ell,A}$ wie erwartet die schnellsten Ergebnisse.

Damit lassen sich die meisten Beobachtungen der Simulationen auf dem L-Shape auch auf das Z-Shape, selbst bei weniger regulären Cauchy-Daten g, ϕ , übertragen. Die Steinbach-Schätzer $\sigma_{\ell,A}, \sigma_{\ell,V}$ liefern ab einem hinreichend feinen Netz bei kritischeren Voraussetzungen an das Gebiet und den Cauchy-Daten zwar noch ganz gute Ergebnisse hinsichtlich des Fehlers $\|\phi - \Phi_\ell\|_V$ – zumindest im Fall des Steinbach-Schätzers $\sigma_{\ell,A}$ – die Resultate insgesamt sind aber nicht zufriedenstellend.

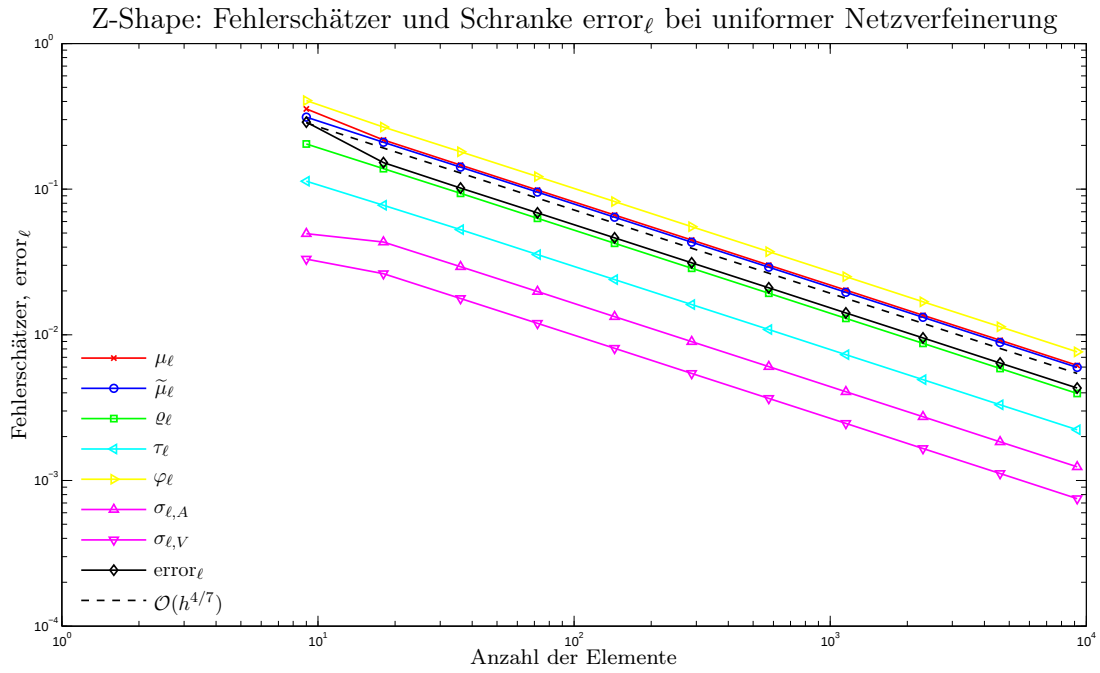


Abbildung 44: Z-Shape: Fehlerschätzer und Schranke error_ℓ bei uniformer Netzverfeinerung

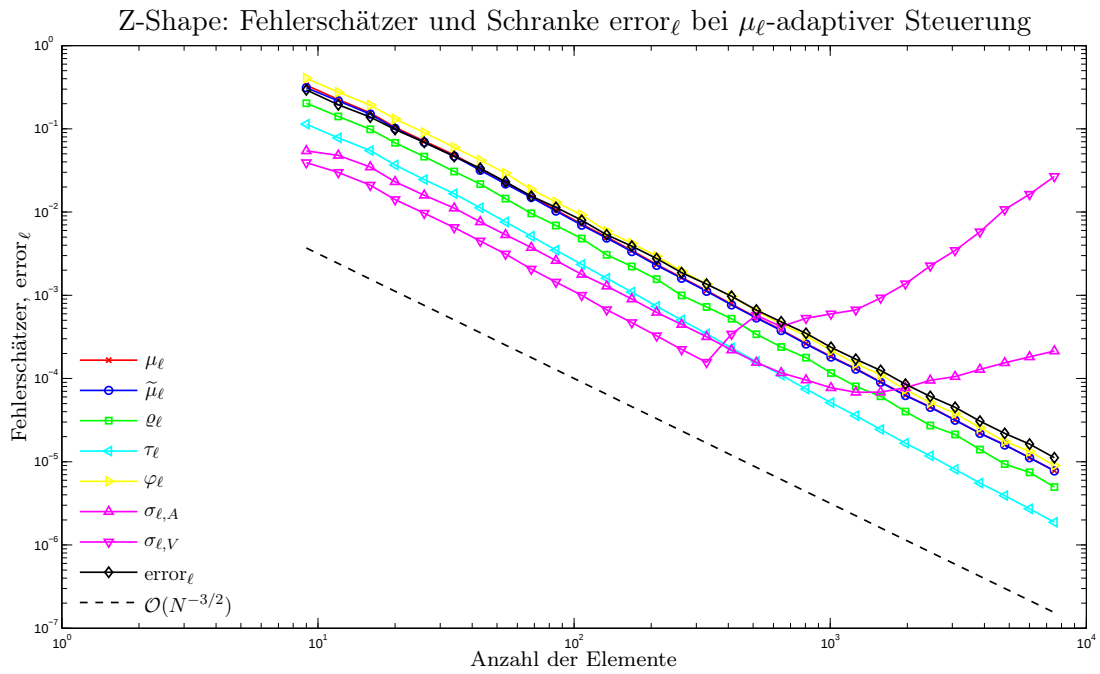
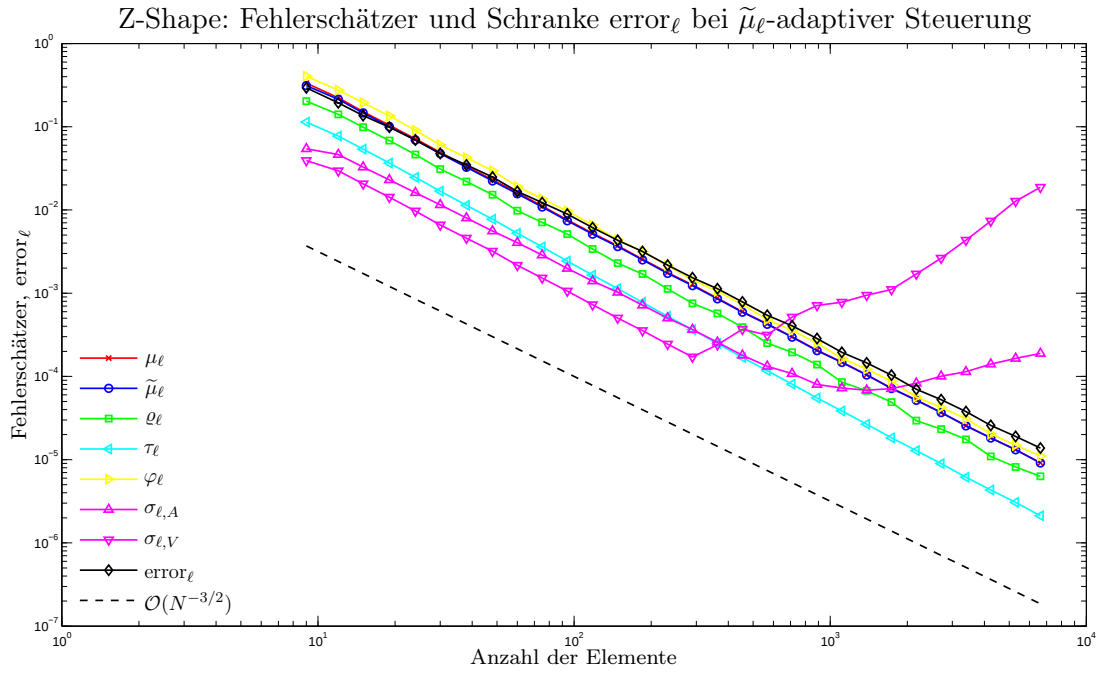
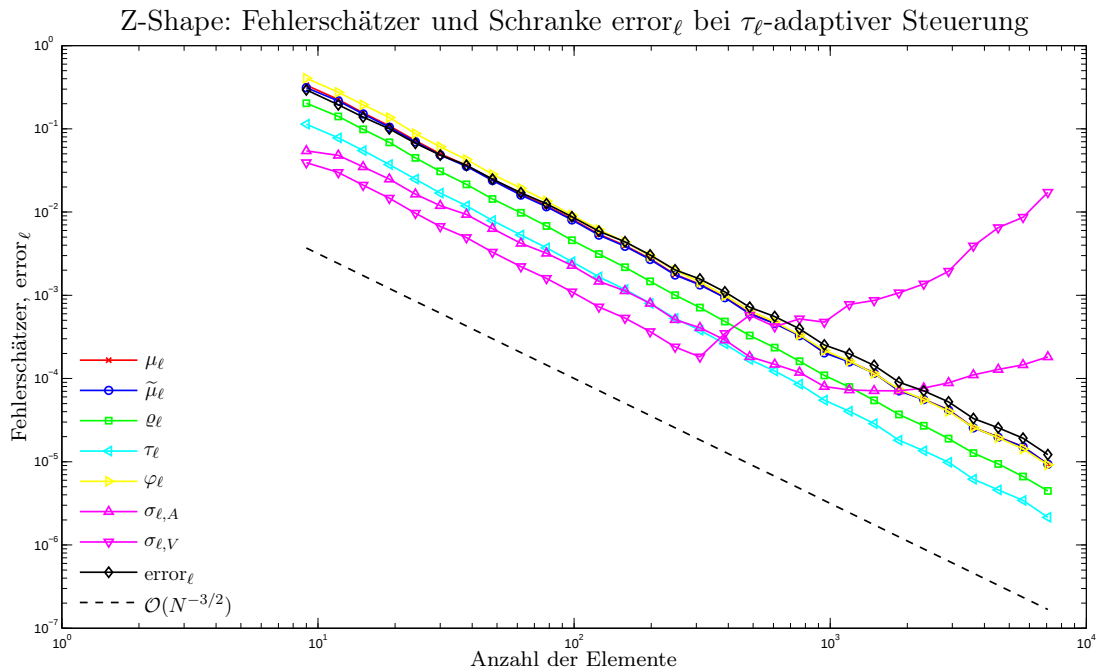


Abbildung 45: Z-Shape: Fehlerschätzer und Schranke error_ℓ bei μ_ℓ -adaptiver Steuerung


 Abbildung 46: Z-Shape: Fehlerschätzer und Schranke error_ℓ bei $\tilde{\mu}_\ell$ -adaptiver Steuerung

 Abbildung 47: Z-Shape: Fehlerschätzer und Schranke error_ℓ bei τ_ℓ -adaptiver Steuerung

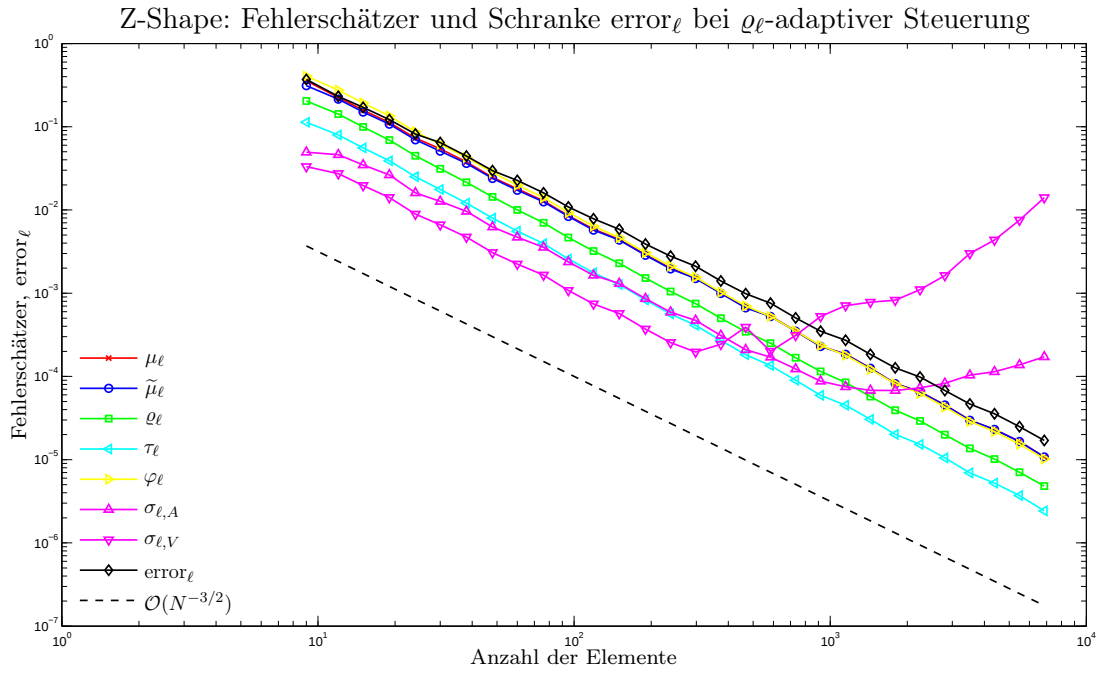


Abbildung 48: Z-Shape: Fehlerschätzer und Schranke error_ℓ bei ϱ_ℓ -adaptiver Steuerung

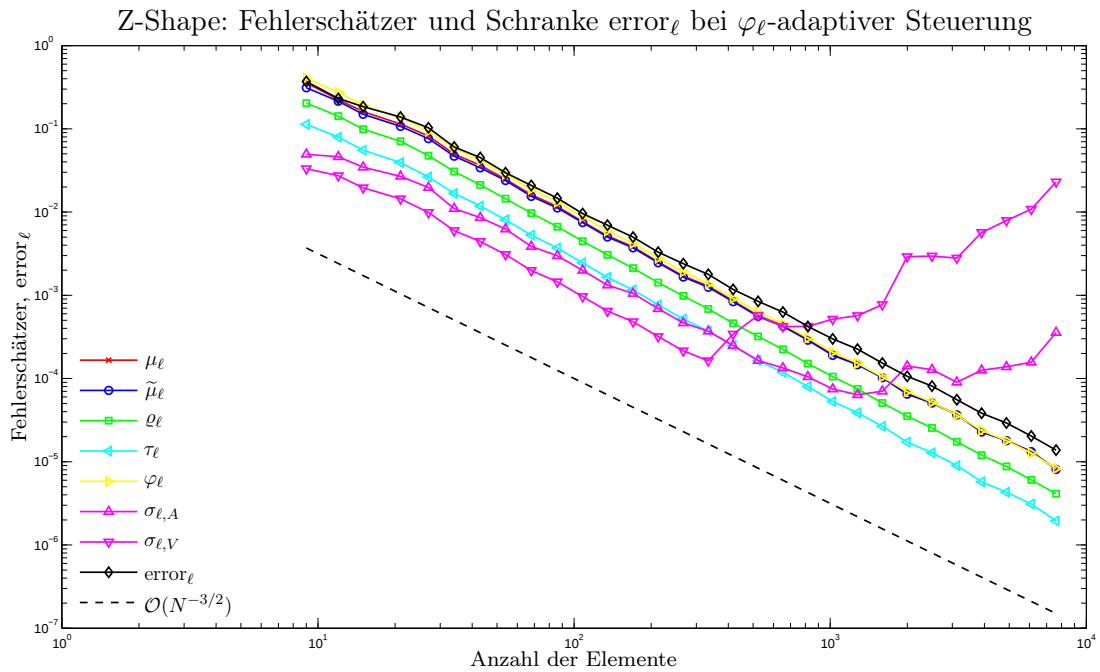
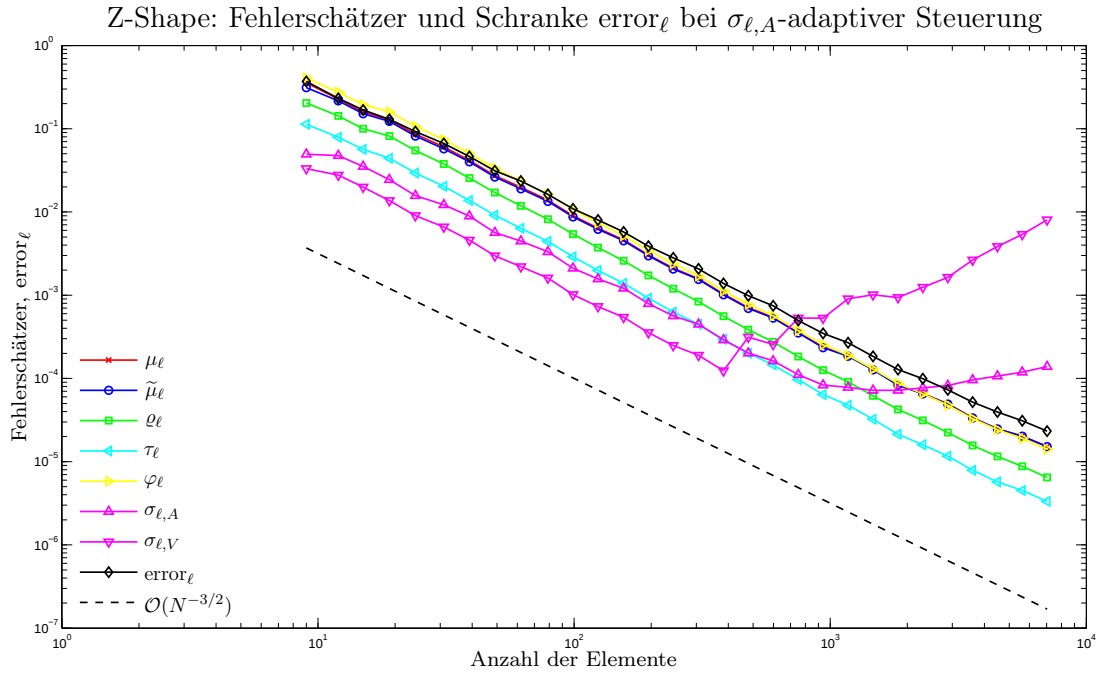
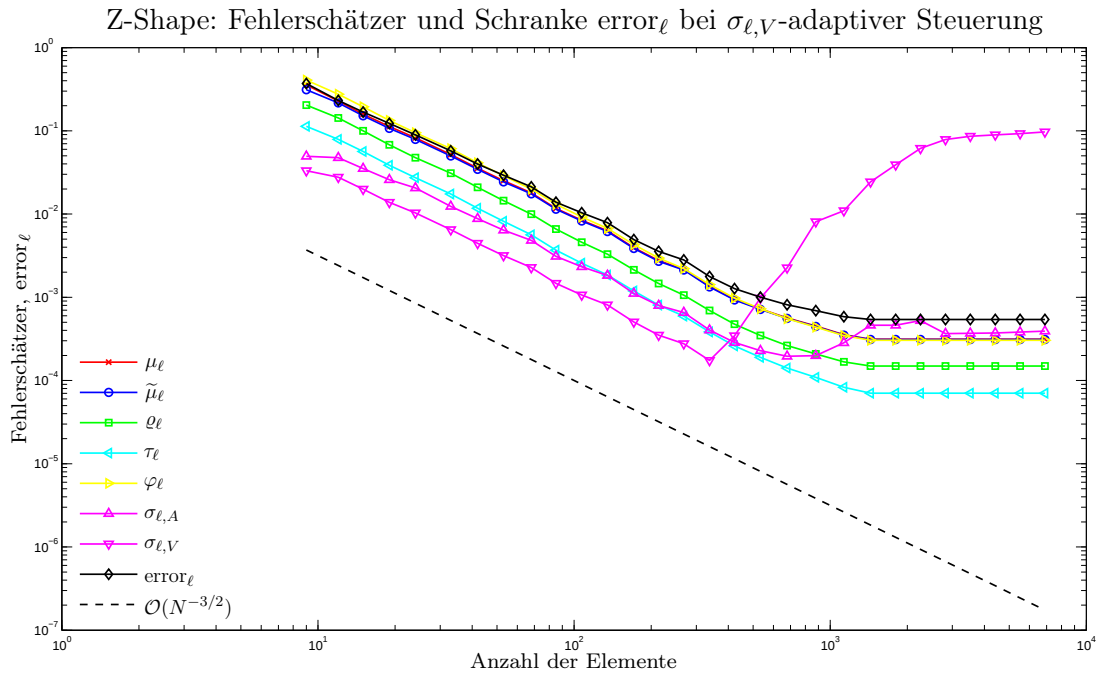


Abbildung 49: Z-Shape: Fehlerschätzer und Schranke error_ℓ bei φ_ℓ -adaptiver Steuerung


 Abbildung 50: Z-Shape: Fehlerschätzer und Schranke error_ℓ bei $\sigma_{\ell,A}$ -adaptiver Steuerung

 Abbildung 51: Z-Shape: Fehlerschätzer und Schranke error_ℓ bei $\sigma_{\ell,V}$ -adaptiver Steuerung

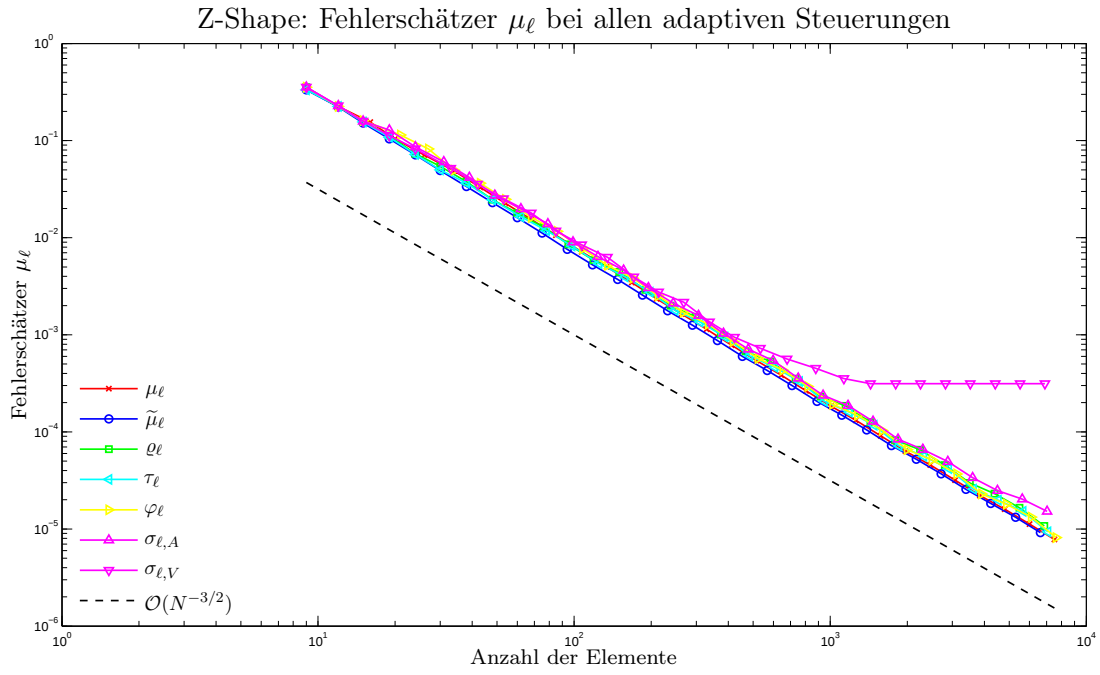


Abbildung 52: Z-Shape: Fehlerschätzer μ_ℓ bei allen adaptiven Steuerungen

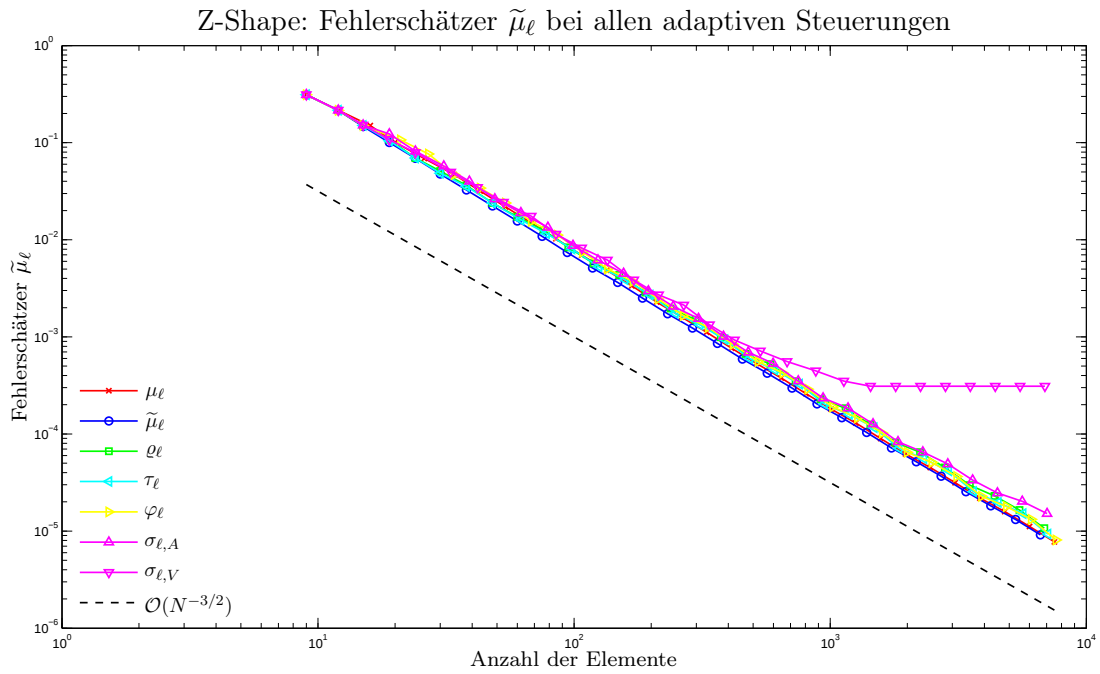


Abbildung 53: Z-Shape: Fehlerschätzer $\tilde{\mu}_\ell$ bei allen adaptiven Steuerungen

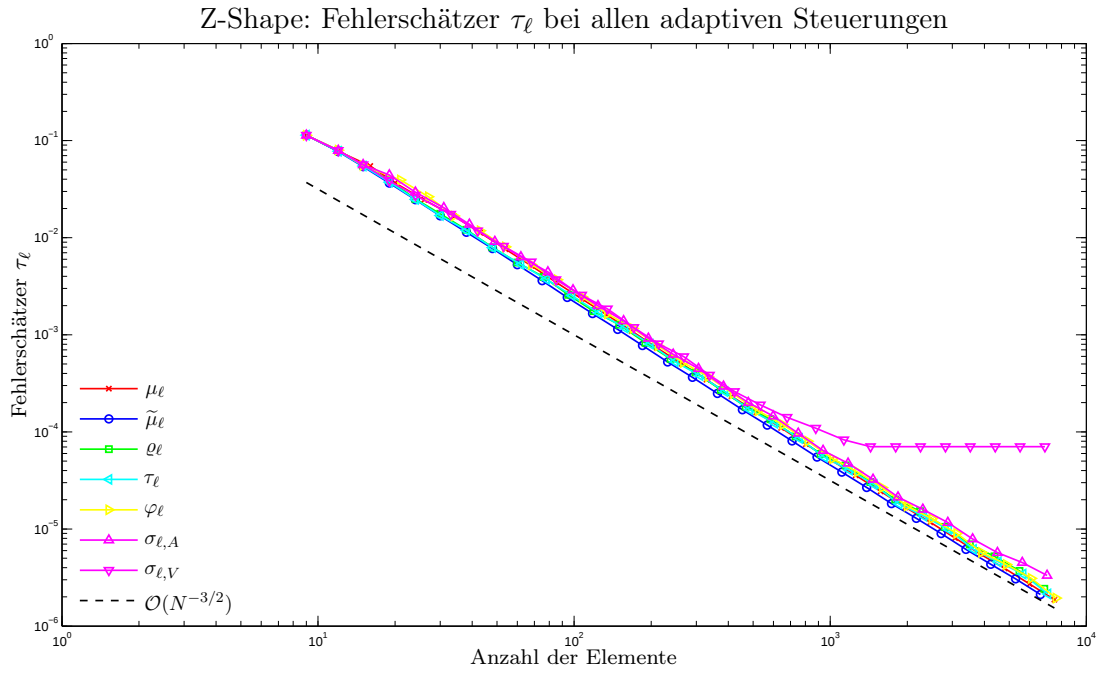


Abbildung 54: Z-Shape: Fehlerschätzer τ_ℓ bei allen adaptiven Steuerungen

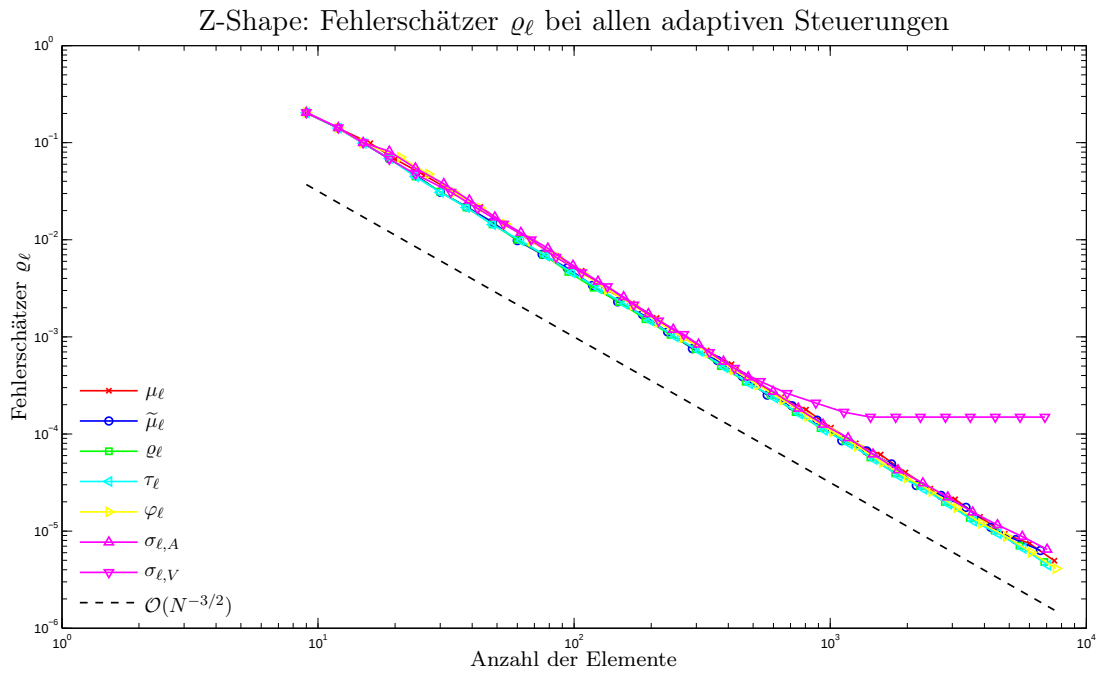


Abbildung 55: Z-Shape: Fehlerschätzer ϱ_ℓ bei allen adaptiven Steuerungen

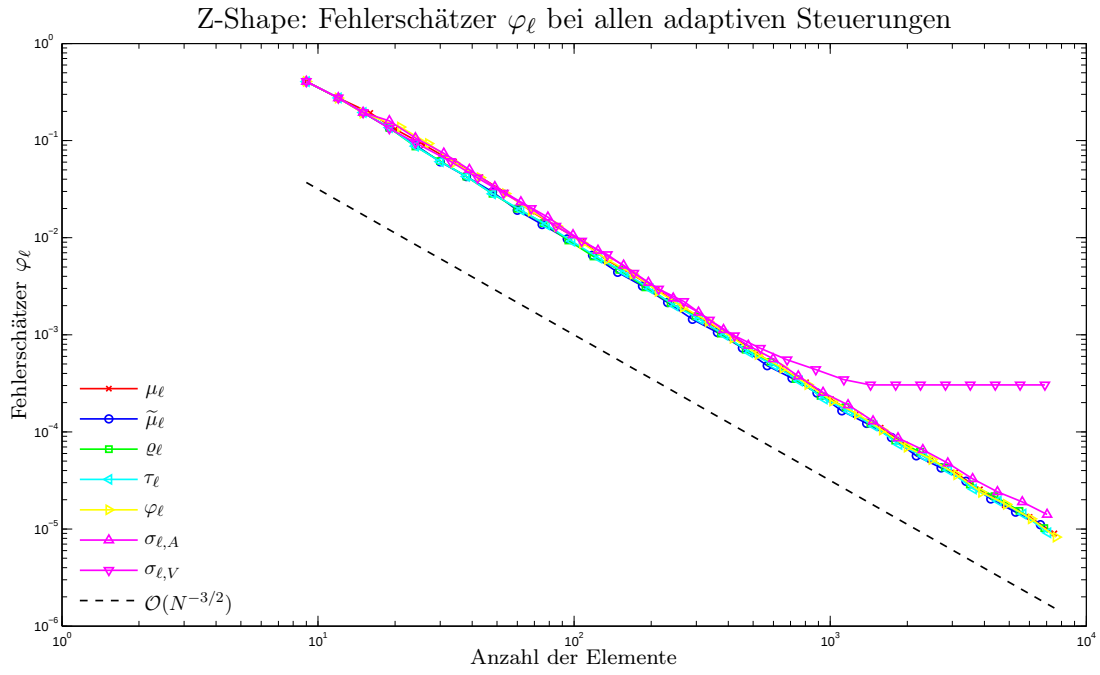


Abbildung 56: Z-Shape: Fehlerschätzer φ_ℓ bei allen adaptiven Steuerungen

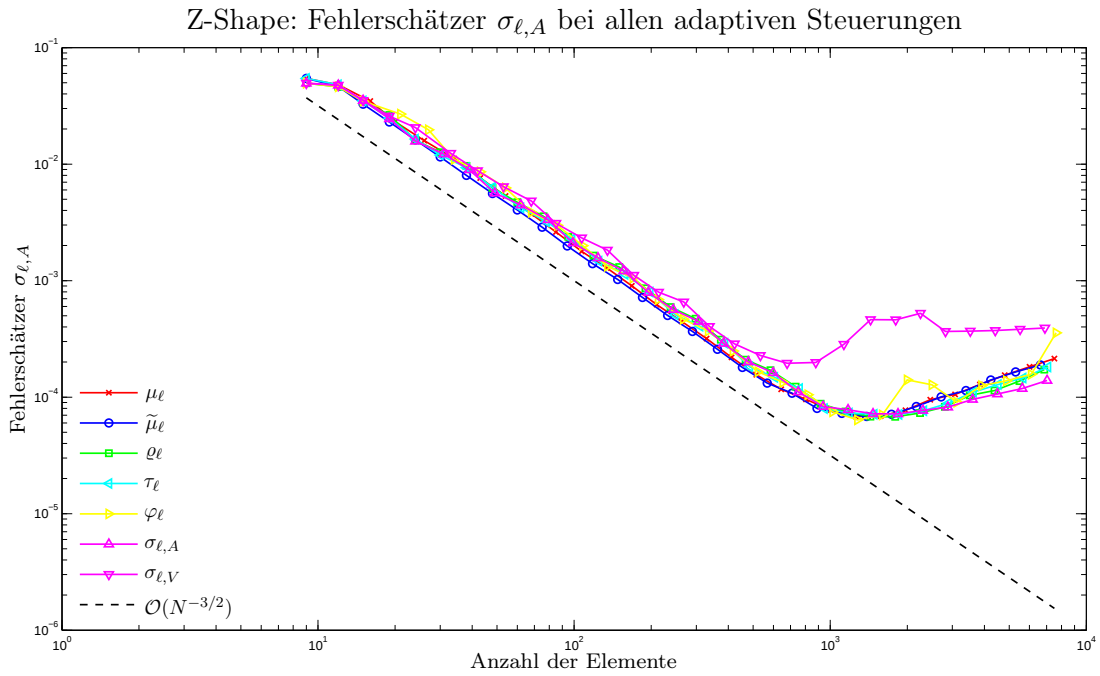


Abbildung 57: Z-Shape: Fehlerschätzer $\sigma_{\ell,A}$ bei allen adaptiven Steuerungen

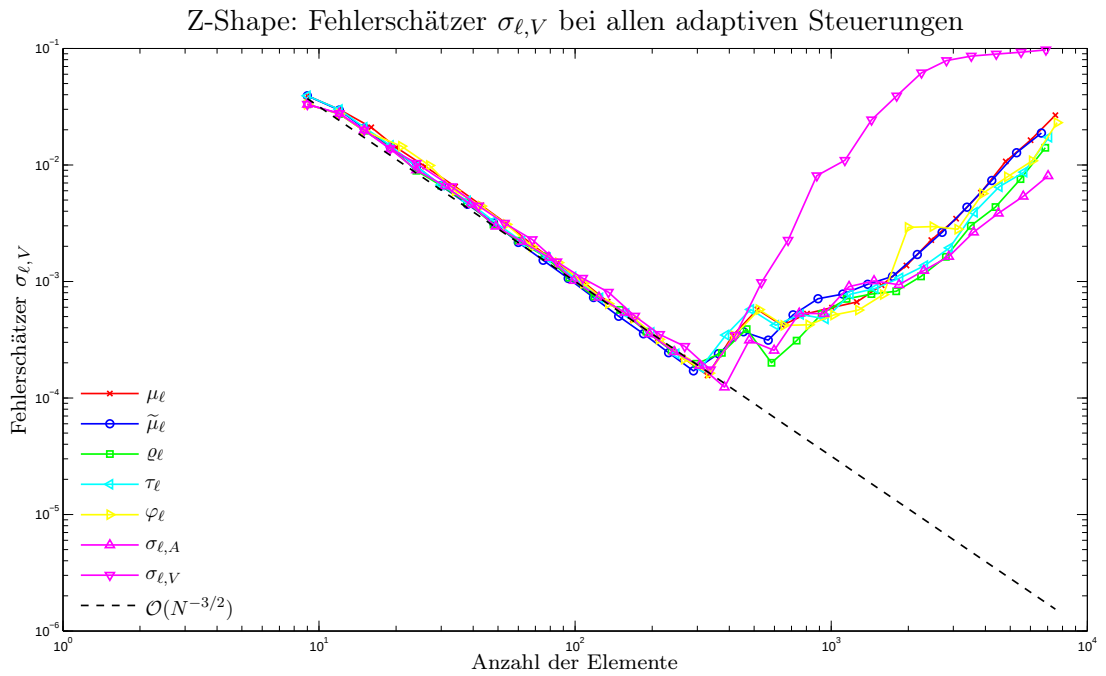


Abbildung 58: Z-Shape: Fehlerschätzer $\sigma_{\ell,V}$ bei allen adaptiven Steuerungen

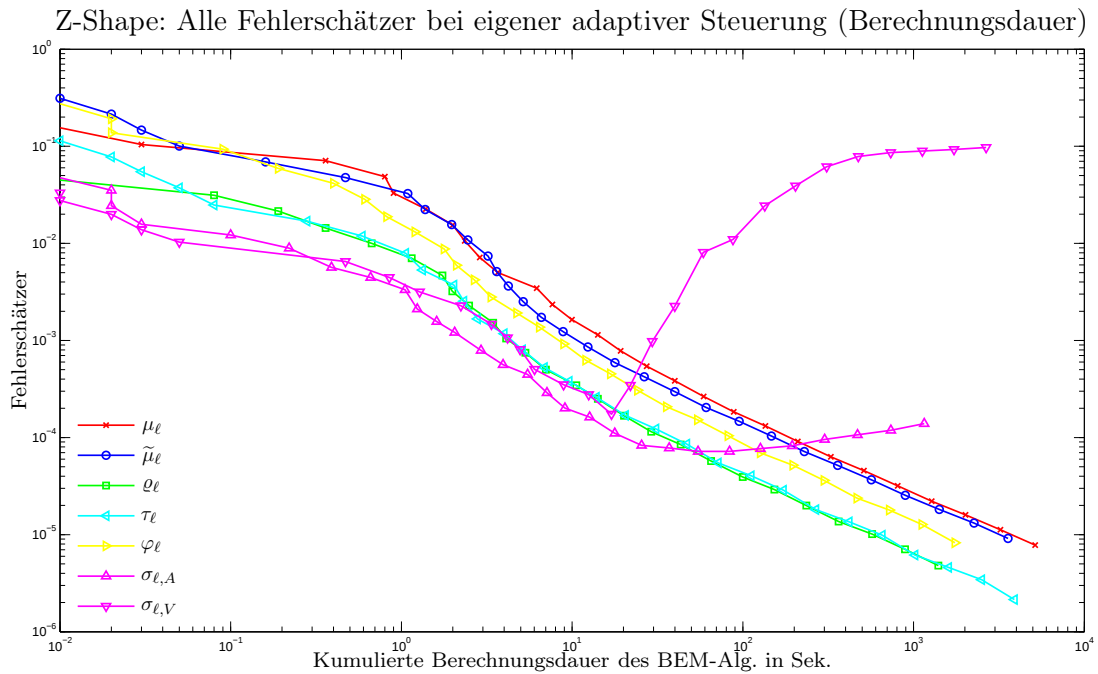
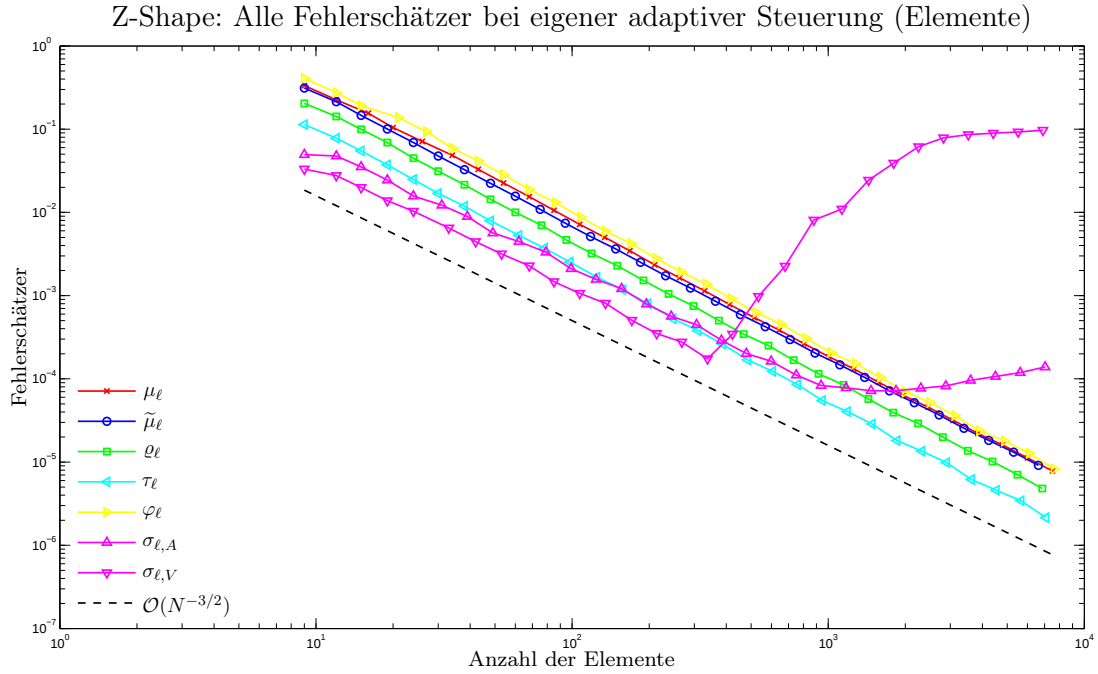


Abbildung 59: Z-Shape: Alle Fehlerschätzer bei eigener adaptiver Steuerung

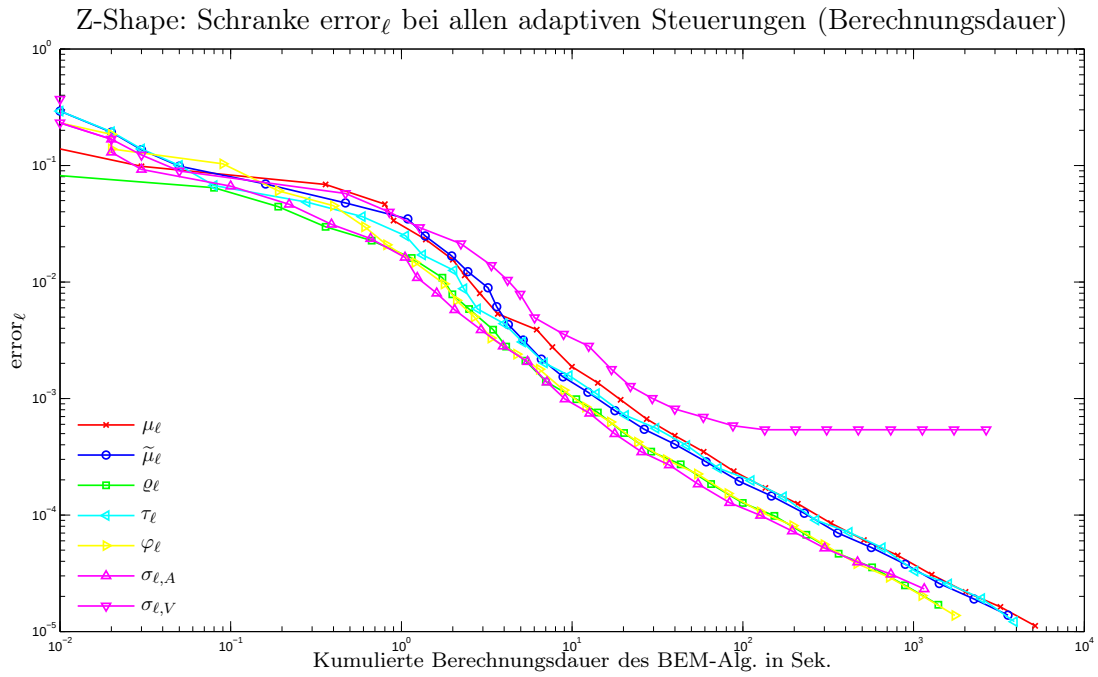
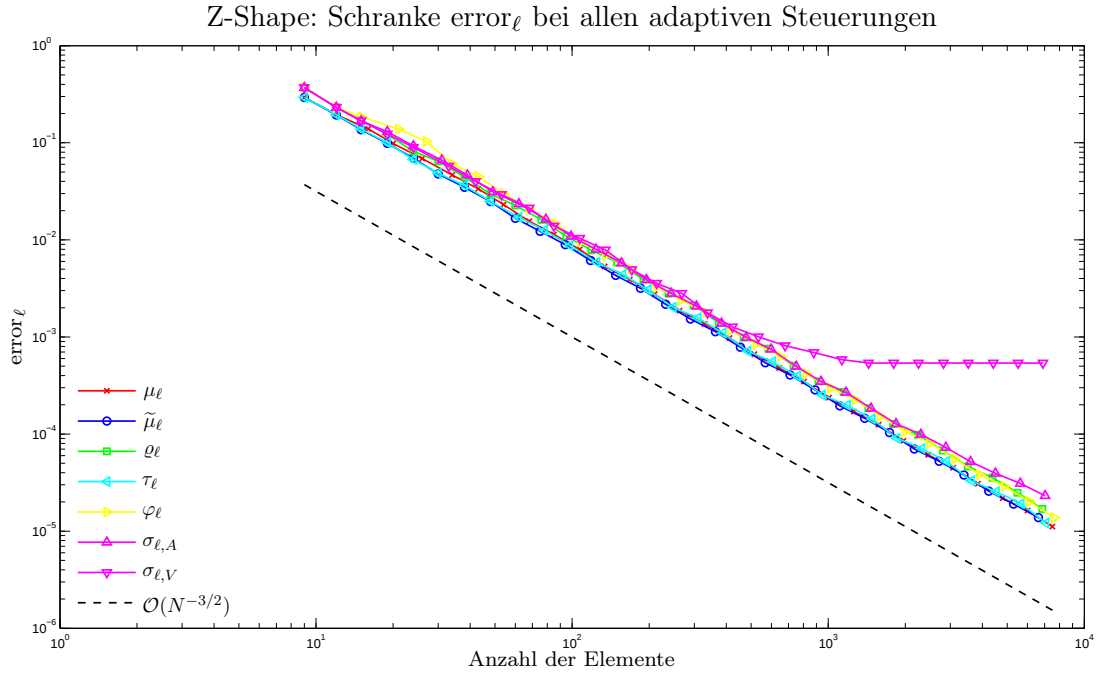


Abbildung 60: Z-Shape: Schranke error_ℓ bei allen adaptiven Steuerungen über Berechnungsdauer und Anzahl der Elemente

6.3.3 Schlitz-Problem

Abschließend untersuchen wir das Schlitz-Problem. Wir beginnen wieder mit der Schätzung der Regularität von $\phi \in H^{-1/2}$ und betrachten den in Abbildung 61 dargestellten Verlauf der Fehlerschranke error_ℓ und der beim Schlitz-Problem zulässigen Fehlerschätzer bei uniformer Netzverfeinerung. Mittels Gleichung (6.7) berechnen wir eine Konvergenzrate von $\|\phi - \Phi_\ell\|_V = \mathcal{O}(h^{1/2})$, was mit (6.5) zu $\phi \in L^2(\Gamma)$ führt. Damit weist die Lösung ϕ noch weniger Regularität auf, verglichen mit jenen aus den vorherigen Beispielen.

Die Abbildungen 62 bis 66 zeigen uns wieder, dass der adaptive Algorithmus mit allen betrachteten Verfeinerungsindikatoren einen $\mathcal{O}(N^{-3/2})$ -Konvergenzverlauf aufweist. Aus der Parallelität aller Kurven schließen wir wieder die Effizienz und Zuverlässigkeit, ohne dass dies von der hier herausgearbeiteten Analysis gewährleistet werden kann. In den Abbildungen 67 bis 71 ist ersichtlich, dass die jeweiligen Fehlerschätzer nicht empfindlich auf die Wahl des Verfeinerungsindikators reagieren. Sie liefern bei jeder adaptiven Steuerung beinahe identische Ergebnisse, insbesondere bietet Abbildung 72 eine gute Übersicht der Fehlerschätzer.

Aus Abbildung 73 schließen wir, dass alle adaptiven Steuerungen in etwa gleichwertig sind bezogen auf die Genauigkeit der aus jeder adaptiven Verfeinerungsstrategie gewonnenen Approximation $\Phi_\ell \approx \phi$ bzgl. der Energienorm, wenngleich der φ_ℓ -gesteuerte Algorithmus eine etwas höhere Fehlerschranke error_ℓ aufweist. Betrachten wir die Fehlerschranke error_ℓ über der kumulierten Berechnungsdauer des jeweiligen Algorithmus, sind wieder die Residualschätzer $\varrho_\ell, \varphi_\ell$ vorteilhafter, wobei auch hier der Faermann-Residualschätzer etwas abgeschlagen ist. Die notwendige Erweiterung des Codes für die Berechnung des Faermann-Residualschätzers schlägt sich demnach merklich im Kurvenverlauf der Fehlerschranke error_ℓ nieder.

Wir kommen zur Rechenzeit-Analyse aller adaptiv-gesteuerten Algorithmen: Betrachten wir zuerst Abbildung 74, so sehen wir wieder einen asymptotischen Zeitaufwand von $\mathcal{O}(N^2)$. Vergleichen wir diesen Plot mit jenem vom L-Shape in Abbildung 36, so sind keine gravierenden Unterschiede feststellbar. Die Codeänderungen beim Faermann-Schätzer wirken sich nicht in der Berechnungszeit aus und aus der vereinfachten rechten Seite $\mathbf{b}$ können die $(h - h/2)$ -Schätzer nur geringfügig profitieren.

Ein genaueres Bild erhalten wir durch die Aufschlüsselung der Berechnungsdauern der jeweiligen adaptiven Algorithmen in den Abbildungen 75 bis 79. Die Berechnungszeit der rechten Seite fällt im Vergleich zu jener beim L- bzw. Z-Shape nicht mehr ins Gewicht (vgl. $\mathbf{b} = -\mathbf{M}\mathbf{z}$ vs. $\mathbf{b} = \mathbf{K}\mathbf{g} + 1/2\mathbf{M}\mathbf{g}$ und die Vorüberlegungen in Abschnitt 6.1.3). Folglich fällt der beim L-Shape noch größte Anteil (bei den hier betrachteten adaptiven Algorithmen) weg.

Betrachten wir die $(h - h/2)$ -Fehlerschätzer, so sind selbst die kumulierten Zeiten von „Sonstige“, als relative Zeit immer noch unbedeutend, nun teurer in der Berechnung als die rechte Seite. Damit ist die für die Berechnung von $(h - h/2)$ -Schätzer gesteuerten Algorithmen benötigte Zeit nur mehr durch das Aufstellen der Einfachschichtpotentialmatrix $\hat{\mathbf{V}}$ ($\mathbf{V}$) und der Lösung des linearen Gleichungssystems $\hat{\mathbf{V}}\hat{\mathbf{x}} = \hat{\mathbf{b}}$ ($\mathbf{V}\mathbf{x} = \mathbf{b}$) bestimmt.

Bei den Residualschätzer-gesteuerten Algorithmen fällt nun der Hauptteil der (kumulierten) Berechnungszeit auf die Bestimmung der jeweiligen Fehlerschätzer.

Insgesamt zeigen alle BEM-Algorithmen mit den verschiedenen Steuerungen die maximal erreichbare $\mathcal{O}(N^{-3/2})$ -Konvergenzrate. Die Fehlerschranke ist bei allen Steuerungen nahezu iden-

tisch, wobei wieder die Residualschätzer die schnellsten Ergebnisse liefern. Auch die stabile Implementierung soll an dieser Stelle nochmal hervorgehoben werden.

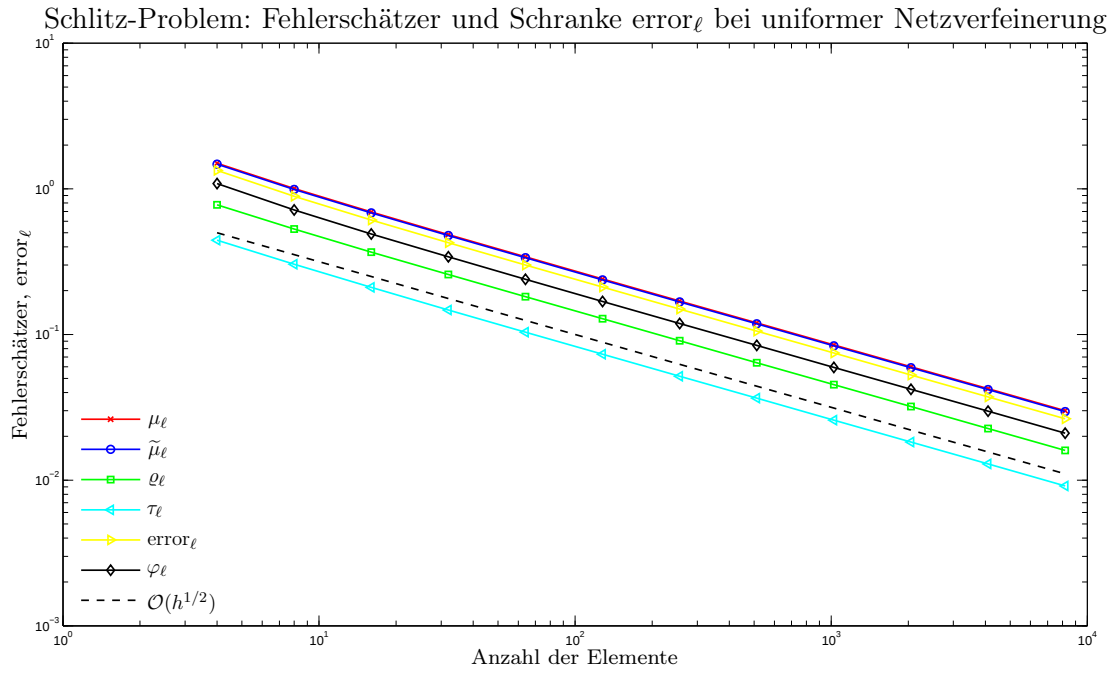


Abbildung 61: Schlitz-Problem: Fehlerschätzer und Schranke error_ℓ bei uniformer Netzverfeinerung

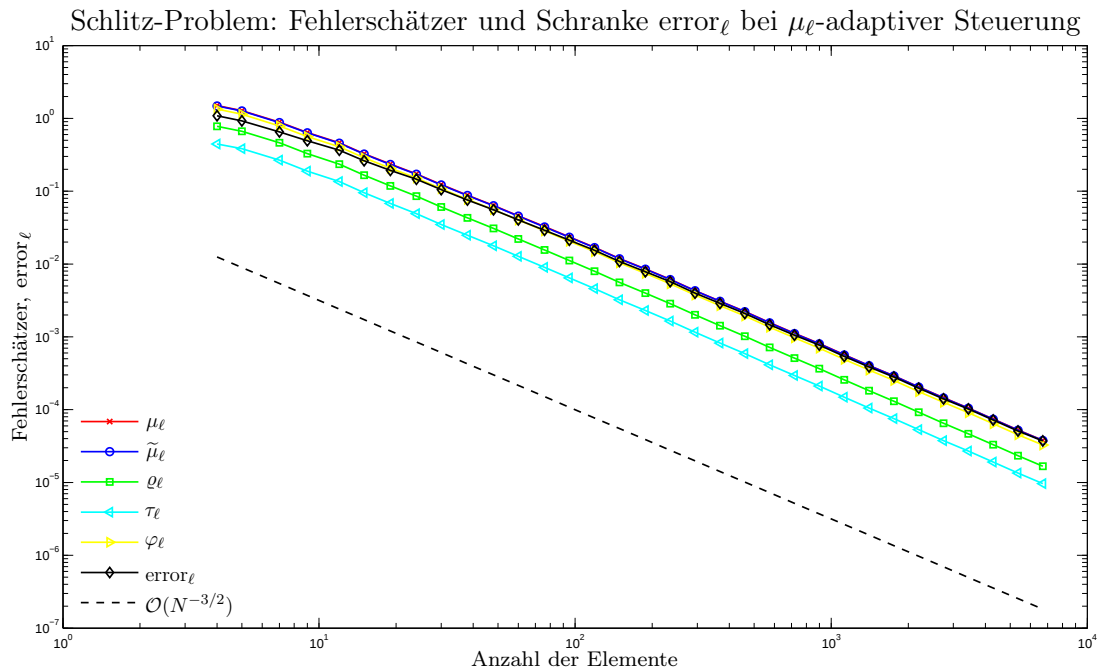


Abbildung 62: Schlitz-Problem: Fehlerschätzer und Schranke error_ℓ bei μ_ℓ -adaptiver Steuerung

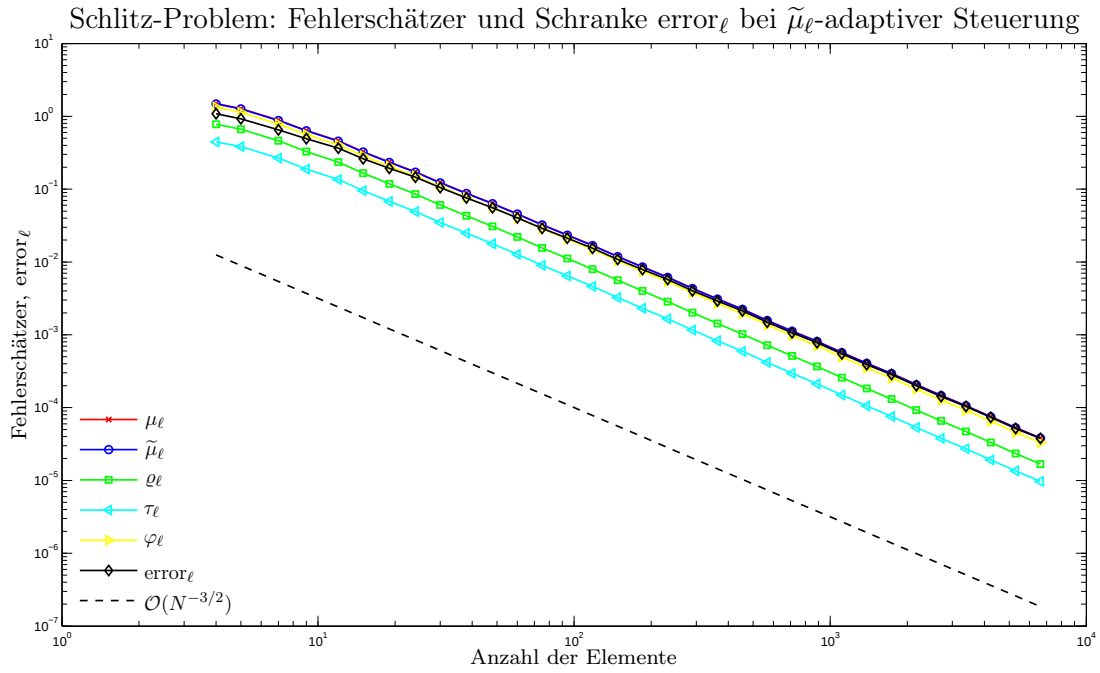


Abbildung 63: Schlitz-Problem: Fehlerschätzer und Schranke error_ℓ bei $\tilde{\mu}_\ell$ -adaptiver Steuerung

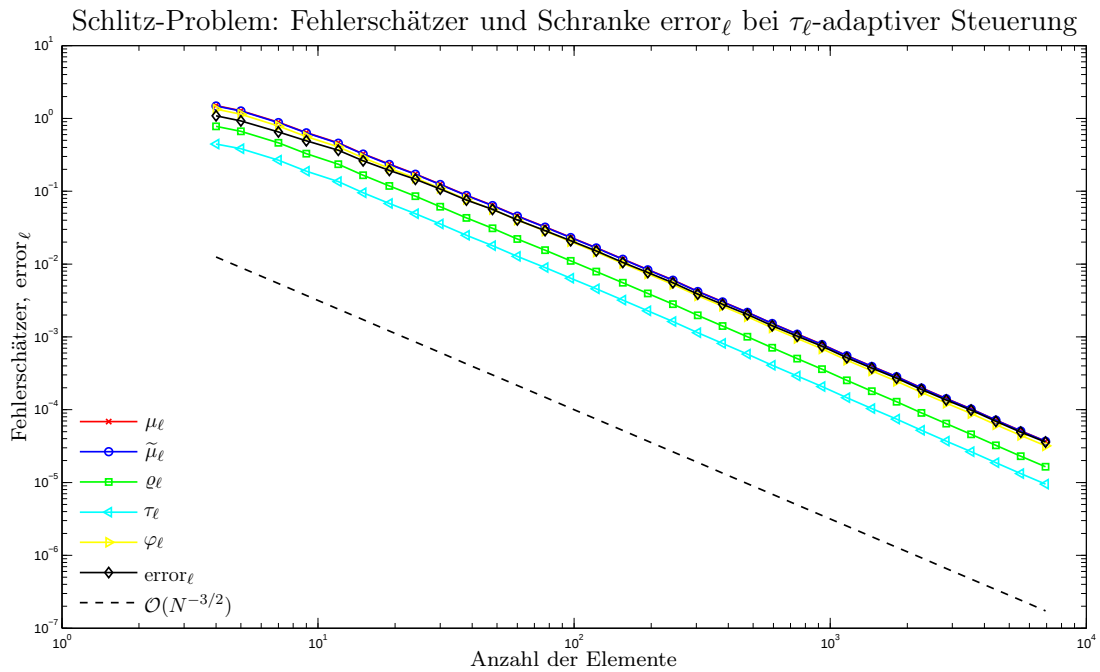


Abbildung 64: Schlitz-Problem: Fehlerschätzer und Schranke error_ℓ bei τ_ℓ -adaptiver Steuerung

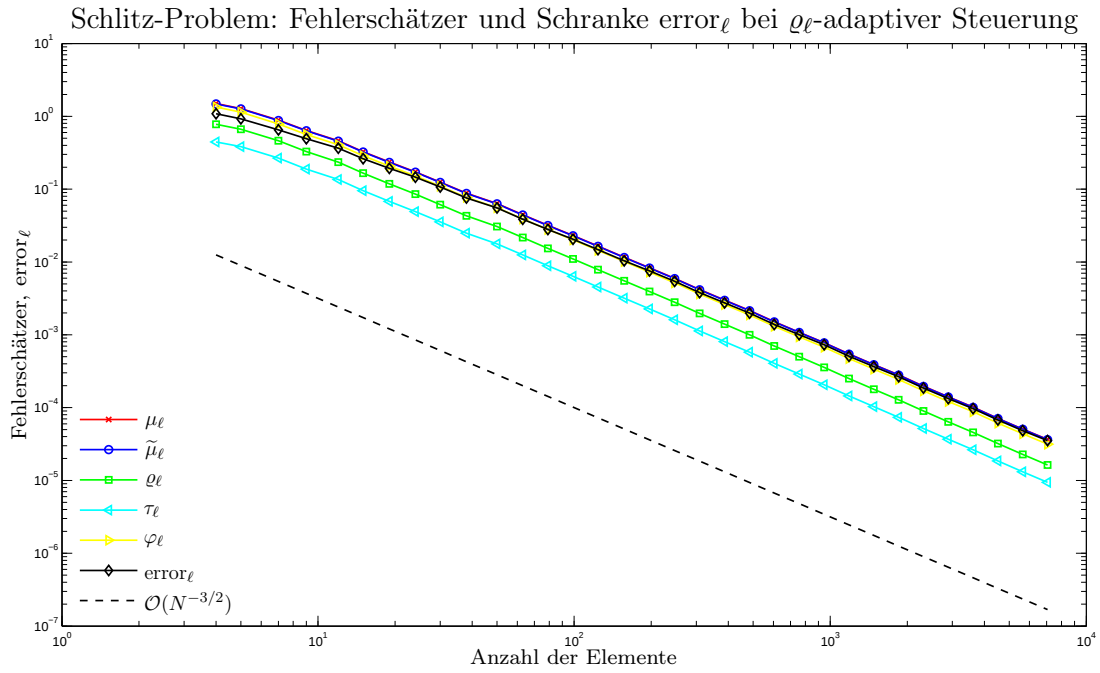


Abbildung 65: Schlitz-Problem: Fehlerschätzer und Schranke error_ℓ bei ϱ_ℓ -adaptiver Steuerung

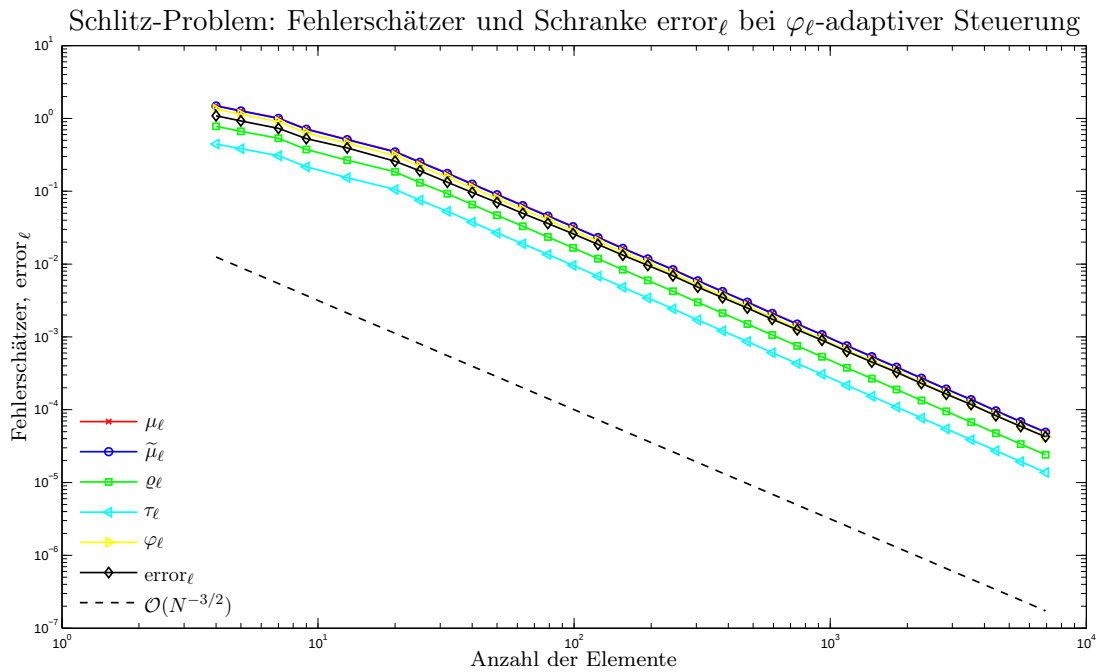


Abbildung 66: Schlitz-Problem: Fehlerschätzer und Schranke error_ℓ bei φ_ℓ -adaptiver Steuerung

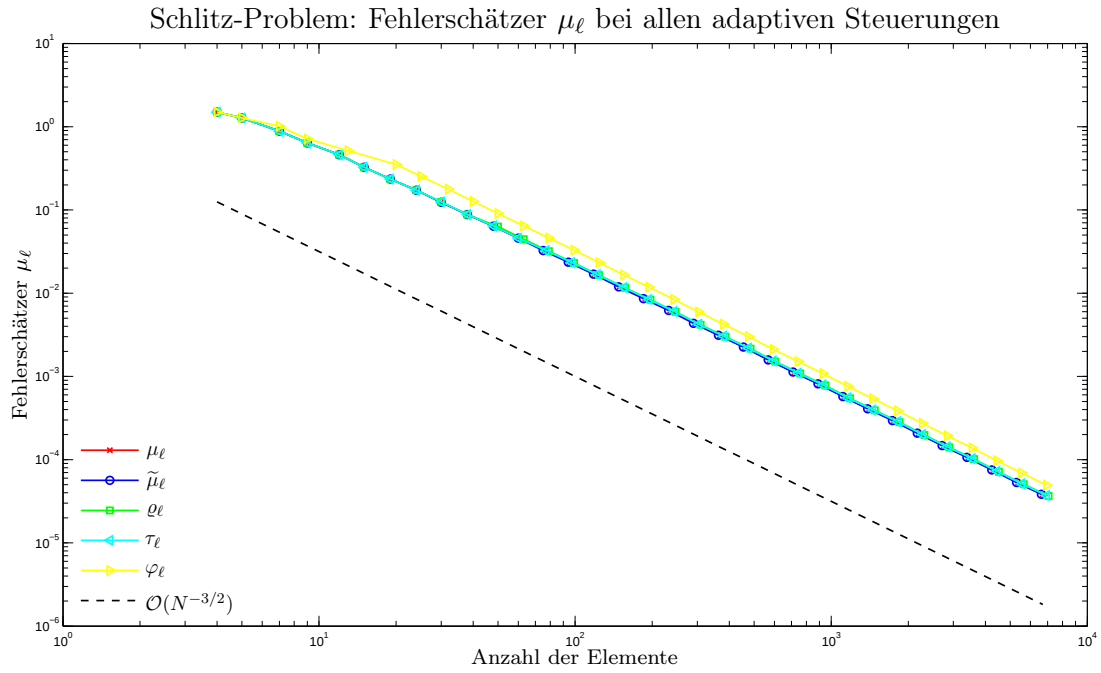


Abbildung 67: Schlitz-Problem: Fehlerschätzer μ_ℓ bei allen adaptiven Steuerungen

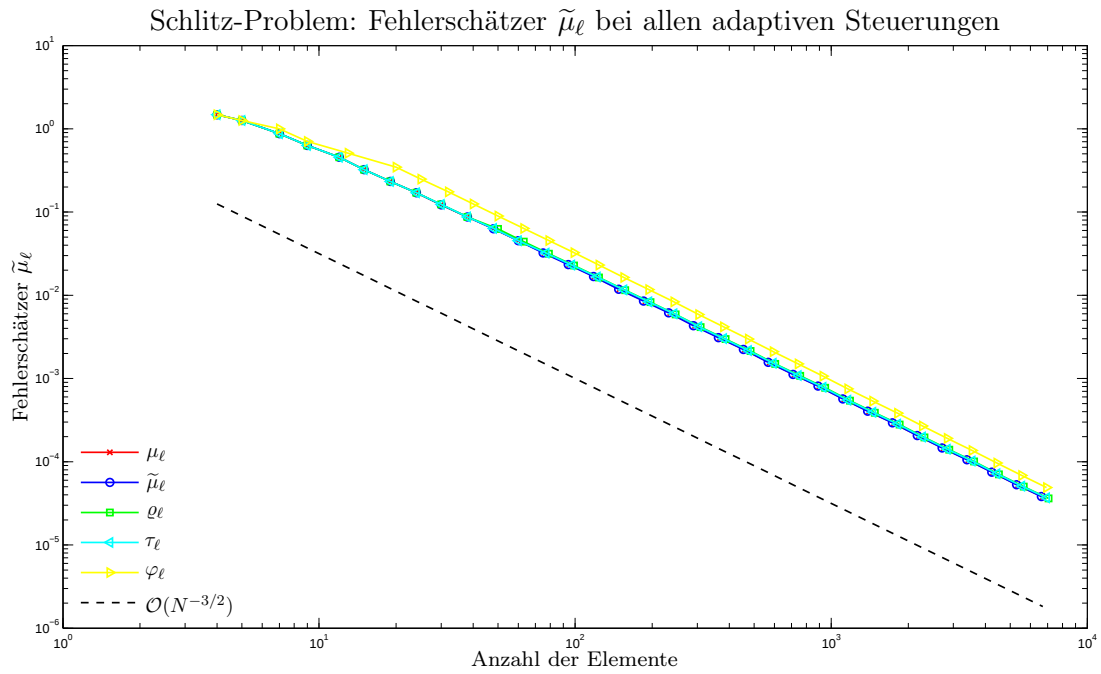


Abbildung 68: Schlitz-Problem: Fehlerschätzer $\tilde{\mu}_\ell$ bei allen adaptiven Steuerungen

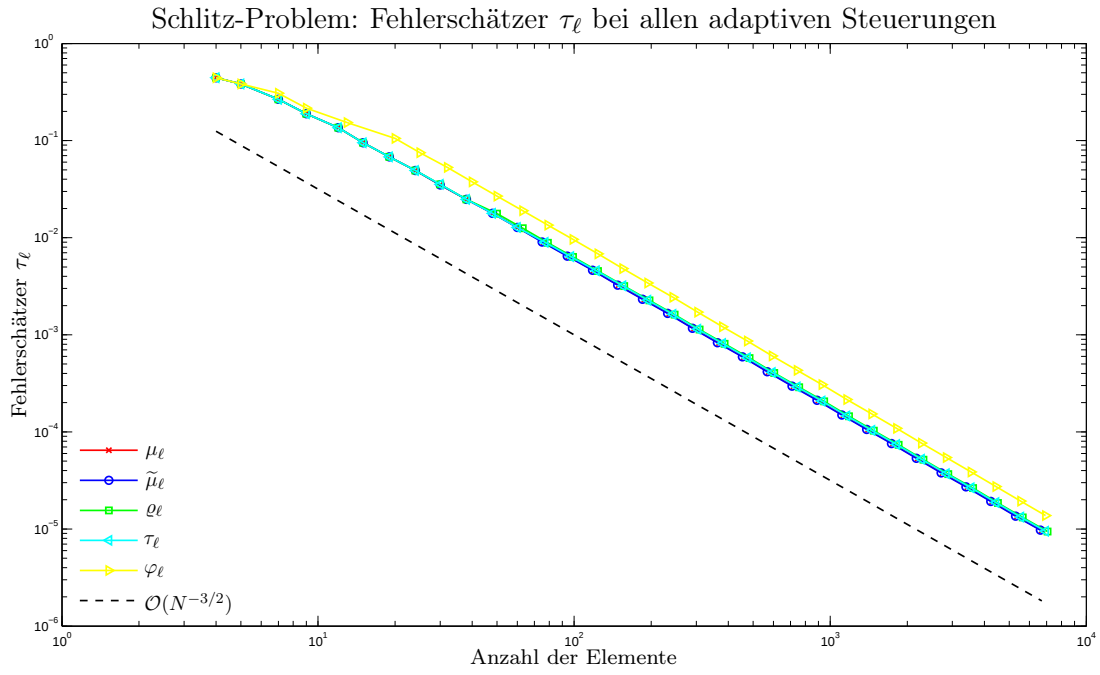


Abbildung 69: Schlitz-Problem: Fehlerschätzer τ_ℓ bei allen adaptiven Steuerungen

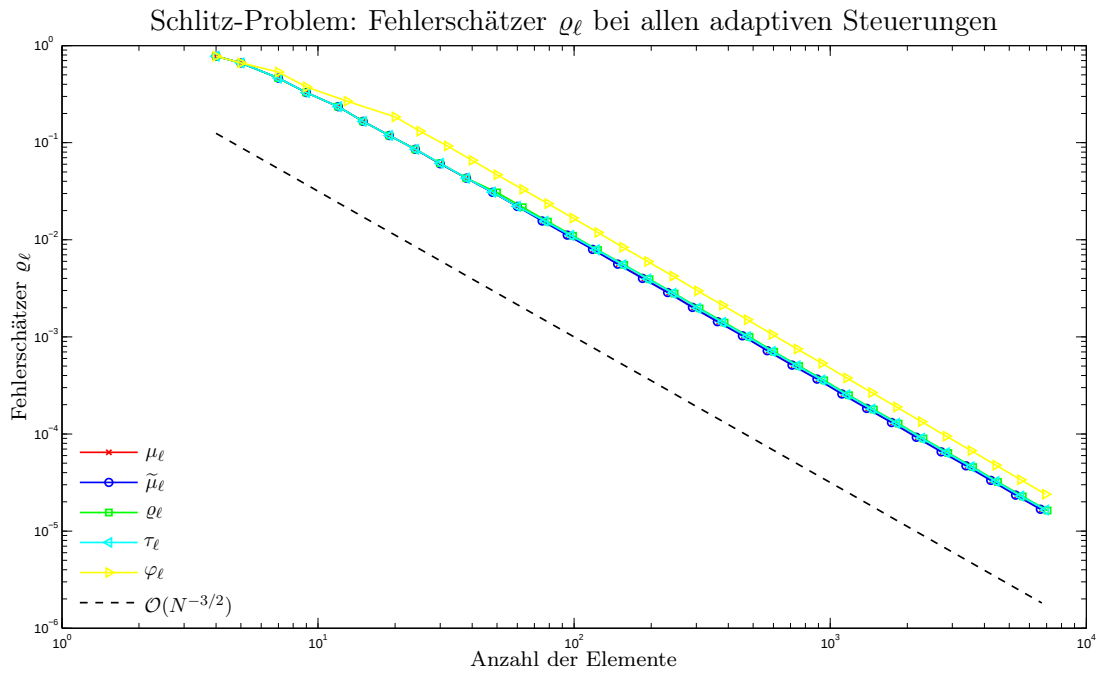


Abbildung 70: Schlitz-Problem: Fehlerschätzer ϱ_ℓ bei allen adaptiven Steuerungen

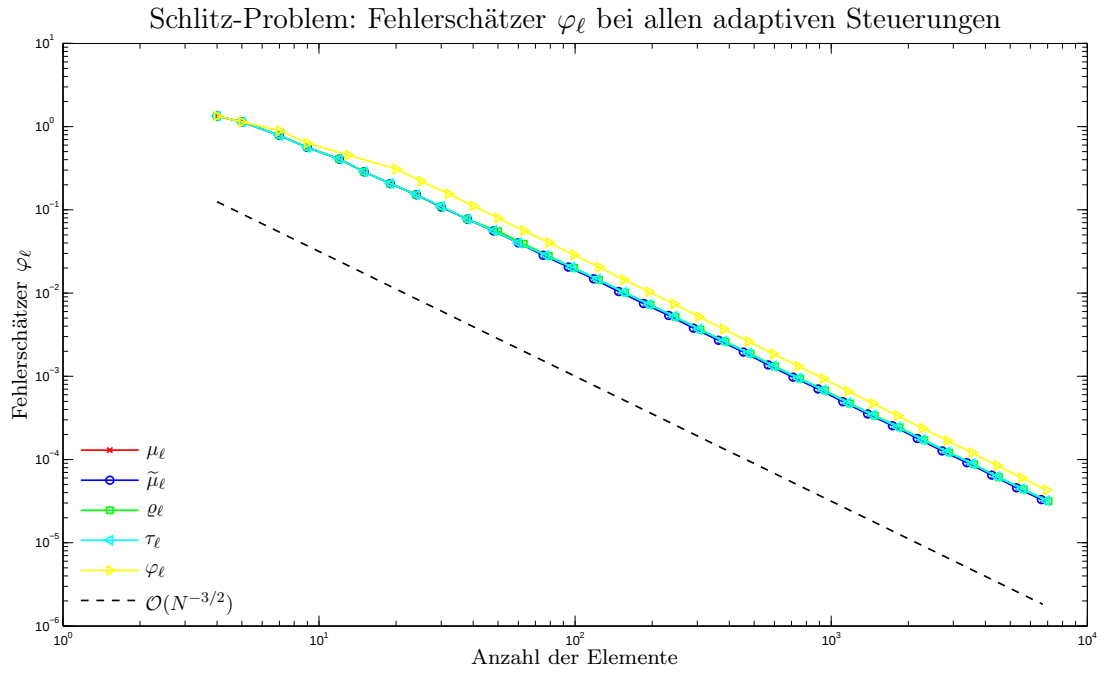


Abbildung 71: Schlitz-Problem: Fehlerschätzer φ_ℓ bei allen adaptiven Steuerungen

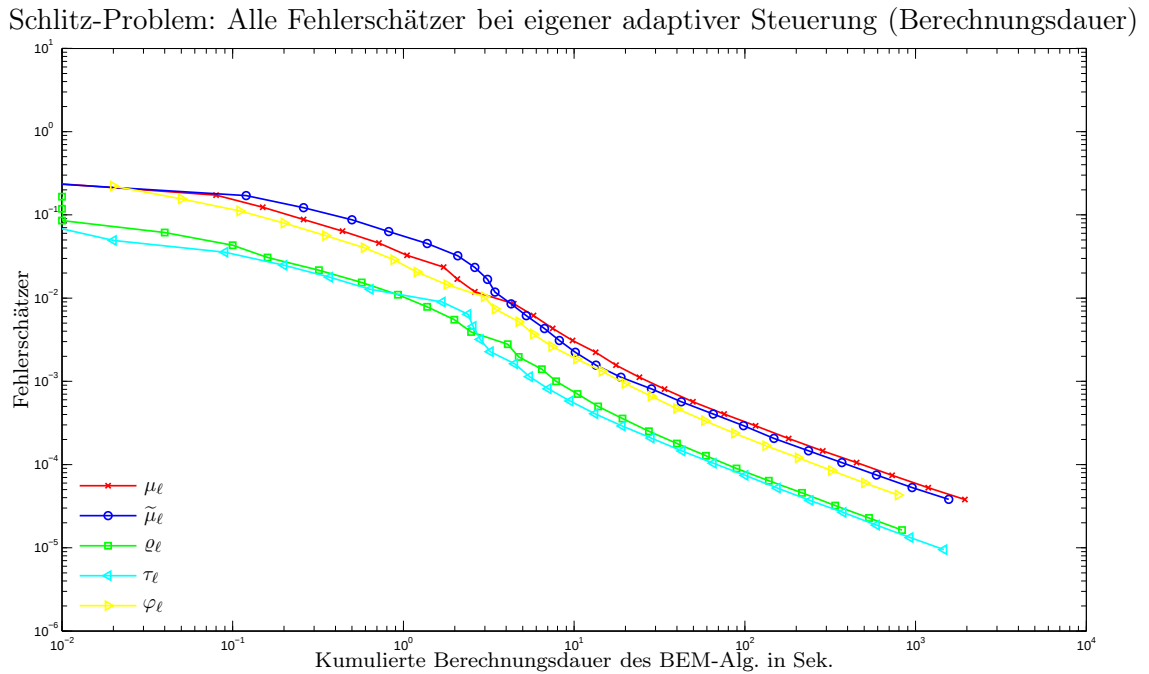
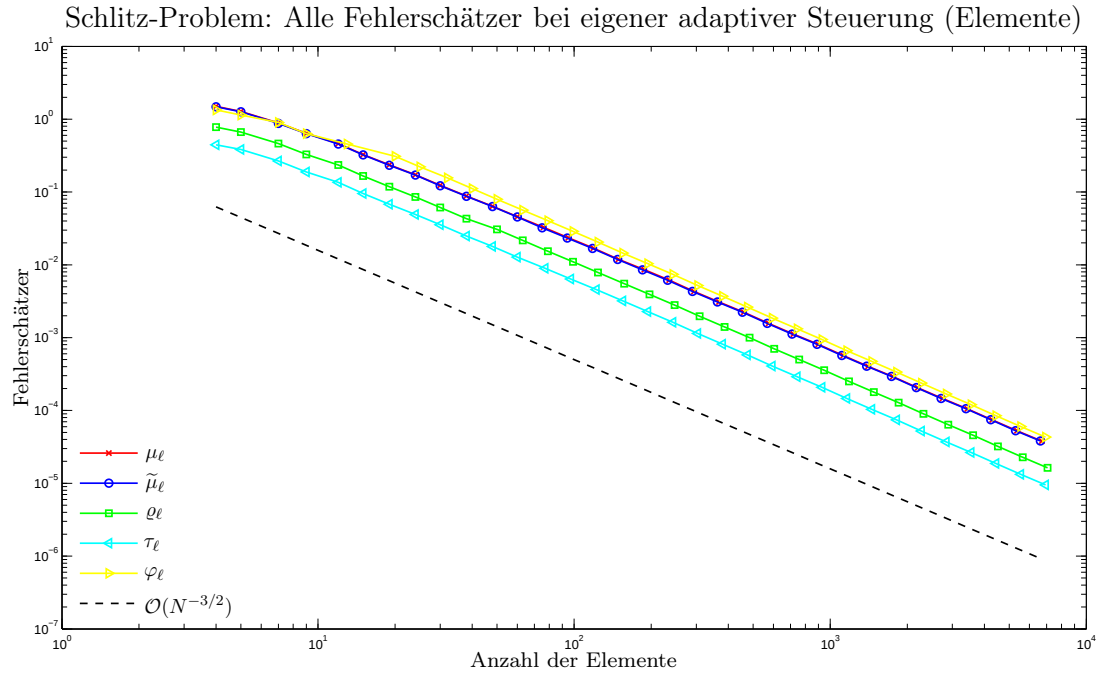


Abbildung 72: Schlitz-Problem: Alle Fehlerschätzer bei eigener adaptiver Steuerung

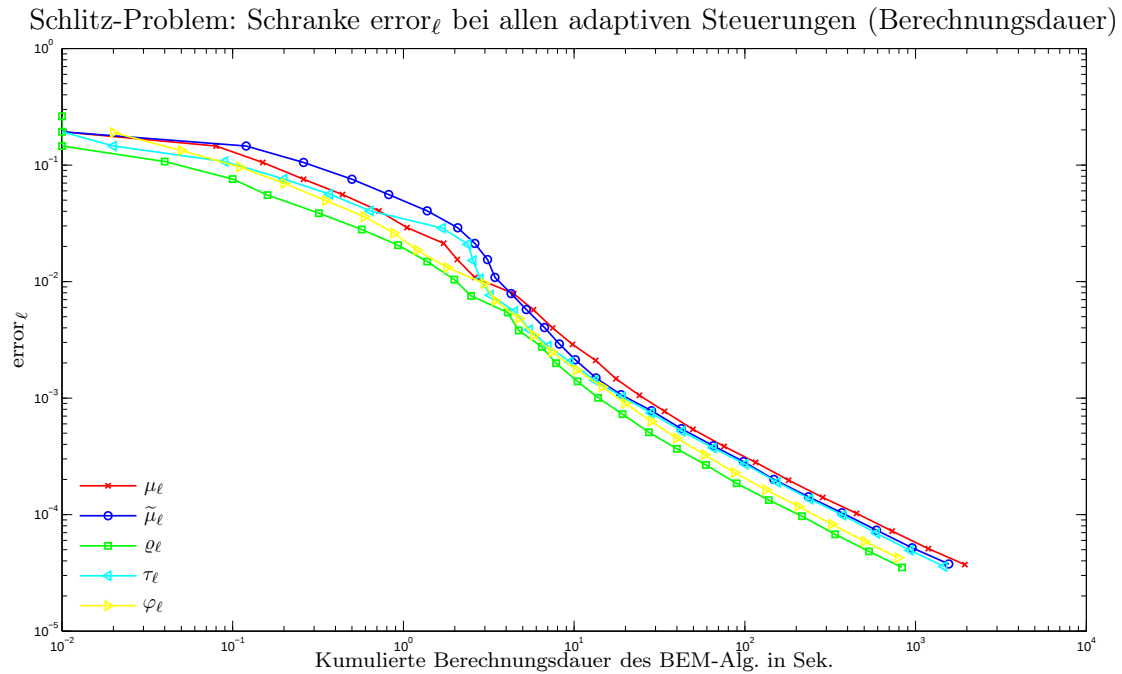
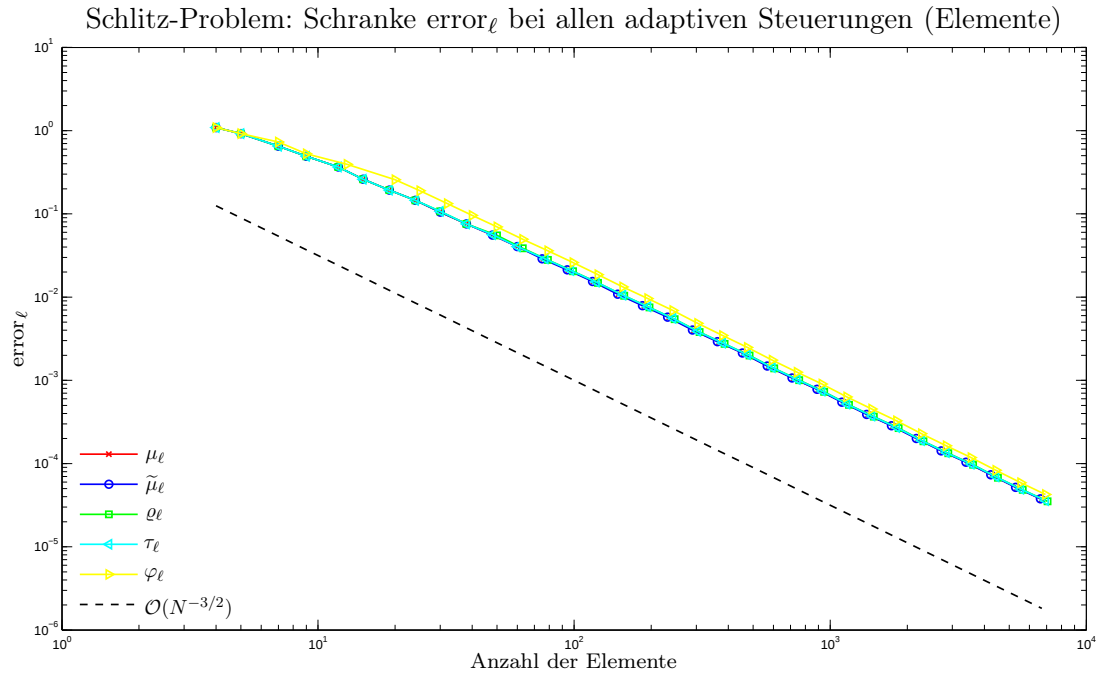


Abbildung 73: Schlitz-Problem: Schranke error_ℓ bei allen adaptiven Steuerungen über Berechnungsdauer und Anzahl der Elemente

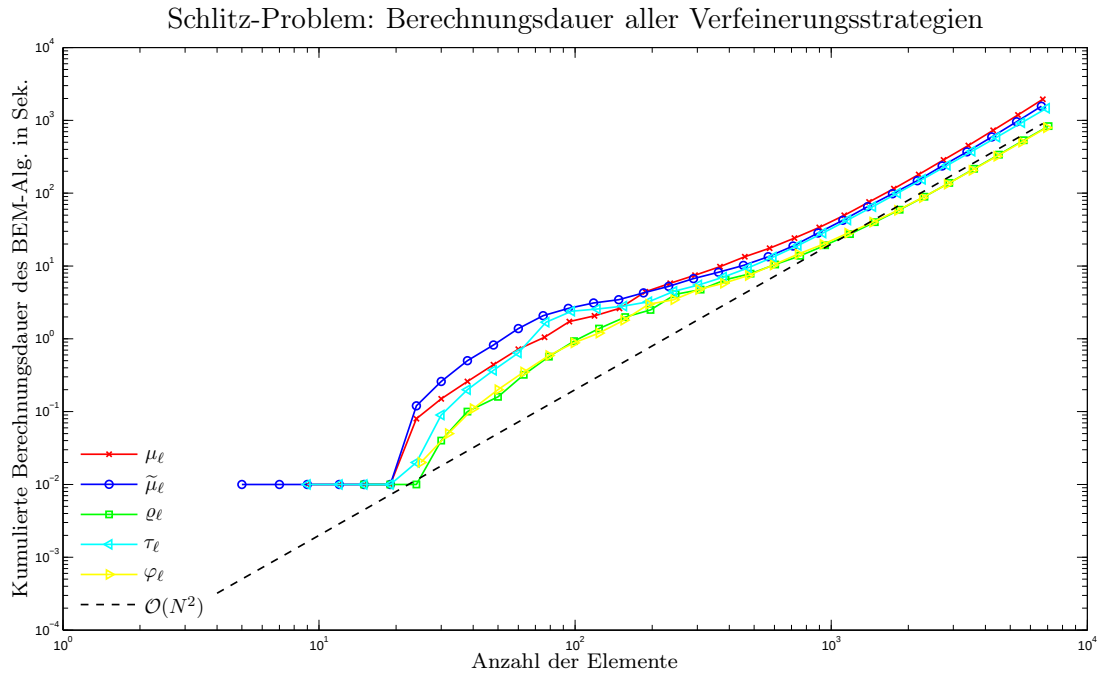


Abbildung 74: Schlitz-Problem: Berechnungsdauer aller Verfeinerungsstrategien

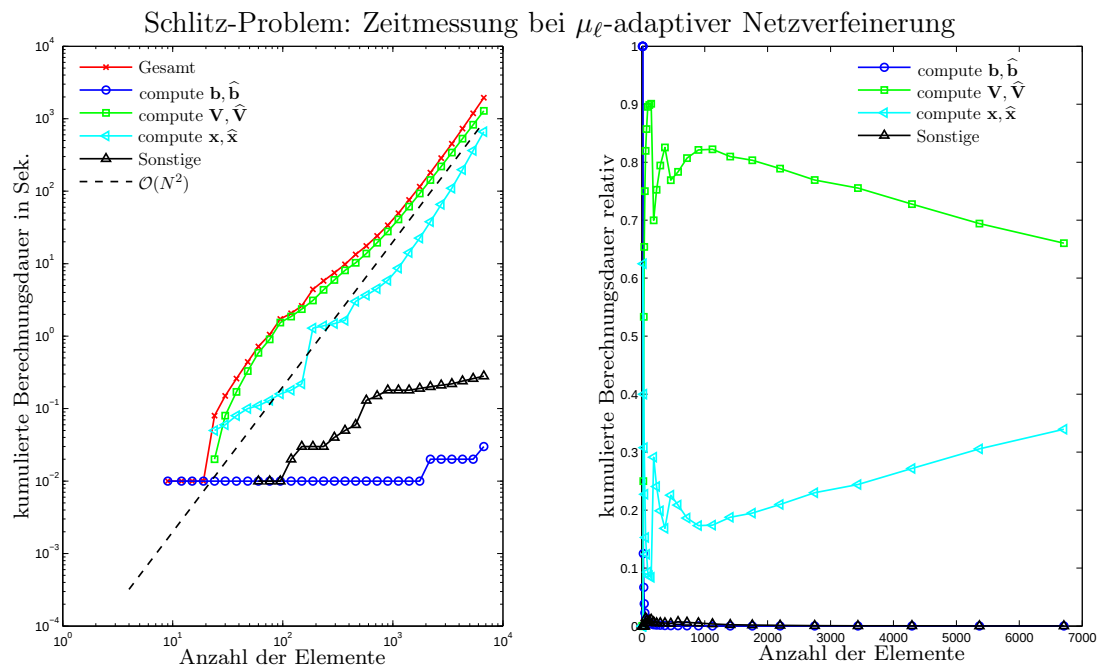
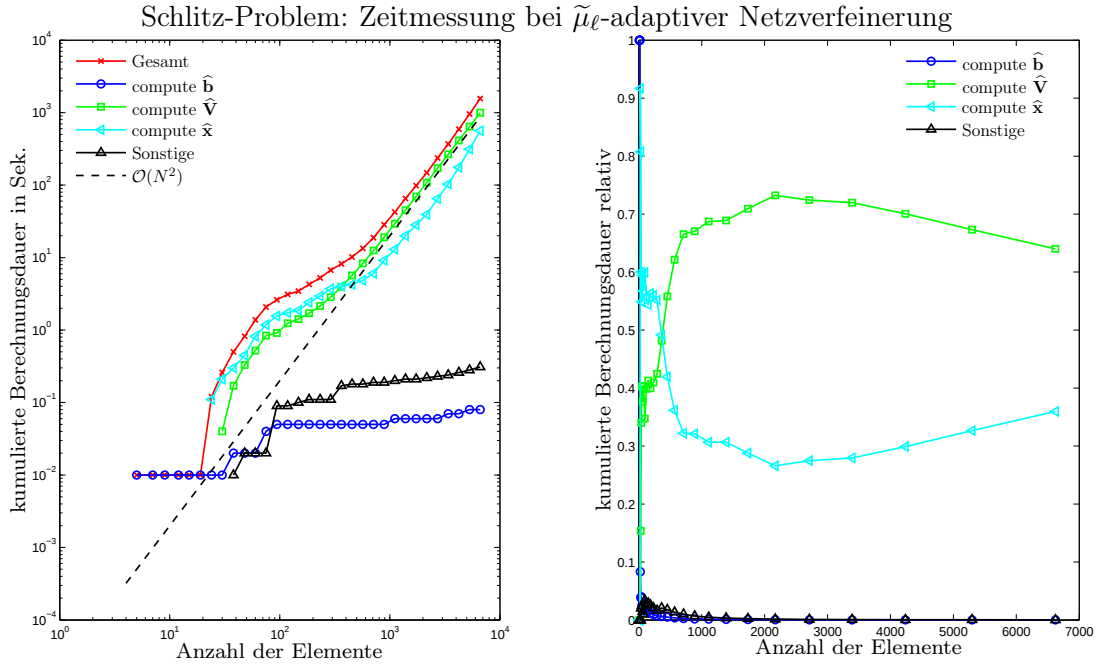
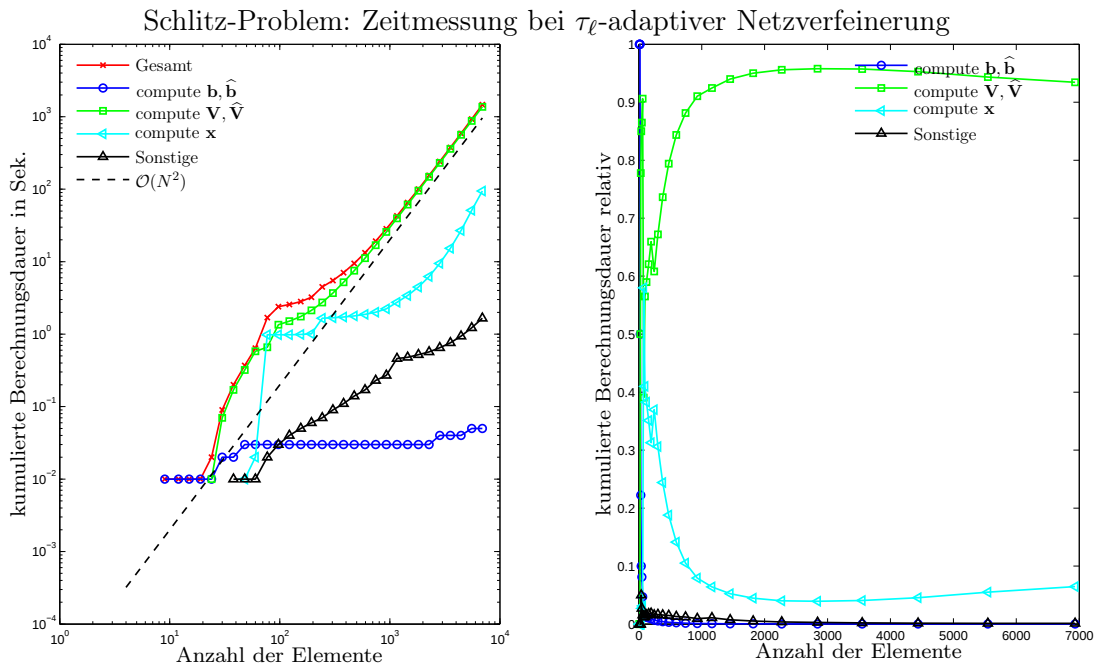


Abbildung 75: Schlitz-Problem: Zeitmessung bei μ_ℓ -adaptiver Netzverfeinerung


 Abbildung 76: Schlitz-Problem: Zeitmessung bei $\tilde{\mu}_\ell$ -adaptiver Netzverfeinerung

 Abbildung 77: Schlitz-Problem: Zeitmessung bei τ_ℓ -adaptiver Netzverfeinerung

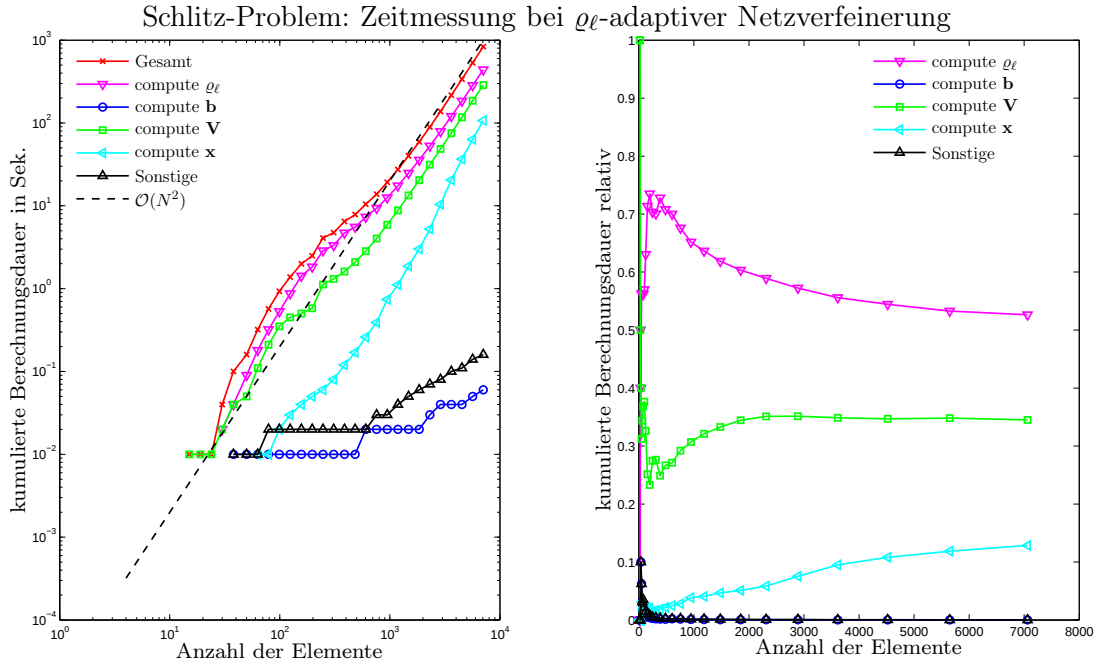


Abbildung 78: Schlitz-Problem: Zeitmessung bei ϱ_ℓ -adaptiver Netzverfeinerung

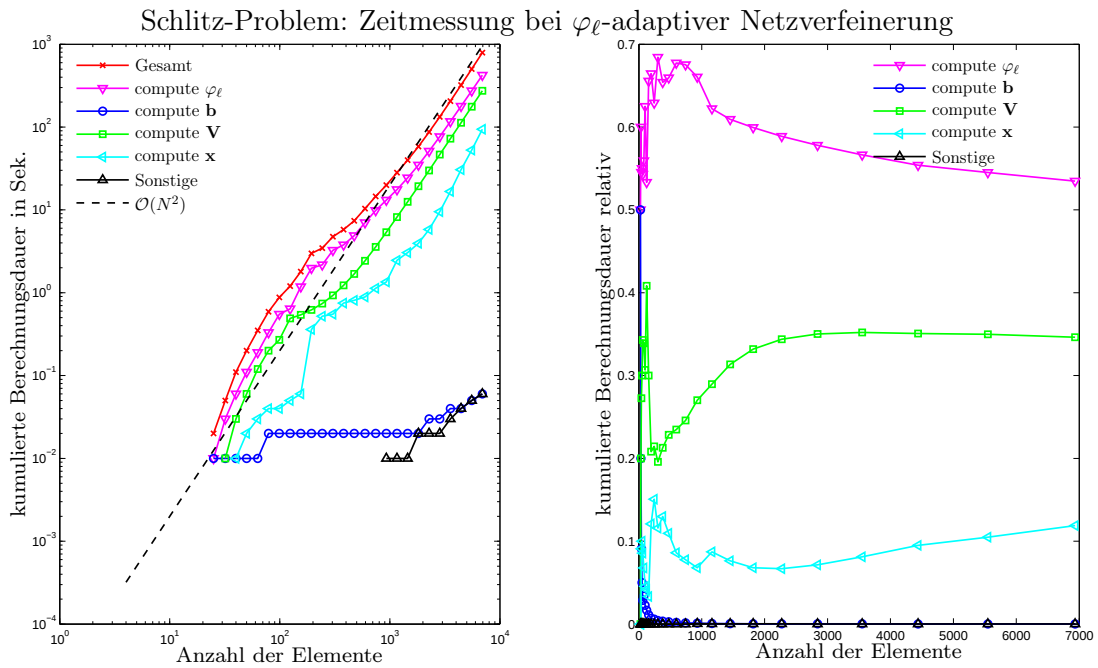


Abbildung 79: Schlitz-Problem: Zeitmessung bei φ_ℓ -adaptiver Netzverfeinerung

A Anhang

A.1 Programmcode Faermann-Residualschätzer für das Schlitz-Problem

Listing A.1: computeEstSlpFaermann_Schlitz

```
1 function ind = computeEstSlpFaermann_Schlitz(coordinates,elements,phih,gh)
2
3 *** Gaussian quadrature on [-1,1] with 2 nodes and exactness 3
4 quad_nodes = [-1 1]/sqrt(3);
5 quad_sum = [1 1 2 2; 1 2 1 2];
6
7 *** elementwise interpolation is done in (gauss_left,gauss_right,midpoint)
8 interpolation_nodes = [quad_nodes,0];
9
10 *** define constants
11 nE = size(elements,1);
12 nC = size(coordinates,1);
13 nQ = length(quad_nodes);
14 nI = length(interpolation_nodes);
15
16 *** build vector of evaluations points as (nI*nE x 2)-matrix
17 a = coordinates(elements(:,1),:);
18 b = coordinates(elements(:,2),:);
19 sx = reshape(a,2*nE,1)*(1-interpolation_nodes) ...
20       + reshape(b,2*nE,1)*(1+interpolation_nodes);
21 sx = 0.5*reshape(sx',nI*nE,2);
22
23 *** evaluate residual = V*phih + x elementwise at (left, right, middle)
24 residual = evaluateV(coordinates,elements,phih,sx) + sx(:,1);
25 residual = reshape(residual,nI,nE)';
26
27 *** ADDITION WHEN SCHLITZ-PROBLEM:
28 elements = [elements; 5,1];
29 sx = [sx; zeros(3,2)];
30 residual = [residual; zeros(1,3)];
31
32 *** determine right neighbour Ek and left neighbour Ei of element Ej
33 node2rightElement = zeros(nC,1);
34 node2leftElement = zeros(nC,1);
35 node2rightElement(elements(:,1)) = (1:nE+1)';
36 node2leftElement(elements(:,2)) = (1:nE+1)';
37 element2rightNeighbour = node2rightElement(elements(:,2));
38 element2leftNeighbour = node2leftElement(elements(:,1));
39
40 *** compute int(Ej x Ej) = int_{Ej} int_{Ej} |p(s)-p(t)|^2 / |s-t|^2 dt ds
41 *** note: after transformation to [-1,1] and polynomial interpolation
42 *** the integrand is of the type |q(s)-q(t)|^2 with a polynomial q
43 *** and can thus be integrated analytically
44 intEjEj = 3*(residual(:,2)-residual(:,1)).^2 + 6*(residual(:,1) ...
45       +residual(:,2)-2*residual(:,3)).^2;
46
47 *** compute local mesh-size
```

```

48 h = sqrt(sum((coordinates(elements(:,1),:)-coordinates(elements(:,2),:)).^2,2));
49
50 %*** compute int(Ej x Ek) = int_{Ej} int_{Ek} |R(x)-R(y)|^2 / |x-y|^2 dGamma(x,y)
51 j = [1:nE+1]';
52 k=element2rightNeighbour(j);
53 res_j = residual(j,quad_sum(1,:));
54 res_k = residual(k,quad_sum(2,:));
55 nom = (res_j - res_k).^2;
56 jidx = [(nI*j - (nI-1)) (nI*j-1)];
57 kidx = [(nI*k - (nI-1)) (nI*k-1)];
58 jidx = reshape(jidx(:,quad_sum(1,:))',nQ*nQ*(nE+1),1);
59 kidx = reshape(kidx(:,quad_sum(2,:))',nQ*nQ*(nE+1),1);
60 denom = sum((sx(jidx,:) - sx(kidx,:)).^2,2);
61 denom = reshape(denom,nQ*nQ,(nE+1))';
62 intEjEk = 0.25*sum(nom./denom,2).*h(j).*h(k);
63
64 %*** return ind(j) = int(Ej x Ej) + 2*int(Ej x Ek) + int(Ek x Ek)
65 %***                      +int(Ei x Ei) + 2*int(Ei x Ej) + int(Ej x Ej)
66 ind = 2*( intEjEj(1:nE)+intEjEk(1:nE)+intEjEk(element2leftNeighbour(1:nE)) )...
67      +intEjEj(element2rightNeighbour(1:nE))+intEjEj(element2leftNeighbour(1:nE));

```

Bezugnehmend auf die Dokumentation von Listing 8 des Faermann-Residualschätzers auf geschlossenen Rändern ist hier nur eine kurze Erklärung hinsichtlich der durchgeführten Änderungen:

- In Zeile 24 wird das Residuum $R_\ell = V\Phi_\ell + x$ berechnet.
- Die Zeilen 28–30 bilden die wesentliche Erweiterung: Es wird dem Array `elements` ein virtuelles Element $\tilde{E} = [z_5, z_1]$ beigefügt, wobei aufgrund der Definition des Netzes $\mathcal{E}_0$ $z_5 = (1, 0)$ und $z_1 = (-1, 0)$ gilt. Weiters wird auch das $(nE \cdot nQ \times 2)$ -Array `sx` um zwei weitere (zwischen -1 und 1 beliebige) Auswertepunkte ergänzt. Anschließend wird das $(nE \times nI)$ -Array `residual` um eine Zeile mit Null-Einträgen erweitert, sodass $R_\ell|_{\tilde{E}} \equiv 0$ gilt.
- Im weiteren Programmcode muss punktuell auf die richtige Indizierung geachtet werden, um am Ende die Fehlerindikatoren $\tilde{\varphi}_\ell(E_j)^2$ für $j = 1, \dots, N$ zu erhalten.

A.2 Programmcode des $\sigma_{\ell,V}$ -adaptiv gesteuerten BEM-Algorithmus

Listing A.2: adaptivSigmaV

```

1 function adaptivSigmaV(coordinates,elements,uD,nEmax,theta,phi_N)
2 %*** adaptive mesh-refining algorithm
3 counter = 1;
4
5 while size(elements,1) < nEmax
6     %
7     %—— start time-measurement for adaptive mesh-refinement
8     %
9     time_ = cputime;

```

```

10
11 %*** discretize Dirichlet data and compute data oscillations
12 [osc_,uDh] = computeOscDirichlet(coordinates,elements,uD);
13 time_computeOscDirichlet = cputime - time_;
14
15 %*** compute coarse-mesh solution for error estimation
16 b = buildSymmRHS(coordinates,elements,uDh);
17 time_buildSymmRHS = cputime - time_ - time_computeOscDirichlet;
18
19 V = buildV(coordinates,elements);
20 time_buildV = cputime - time_ - time_computeOscDirichlet ...
21             - time_buildSymmRHS;
22
23 x = V\b;
24 time_solve = cputime - time_ - time_computeOscDirichlet ...
25             - time_buildSymmRHS - time_buildV;
26
27 %*** compute estimator sigmaV:
28 [sigmaVInd_,sigmaV_] = computeEstSigmaV(coordinates,elements,x,uDh);
29 time_estimator = cputime - time_ - time_computeOscDirichlet ...
30             - time_buildSymmRHS - time_buildV - time_solve;
31
32 %*** mark elements for refinement
33 marked = markElements(theta,1/4,sigmaVInd_+osc_);
34
35 %*** generate new mesh
36 [coordinates_new,elements_new] ...
37     = refineBoundaryMesh(coordinates,elements,marked);
38 time_generateNewMesh = cputime - time_ - time_computeOscDirichlet ...
39             - time_buildSymmRHS - time_buildV - time_solve ...
40             - time_estimator;
41
42 %
43 %— stop time-measurement for adaptive mesh-refinement
44 %
45 time_ = cputime - time_;
46 if counter>1
47     time_ = time_sigmaV(counter-1)+time_;
48     time_computeOscDirichlet = time_sigmaV_computeOscDirichlet(counter-1) ...
49         + time_computeOscDirichlet;
50     time_buildSymmRHS = time_sigmaV_buildSymmRHS(counter-1) + time_buildSymmRHS;
51     time_buildV = time_sigmaV_buildV(counter-1) + time_buildV;
52     time_solve = time_sigmaV_solve(counter-1) + time_solve;
53     time_estimator = time_sigmaV_estimator(counter-1) + time_estimator;
54     time_generateNewMesh = time_sigmaV_generateNewMesh(counter-1) ...
55         + time_generateNewMesh;
56 end
57
58 %*** free obsolete memory
59 clear V b
60
61 %*** generate uniformly refined mesh
62 [coordinates_fine,elements_fine,father2son] ...
63     = refineBoundaryMesh(coordinates,elements);

```

```
64     %*** compute fine-mesh solution
65     V_fine = buildV(coordinates_fine,elements_fine);
66     b_fine = buildSymmRHS(coordinates_fine,elements_fine,uD(coordinates_fine));
67     x_fine = V_fine\b_fine;
68
69     %*** compute all the other estimators:
70     fae_ = computeEstSlpFaermann(coordinates,elements,x,uDh);
71     mu_ = computeEstSlpMu(coordinates,elements,father2son,x_fine,x);
72     muTilde_ = computeEstSlpMuTilde(coordinates,elements,father2son,x_fine);
73     rho_ = computeEstSlpResidual(coordinates,elements,x,uDh);
74     sigmaA_ = computeEstSigmaA(coordinates,elements,x,uDh);
75     tau_ = computeEstSlpTau(father2son,V_fine,b_fine,x);
76
77     %*** compute error bound
78     err_ = computeErrNeumann(coordinates,elements,x,phi_N);
79     err_sigmaV(counter) = sqrt(sum(err_));
80
81     %*** free obsolete memory
82     clear coordinates_fine elements_fine father2son
83     clear V_fine b_fine x_fine x
84
85     %*** store computed data
86     nE_sigmaV(counter) = size(elements,1);
87     osc_sigmaV(counter) = sqrt(sum(osc_));
88     time_sigmaV(counter) = time_;
89     time_sigmaV_computeOscDirichlet(counter) = time_computeOscDirichlet;
90     time_sigmaV_buildSymmRHS(counter) = time_buildSymmRHS;
91     time_sigmaV_buildV(counter) = time_buildV;
92     time_sigmaV_solve(counter) = time_solve;
93     time_sigmaV_estimator(counter) = time_estimator;
94     time_sigmaV_generateNewMesh(counter) = time_generateNewMesh;
95     fae_sigmaV(counter) = sqrt(sum(fae_));
96     mu_sigmaV(counter) = sqrt(sum(mu_));
97     muTilde_sigmaV(counter) = sqrt(sum(muTilde_));
98     rho_sigmaV(counter) = sqrt(sum(rho_));
99     sigmaA_sigmaV(counter) = sqrt(sum(sigmaA_));
100    sigmaV_sigmaV(counter) = sqrt(sigmaV_);
101    sigmaVInd_sigmaV(counter) = sqrt(sum(sigmaVInd_));
102    tau_sigmaV(counter) = sqrt(sum(tau_));
103
104    %*** increase counter and iterate
105    counter = counter + 1;
106    coordinates = coordinates_new;
107    elements = elements_new;
108 end
109
110 save('adaptiv.mat','-append');
```

Die Dokumentation des obigen Codes:

- Es wird das Netz $\mathcal{E}_0$ in Form von `coordinates` und `elements` eingelesen. Das Dirichlet-Datum g und die (vorgegebene) Lösung des Neumann-Datums ϕ werden als Funktionen `uD` bzw. `phi_N` übergeben. Weiters sind noch die maximale Anzahl der Elemente `nEmax` und

der Parameter **theta** für den adaptiven Algorithmus gegeben, wobei in unserem Falle stets **theta** = 1/4 gilt (Zeile 1).

- Nach der Initialisierung des Counters **counter** als Indikator des ($\ell = \mathbf{counter}$)-ten Netzes wird der adaptive BEM-Algorithmus so lange durchgeführt bis die maximale Elementanzahl **nEmax** überschritten wird (Zeile 5).
- Es beginnt die Zeitmessung der adaptiven Verfeinerungsstrategie (Zeile 9). Bezugnehmend auf die Beschreibung der $\sigma_{\ell,V}$ -adaptiven Verfeinerung in Algorithmus 6.5 werden alle Schritte durchgeführt und die je Schritt benötigte Zeit berechnet. Dabei wird die jeweilige Zeit auf dem ℓ -ten Netz als Summe aller Berechnungsdauern des jeweiligen Schrittes auf den vorhergehenden Netzen gespeichert. Nicht mehr benötigter Speicher wird sofort gelöscht (Zeilen 12–83).
- Alle Ergebnisse des ℓ -ten Netzes werden in den ℓ -ten Array-Eintrag geschrieben (Zeilen 86–102).
- Abschließend werden alle Resultate in die Datei **adaptiv.mat** beigefügt (Zeile 110).

Literatur

- [1] AURADA, M., M. EBNER, M. FEISCHL, S. FERRAZ-LEITE, P. GOLDENITS, M. KARKULIK, M. MAYR und D. PRAETORIUS: *Hilbert (Release 2): A MATLAB Implementation of Adaptive BEM*. ASC Report No. 14, 2010. Institute for Analysis and Scientific Computing, Vienna University of Technology.
- [2] AURADA, M., P. GOLDENITS und D. PRAETORIUS: *Convergence of Data Perturbed Adaptive Boundary Element Methods*. ASC Report No. 40, 2009. Institute for Analysis and Scientific Computing, Vienna University of Technology.
- [3] CARSTENSEN, C.: *Efficiency of a posteriori BEM-error estimates for first-kind integral equations on quasi-uniform meshes*. Math. Comput., 65(213):69–84, 1996.
- [4] CARSTENSEN, C.: *An a posteriori error estimate for a first-kind integral equation*. Math. Comp., 66(217):139–155, 1997.
- [5] CARSTENSEN, C. und B. FAERMANN: *Mathematical foundation of a posteriori error estimates and adaptive mesh-refining algorithms for boundary integral equations of the first kind*. Eng. Anal. Bound. Elem., 25(7):497–509, 2001.
- [6] CARSTENSEN, C. und D. PRAETORIUS: *A posteriori error control in adaptive quadrature boundary element analysis for a logarithmic-kernel integral equation of the first kind*. SIAM J. Sci. Comput., 25(1):259–283, 2003.
- [7] DÖRFLER, W.: *A convergent adaptive algorithm for Poisson’s equation*. SIAM J. Numer. Anal., 33(3):1106–1124, 1996.
- [8] ERATH, C., S. FERRAZ-LEITE, S. FUNKEN und D. PRAETORIUS: *Energy norm based a posteriori error estimation for boundary element methods in two dimensions*. Appl. Numer. Math., 59(11):2713–2734, 2009.
- [9] FERRAZ-LEITE, S. und D. PRAETORIUS: *Simple a posteriori error estimators for the h-version of the boundary element method*. Computing, 83(4):135–162, 2008.
- [10] GRAHAM, I., W. HACKBUSCH und S. SAUTER: *Finite elements on degenerate meshes: inverse-type inequalities and applications*. IMA J. Numer. Anal., 25(2):379–407, 2005.
- [11] KREYSZIG, E.: *Introductory functional analysis with applications*. John Wiley & Sons. Inc., 1978.
- [12] MCLEAN, W.: *Strongly elliptic systems and boundary integral equations*. Cambridge: Cambridge University Press, 2000.
- [13] OSWALD, P.: *Multilevel norms for $H^{-1/2}$* . Computing, 61(3):235–255, 1998.
- [14] PRAETORIUS, D.: *Vorlesungsmanskript zur LVA Scientific Computing*. Institute for Analysis and Scientific Computing, Vienna University of Technology, 2010.

- [15] SAUTER, S. und C. SCHWAB: *Randelementmethoden. Analyse, Numerik und Implementierung schneller Algorithmen*. Stuttgart: Teubner, 2004.
- [16] SCHULZ, H. und O. STEINBACH: *A new a posteriori error estimator in adaptive direct boundary element methods: The Dirichlet problem*. *Calcolo*, 37(2):79–96, 2000.
- [17] STEINBACH, O.: *Adaptive boundary element methods based on computational schemes for Sobolev norms*. *SIAM J. Sci. Comput*, 22(2):604–616, 2000.
- [18] STEINBACH, O.: *Numerische Näherungsverfahren für elliptische Randwertprobleme. Finite Elemente und Randelemente*. Stuttgart: Teubner, 2003.