Philipp Bringmann, Carsten Carstensen*, and Julian Streitberger

Local parameter selection in the C^0 interior penalty method for the biharmonic equation

https://doi.org/10.1515/jnma-2023-0028
Received February 24, 2023; revised August 24, 2023; accepted August 26, 2023

Abstract: The symmetric C^0 interior penalty method is one of the most popular discontinuous Galerkin methods for the biharmonic equation. This paper introduces an automatic local selection of the involved stability parameter in terms of the geometry of the underlying triangulation for arbitrary polynomial degrees. The proposed choice ensures a stable discretization with guaranteed discrete ellipticity constant. Numerical evidence for uniform and adaptive mesh refinement and various polynomial degrees supports the reliability and efficiency of the local parameter selection and recommends this in practice. The approach is documented in 2D for triangles, but the methodology behind can be generalized to higher dimensions, to non-uniform polynomial degrees, and to rectangular discretizations. An appendix presents the realization of our proposed parameter selection in various established finite element software packages.

Keywords: C^0 interior penalty method, discontinuous Galerkin method, biharmonic equation, implementation, local parameter selection, penalty parameter

Classification: 65N12, 65N15, 65N30, 65N50, 65Y20

1 Introduction

A classical conforming finite element method for the plate problem originates from the work of Argyris [4] with a quintic polynomial ansatz space $P_5(T)$ and 21 degrees of freedom on each triangle T. The practical application appears less prominent due to the higher computational efforts [28], although recent works [23, 33] establish a highly efficient adaptive multilevel solver for the hierarchical Argyris FEM. Due to their easier implementation, classical nonconforming FEMs [19, 25, 26, 30] and discontinuous Galerkin schemes appear advantageous compared to conforming discretizations. At least the growing number of publications on these schemes indicates a higher popularity.

Discontinuous Galerkin finite element methods [6, 16, 19, 27, 44] such as the C^0 interior penalty method are a popular choice for fourth-order problems in order to avoid C^1-conforming finite elements. Their main drawback is the dependence of the stability on a sufficiently large penalty parameter. The choice of this parameter is often heuristical and based on the individual experience of the user. This paper proposes an explicit geometry-dependent and local selection of the penalty parameter that guarantees the stability of the resulting scheme.

Given a source term $f \in L^2(\Omega)$ in a bounded polygonal Lipschitz domain $\Omega \subset \mathbb{R}^2$, let $u \in V := H^2_0(\Omega)$ be the weak solution to the biharmonic equation $\Delta^2 u = f$,

$$a(u, v) := \int_\Omega \mathrm{D}^2 u : \mathrm{D}^2 v \,\mathrm{d}x = \int_\Omega f v \,\mathrm{d}x \quad \forall v \in V. \tag{1.1}$$

The Riesz representation theorem applies in the Hilbert space (V, a) and proves well-posedness of the formulation (1.1). Elliptic regularity theory verifies that $f \in L^2(\Omega)$ implies $u \in H^{2+\alpha}(\Omega) \cap H^2_0(\Omega)$ [1, 8, 31, 37]. The pure clamped boundary conditions in the model example lead to $\alpha > 1/2$.

For a regular triangulation $\mathcal{T}$ of Ω into closed triangles and the polynomial degree $k \geqslant 2$, the symmetric C^0 interior penalty method (C0IP) seeks $u_{\mathrm{IP}} \in V_h := S^k_0(\mathcal{T}) := P_k(\mathcal{T}) \cap H^1_0(\Omega)$ in the Lagrange finite element space

Philipp Bringmann, Julian Streitberger, TU Wien, Institute of Analysis and Scientific Computing, Wien, Austria.
***Corresponding author: Carsten Carstensen,** Department of Mathematics, Humboldt-Universität zu Berlin, Germany.
Email: cc@math.hu-berlin.de

with

$$A_h(u_{\mathrm{IP}}, v_{\mathrm{IP}}) = \int_\Omega f v_{\mathrm{IP}} \,\mathrm{d}x \quad \forall v_{\mathrm{IP}} \in V_h. \tag{1.2}$$

The three contributions to the bilinear form $A_h : V_h \times V_h \to \mathbb{R}$ defined, for $u_{\mathrm{IP}}, v_{\mathrm{IP}} \in V_h$, by

$$A_h(u_{\mathrm{IP}}, v_{\mathrm{IP}}) := a_{\mathrm{pw}}(u_{\mathrm{IP}}, v_{\mathrm{IP}}) - (\jmath(u_{\mathrm{IP}}, v_{\mathrm{IP}}) + \jmath(v_{\mathrm{IP}}, u_{\mathrm{IP}})) + c_{\mathrm{IP}}(u_{\mathrm{IP}}, v_{\mathrm{IP}}) \tag{1.3}$$

originate from the piecewise integration by parts in the derivation of (1.1) (see [16]). While the boundedness of A_h follows from standard arguments, its coercivity is subject to the assumption of a sufficiently large penalty parameter $\sigma_{\mathrm{IP},E} > 0$ in the bilinear form $c_{\mathrm{IP}} : V_h \times V_h \to \mathbb{R}$ defined, for $u_{\mathrm{IP}}, v_{\mathrm{IP}} \in V_h$, by

$$c_{\mathrm{IP}}(u_{\mathrm{IP}}, v_{\mathrm{IP}}) := \sum_{E\in\mathcal{E}} \frac{\sigma_{\mathrm{IP},E}}{h_E} \int_E [\nabla u_{\mathrm{IP}} \cdot \nu_E]_E [\nabla v_{\mathrm{IP}} \cdot \nu_E]_E \,\mathrm{d}s \tag{1.4}$$

with the normal jumps $[\nabla v_{\mathrm{IP}} \cdot \nu_E]_E$ (see [16, 24]). This paper introduces a local parameter

$$\sigma_{\mathrm{IP},E} := \begin{cases} \dfrac{3ak(k-1)h_E^2}{8}\left(\dfrac{1}{|T_+|} + \dfrac{1}{|T_-|}\right) & \text{if } E = \partial T_+ \cap \partial T_- \in \mathcal{E}(\Omega) \\ \dfrac{3ak(k-1)h_E^2}{2|T_+|} & \text{if } E \in \mathcal{E}(T_+) \cap \mathcal{E}(\partial\Omega). \end{cases} \tag{1.5}$$

The penalty parameter $\sigma_{\mathrm{IP},E}$ in (1.5) contains a prefactor $a > 1$. Theorem 3.1 below establishes that *every* choice of $a > 1$ leads to guaranteed stability with stability constant at least $\varkappa = 1 - 1/\sqrt{a}$. In the case of very large penalization with $a \to \infty$, i.e., $\sigma_{\mathrm{IP},E} \to \infty$, the lower bound $\varkappa$ tends to 1. This fine-tuning of the penalty parameter $\sigma_{\mathrm{IP},E}$ enables strong penalization as employed in [9] for the analysis of optimal convergence rates of adaptive discontinuous Galerkin methods.

Numerical experiments confirm the guaranteed stability and exhibit rate-optimal convergence of an adaptive C0IP for various polynomial degrees in Section 4 below. The computation of the discrete inf-sup constants (as certain eigenvalues of the discrete operator) reveals little overestimation only and recommends the proposed local parameter selection in practise. A detailed investigation of the influence of the parameter a reveals that large penalty parameters lead to a substantial increase of the condition numbers of the system matrix. Hence, we recommend a small choice of a, e.g., $a = 2$, in order to avoid large condition numbers. Alternatively, a strong penalization requires the application of suitable preconditioners for C0IP [11, 13, 17, 18, 20, 21].

Example 1.1. On a uniform triangulation into right isosceles triangles of the same area, the penalty parameter (1.5) for the quadratic C0IP method reads

$$\sigma_{\mathrm{IP},E} = \begin{cases} 3a/4 & \text{for a cathetus } E \in \mathcal{E}(\Omega) \text{ in the interior} \\ 3a/2 & \text{for a hypotenuse } E \in \mathcal{E}(\Omega) \text{ in the interior} \\ 6a & \text{for a cathetus } E \in \mathcal{E}(\partial\Omega) \text{ on the boundary} \\ 12a & \text{for a hypotenuse } E \in \mathcal{E}(\partial\Omega) \text{ on the boundary.} \end{cases}$$

A choice of a close to 1 leads to smaller penalty parameters for the majority of interior edges compared to values from the literature, e.g., $\sigma_{\mathrm{IP},E} = 5$ in [14].

The remaining parts of the paper are organized as follows. Section 2 specifies the notation such that Section 3 can present the main result of the paper in Theorem 3.1. Numerical experiments investigate the stability of the scheme with the suggested automatic penalty selection in Section 4. The coercivity constants are computed numerically and compared with the theoretically established value. Moreover, the performance of an adaptive mesh-refinement algorithm is examined. Appendix A presents the numerical realization of this automatic choice of the penalty parameter to existing FEM software packages such as the unified form language [35, 36, 39], deal.II [5], and NGSolve [42]. The implementation of the C^0 interior penalty method in these packages turns out to be very compact. A complete and accessible documentation of a self-contained MATLAB code for the lowest-order discretization is available in [22, App. B].

2 Notation

Standard notation of Lebesgue and Sobolev spaces, their norms, and L^2 scalar products applies throughout the paper. Instead of the space V in the weak formulation (1.1), discontinuous Galerkin methods employ the piecewise Sobolev space

$$H^2(\mathcal{T}) := \{v \in L^2(\Omega) : \forall T \in \mathcal{T},\ v|_T \in H^2(T) := H^2(\mathrm{int}(T))\} \tag{2.1}$$

with respect to a shape-regular triangulation $\mathcal{T}$ of the domain Ω into closed triangles $T \in \mathcal{T}$ with interior $\mathrm{int}(T) \subset \Omega$. Define the space of piecewise polynomials

$$P_k(\mathcal{T}) := \{v \in L^2(\Omega) : \forall T \in \mathcal{T},\ v|_T \in P_k(T)\}$$

of total degree at most $k \in \mathbb{N}_0$. For $v \in H^2(\mathcal{T})$, the piecewise application of the distributional derivatives leads to the piecewise Hessian $\mathrm{D}^2_{\mathrm{pw}} v \in L^2(\Omega;\mathbb{S})$, $(\mathrm{D}^2_{\mathrm{pw}} v)|_T := \mathrm{D}^2(v|_T)$ with values in the space $\mathbb{S} \subset \mathbb{R}^{2\times 2}$ of symmetric 2×2 matrices.

The piecewise Sobolev functions $v \in H^2(\mathcal{T})$ allow for the evaluation of averages $\langle v\rangle_E$ and jumps $[v]_E$ across an edge $E \in \mathcal{E}$. Each interior edge $E \in \mathcal{E}(\Omega)$ of length $h_E := |E|$ is the common edge of exactly two triangles $T_+, T_- \in \mathcal{T}$, written $E = \partial T_+ \cap \partial T_-$. Then

$$[v]_E := (v|_{T_+})|_E - (v|_{T_-})|_E, \qquad \langle v\rangle_E := \frac{1}{2}(v|_{T_+})|_E + \frac{1}{2}(v|_{T_-})|_E. \tag{2.2}$$

The unit normal vector ν_E is oriented such that $\nu_E \cdot \nu_{T_\pm}|_E = \pm 1$ for the outward unit normal vectors $\nu_{T_\pm}$ of $T_\pm$. For each boundary edge $E \in \mathcal{E}(\partial\Omega)$, let $T_+ \in \mathcal{T}$ denote the unique triangle with edge $E \in \mathcal{E}(T_+)$ and set $[v]_E := \langle v\rangle_E := (v|_{T_+})|_E$. Analogous definitions apply for vector- or matrix-valued polynomials. Let $\mathcal{T}$ denote a shape regular triangulation of the polygonal Lipschitz domain Ω into closed triangles and $\mathcal{V}$ (resp. $\mathcal{V}(\Omega)$ or $\mathcal{V}(\partial\Omega)$) the set of all (resp. interior or boundary) vertices [10, 15]. Let $\mathcal{E}$ (resp. $\mathcal{E}(\Omega)$ or $\mathcal{E}(\partial\Omega)$) be the set of all (resp. interior or boundary) edges. For each triangle $T \in \mathcal{T}$ of area $|T|$, let $\mathcal{V}(T)$ denote the set of its three vertices and $\mathcal{E}(T)$ the set of its three edges. Abbreviate the edge patch by $\omega_E := \mathrm{int}(T_+ \cup T_-) \subseteq \Omega$ for an interior edge $E = \partial T_+ \cap \partial T_- \in \mathcal{E}(\Omega)$ and by $\omega_E := \mathrm{int}(T_+) \subseteq \Omega$ for a boundary edge $E \in \mathcal{E}(T_+) \cap \mathcal{E}(\partial\Omega)$.

The remaining contributions to the bilinear form A_h in (1.3) consist of the piecewise energy scalar product $a_{\mathrm{pw}} : V_h \times V_h \to \mathbb{R}$ and the jump term $\mathcal{J} : V_h \times V_h \to \mathbb{R}$ defined, for $u_{\mathrm{IP}}, v_{\mathrm{IP}} \in V_h$, by

$$\begin{aligned} a_{\mathrm{pw}}(u_{\mathrm{IP}}, v_{\mathrm{IP}}) &:= \sum_{T\in\mathcal{T}} \int_T \mathrm{D}^2 u_{\mathrm{IP}} : \mathrm{D}^2 v_{\mathrm{IP}}\, \mathrm{d}x \\ \mathcal{J}(u_{\mathrm{IP}}, v_{\mathrm{IP}}) &:= \sum_{E\in\mathcal{E}} \int_E \langle (\mathrm{D}^2_{\mathrm{pw}} u_{\mathrm{IP}}\, \nu_E)\cdot \nu_E\rangle_E\, [\nabla v_{\mathrm{IP}} \cdot \nu_E]_E\, \mathrm{d}s. \end{aligned} \tag{2.3}$$

Note that a different convention for the orientation of the normal vector ν_E in [16] leads to other signs in the bilinear form A_h compared to (1.3).

The notation $A \lesssim B$ abbreviates $A \leqslant CB$ for a positive, generic constant C, solely depending on the domain Ω, the polynomial degree, and the shape regularity of the triangulation $\mathcal{T}$; $A \approx B$ abbreviates $A \lesssim B \lesssim A$.

For two indices $j, k \in \mathbb{N}$, let $\delta_{jk} \in \{0, 1\}$ denote the Kronecker symbol defined by $\delta_{jk} := 1$ if and only if $j = k$. The enclosing single bars $|\bullet|$ apply context-sensitively and denote not only the modulus of real numbers, the Euclidean norm of vectors in $\mathbb{R}^2$, but also the cardinality of finite sets, the area of two-dimensional Lebesgue sets, and the length of edges.

3 Stability

This section develops a *novel* stability analysis with a penalty parameter $\sigma_{\mathrm{IP},E}$ in the bilinear form c_{IP} from (1.4). Let the discrete space V_h be equipped with the mesh-dependent norm $\|\bullet\|_h$ defined, for $v_{\mathrm{IP}} \in V_h$, by

$$\|v_{\mathrm{IP}}\|_h^2 := |||v_{\mathrm{IP}}|||_{\mathrm{pw}}^2 + c_{\mathrm{IP}}(v_{\mathrm{IP}}, v_{\mathrm{IP}})$$

with the piecewise semi-norm $|||\bullet|||_{\mathrm{pw}} := |\bullet|_{H^2(\mathcal{T})} := \|\mathrm{D}^2_{\mathrm{pw}}\bullet\|_{L^2(\Omega)}$. The parameter $\sigma_{\mathrm{IP},E}$ from (1.5) solely depends on the underlying triangulation and allows for a guaranteed discrete stability [27, Sect. 1.3.2] of the bilinear from A_h from (1.3) with respect to the mesh-dependent norm $\|\bullet\|_h$ in the following theorem.

Theorem 3.1. *Given any $a > 1$, define the penalty parameter $\sigma_{\mathrm{IP},E} > 0$ as in (1.5). For every choice of $a > 1$, the constant $\varkappa := 1 - 1/\sqrt{a} > 0$ and all discrete functions $v_{\mathrm{IP}} \in V_h$ satisfy the stability estimate*

$$\varkappa \|v_{\mathrm{IP}}\|_h^2 \leqslant A_h(v_{\mathrm{IP}}, v_{\mathrm{IP}}). \tag{3.1}$$

Four remarks are in order before the proof of the Theorem 3.1 concludes this section.

Remark 3.1 (choice of energy norm). The σ-independence of the stability constant $\varkappa$ in (3.1) justifies that $\|\bullet\|_h$ is a suitable choice for the energy norm on V_h. Any equivalent norm lead to a stable discretization as well. However, the stability constant $\varkappa$ might depend on $\sigma_{\mathrm{IP},E} \approx 1$. For instance, the stability (3.1) remains valid for all alternative weights ensuring $\widetilde{\sigma}_{\mathrm{IP},E} \geqslant \sigma_{\mathrm{IP},E}$ for all edges $E \in \mathcal{E}$. In the case of a uniform triangulation into right isosceles triangles and $a \geqslant 4/3$, Example 1.1 shows that $1 \leqslant \sigma_{\mathrm{IP},E}$ and discrete stability holds for the choice of the following norm not including $\sigma_{\mathrm{IP},E}$

$$v_{\mathrm{IP}} \mapsto \left(|||v_{\mathrm{IP}}|||_{\mathrm{pw}}^2 + \sum_{E\in\mathcal{E}} \frac{1}{h_E} \|[\nabla u_{\mathrm{IP}} \cdot \nu_E]_E\|_{L^2(E)}^2 \right)^{1/2}.$$

Remark 3.2 (other types of boundary conditions). The stability analysis generalizes to other types of boundary conditions. In the case of simply supported plates, i.e., $u = \Delta u = 0$ on $\partial\Omega$, the bilinear form A_h from (1.3) is replaced by [12, Rem. 2]:

$$\widetilde{A}_h(u_{\mathrm{IP}}, v_{\mathrm{IP}}) := a_{\mathrm{pw}}(u_{\mathrm{IP}}, v_{\mathrm{IP}}) - (\widetilde{\jmath}(u_{\mathrm{IP}}, v_{\mathrm{IP}}) + \widetilde{\jmath}(v_{\mathrm{IP}}, u_{\mathrm{IP}})) + \widetilde{c}_{\mathrm{IP}}(u_{\mathrm{IP}}, v_{\mathrm{IP}})$$

with the modified contributions

$$\widetilde{\jmath}(u_{\mathrm{IP}}, v_{\mathrm{IP}}) := \sum_{E\in\mathcal{E}(\Omega)} \int_E \langle (\mathrm{D}^2_{\mathrm{pw}} u_{\mathrm{IP}}\, \nu_E) \cdot \nu_E \rangle_E\, [\nabla v_{\mathrm{IP}} \cdot \nu_E]_E \,\mathrm{d}s$$

$$\widetilde{c}_{\mathrm{IP}}(u_{\mathrm{IP}}, v_{\mathrm{IP}}) := \sum_{E\in\mathcal{E}(\Omega)} \frac{\sigma_{\mathrm{IP},E}}{h_E} \int_E [\nabla u_{\mathrm{IP}} \cdot \nu_E]_E [\nabla v_{\mathrm{IP}} \cdot \nu_E]_E \,\mathrm{d}s.$$

The proof of Theorem 3.1 applies verbatim in this case. For Cahn–Hillard boundary conditions $\partial u/\partial n = \partial\Delta u/\partial n = 0$, the bilinear form is equal to the bilinear form in (1.3), but the space V_h consists of $P_k(\mathcal{T})$ functions vanishing at a single point in $\overline{\Omega}$ [12, Rem. 2]. Since the proof of Theorem 3.1 does not exploit the boundary conditions of $v_h \in V_h$, the estimate (3.1) holds for all piecewise polynomials $v_h \in P_k(\mathcal{T})$ for $k \geqslant 2$. Consequently, the result generalizes to Cahn–Hillard boundary conditions as well.

Remark 3.3 (variable polynomial degrees). The assumption of uniform polynomial degree can be weakened, as follows for a different polynomial degree k_T on every triangle $T \in \mathcal{T}$. For an edge $E \in \mathcal{E}$, an analogous argumentation leads to the choice of the edge-dependent parameter

$$\sigma_{\mathrm{IP},E} := \begin{cases} \dfrac{3ah_E^2}{8}\left(\dfrac{k_{T_+}(k_{T_+}-1)}{|T_+|} + \dfrac{k_{T_-}(k_{T_-}-1)}{|T_-|}\right) & \text{if } E = \partial T_+ \cap \partial T_- \in \mathcal{E}(\Omega) \\ \dfrac{3ak_{T_+}(k_{T_+}-1)h_E^2}{2|T_+|} & \text{if } E \in \mathcal{E}(T_+) \cap \mathcal{E}(\partial\Omega). \end{cases}$$

Remark 3.4 (generalization to rectangles). Theorem 3.1 can be generalized to rectangular meshes. To this end, let $\mathcal{T}$ denote a regular triangulation into closed rectangles and $Q_k(\mathcal{T})$ the space of continuous piecewise polynomials of partial degree at most $k \in \mathbb{N}$. The sharp discrete trace inequality from [34, Eq. (C.20)] for rectangles $T \in \mathcal{T}$ with edge $E \in \mathcal{E}(T)$ reads, for all $q_k \in Q_k(T)$,

$$\|q_k\|_{L^2(E)}^2 \leqslant \frac{(k+1)^2 h_E}{|T|} \|q_k\|_{L^2(T)}^2.$$

Then, for edge $E \in \mathcal{E}$ and $a > 1$, an analog of Theorem 3.1 leads to the following choice of the edge-dependent parameter

$$\sigma_{\mathrm{IP},E} := \begin{cases} a(k-1)^2 h_E^2 \left(\frac{1}{|T_+|} + \frac{1}{|T_-|} \right) & \text{if } E = \partial T_+ \cap \partial T_- \in \mathcal{E}(\Omega) \\ \frac{4a(k-1)^2 h_E^2}{|T_+|} & \text{if } E \in \mathcal{E}(T_+) \cap \mathcal{E}(\partial\Omega). \end{cases} \tag{3.2}$$

The proof of the Theorem 3.1 employs a discrete trace inequality.

Lemma 3.1 (see [45, Thm. 3]). *Every polynomial $q_k \in P_k(T)$ of degree at most $k \in \mathbb{N}_0$ satisfies*

$$\|q_k\|_{L^2(E)}^2 \leqslant \frac{(k+2)(k+1)h_E}{2|T|} \|q_k\|_{L^2(T)}^2.$$

The multiplicative constant is sharp in the sense that it cannot be replaced by any smaller constant in the absence of further conditions on the function q_k.

Proof of Theorem 3.1. The point of departure is the difference, for every $\varkappa > 0$,

$$A_h(v_{\mathrm{IP}}, v_{\mathrm{IP}}) - \varkappa \|v_{\mathrm{IP}}\|_h^2 = (1-\varkappa)|||v_{\mathrm{IP}}|||_{\mathrm{pw}}^2 - 2\mathcal{J}(v_{\mathrm{IP}}, v_{\mathrm{IP}}) + (1-\varkappa) \sum_{E\in\mathcal{E}} \frac{\sigma_{\mathrm{IP},E}}{h_E} \|[\nabla v_{\mathrm{IP}} \cdot \nu_E]_E\|_{L^2(E)}^2. \tag{3.3}$$

The first step of the proof bounds the jump term $\mathcal{J}$ with the weighted Young inequality, for every $\varepsilon > 0$,

$$2|\mathcal{J}(v_{\mathrm{IP}}, v_{\mathrm{IP}})| \leqslant \sum_{E\in\mathcal{E}} \left(\frac{\varepsilon\sigma_{\mathrm{IP},E}}{h_E} \|[\nabla v_{\mathrm{IP}} \cdot \nu_E]_E\|_{L^2(E)}^2 + \frac{h_E}{\varepsilon\sigma_{\mathrm{IP},E}} \|\langle (\mathrm{D}_{\mathrm{pw}}^2 v_{\mathrm{IP}}\, \nu_E) \cdot \nu_E \rangle_E\|_{L^2(E)}^2 \right).$$

Inserting this estimate into (3.3) confirms the lower bound

$$\begin{aligned} &(1-\varkappa)|||v_{\mathrm{IP}}|||_{\mathrm{pw}}^2 + (1-\varkappa) \sum_{E\in\mathcal{E}} \frac{\sigma_{\mathrm{IP},E}}{h_E} \|[\nabla v_{\mathrm{IP}} \cdot \nu_E]_E\|_{L^2(E)}^2 \\ &\quad - \sum_{E\in\mathcal{E}} \left(\frac{\varepsilon\sigma_{\mathrm{IP},E}}{h_E} \|[\nabla v_{\mathrm{IP}} \cdot \nu_E]_E\|_{L^2(E)}^2 + \frac{h_E}{\varepsilon\sigma_{\mathrm{IP},E}} \|\langle (\mathrm{D}_{\mathrm{pw}}^2 v_{\mathrm{IP}}\, \nu_E) \cdot \nu_E \rangle_E\|_{L^2(E)}^2 \right) \\ &\quad \leqslant A_h(v_{\mathrm{IP}}, v_{\mathrm{IP}}) - \varkappa \|v_{\mathrm{IP}}\|_h^2. \end{aligned} \tag{3.4}$$

The second step estimates the average $\langle (\mathrm{D}_{\mathrm{pw}}^2 v_{\mathrm{IP}}\, \nu_E) \cdot \nu_E \rangle_E$. Since the piecewise second derivative $\mathrm{D}_{\mathrm{pw}}^2 v_{\mathrm{IP}}$ of any function $v_{\mathrm{IP}} \in V_h$ is a polynomial of degree $k-2$ on each triangle T with edge $E \in \mathcal{E}(T)$, Lemma 3.1 implies that

$$\|\mathrm{D}_{\mathrm{pw}}^2 v_{\mathrm{IP}}|_T\|_{L^2(E)}^2 \leqslant \frac{k(k-1)h_E}{2|T|} \|\mathrm{D}_{\mathrm{pw}}^2 v_{\mathrm{IP}}|_T\|_{L^2(T)}^2. \tag{3.5}$$

In accordance with the bilinear form a_{pw} from (2.3), the L^2 norm employs the Frobenius norm $|\bullet|$ for matrix-valued functions. A Cauchy–Schwarz inequality in $\mathbb{R}^2$, the submultiplicativity of the Frobenius norm, and the normalization $|\nu_E| = 1$ prove

$$|(\mathrm{D}_{\mathrm{pw}}^2 v_{\mathrm{IP}}(x)\, \nu_E) \cdot \nu_E|^2 \leqslant |\,\mathrm{D}^2 v_{\mathrm{IP}}(x)|^2$$

for almost every $x \in E$. For an interior edge $E = \partial T_+ \cap \partial T_- \in \mathcal{E}(\Omega)$ with the neighboring triangles T_+ and T_-, this and the definition of the average from (2.2) show

$$\|\langle (\mathrm{D}_{\mathrm{pw}}^2 v_{\mathrm{IP}}\, \nu_E) \cdot \nu_E \rangle_E\|_{L^2(E)}^2 \leqslant \frac{1}{4} \|\,\mathrm{D}^2 v_{\mathrm{IP}}|_{T_+} + \mathrm{D}^2 v_{\mathrm{IP}}|_{T_-}\|_{L^2(E)}^2.$$

The combination with a triangle inequality, another weighted Young inequality with $\gamma > 0$, and (3.5) results in

$$\|\langle (\mathrm{D}_{\mathrm{pw}}^2 v_{\mathrm{IP}}\, \nu_E) \cdot \nu_E \rangle_E\|_{L^2(E)}^2 \leqslant \frac{k(k-1)h_E}{8} \left(\frac{1+\gamma}{|T_+|} \|\,\mathrm{D}^2 v_{\mathrm{IP}}\|_{L^2(T_+)}^2 + \frac{1+1/\gamma}{|T_-|} \|\,\mathrm{D}^2 v_{\mathrm{IP}}\|_{L^2(T_-)}^2 \right).$$

The multiplicative constant $\max\{(1+\gamma)/|T_+|, (1+\gamma^{-1})/|T_-|\}$ is minimal in the case of the equality $(1+\gamma)/|T_+| = (1+\gamma^{-1})/|T_-|$. Hence the optimal value $\gamma = |T_+|/|T_-|$ leads to

$$\|\langle (\mathrm{D}_{\mathrm{pw}}^2 v_{\mathrm{IP}}\, \nu_E) \cdot \nu_E \rangle_E\|_{L^2(E)}^2 \leqslant \frac{k(k-1)h_E}{8} \left(\frac{1}{|T_+|} + \frac{1}{|T_-|} \right) \|\,\mathrm{D}_{\mathrm{pw}}^2 v_{\mathrm{IP}}\|_{L^2(\omega_E)}^2.$$

The multiplication with $h_E/(\varepsilon\sigma_{\mathrm{IP},E})$ and the definition of $\sigma_{\mathrm{IP},E}$ in (1.5) show

$$\frac{h_E}{\varepsilon\sigma_{\mathrm{IP},E}}\|\langle(\mathrm{D}^2_{\mathrm{pw}}v_{\mathrm{IP}}\,\nu_E)\cdot\nu_E\rangle_E\|^2_{L^2(E)} \leqslant \frac{1}{3a\varepsilon}\|\,\mathrm{D}^2_{\mathrm{pw}}v_{\mathrm{IP}}\|^2_{L^2(\omega_E)}. \tag{3.6}$$

For a boundary edge $E \in \mathcal{E}(\partial\Omega)$ with adjacent triangle $T_+ \in \mathcal{T}$, the definition of the average, the equality $|\nu_E| = 1$, and (3.5) verify

$$\|\langle(\mathrm{D}^2_{\mathrm{pw}}v_{\mathrm{IP}}\,\nu_E)\cdot\nu_E\rangle_E\|^2_{L^2(E)} = \|(\mathrm{D}^2_{\mathrm{pw}}v_{\mathrm{IP}}\,\nu_E)\cdot\nu_E\|^2_{L^2(E)} \leqslant \|\,\mathrm{D}^2_{\mathrm{pw}}v_{\mathrm{IP}}\|^2_{L^2(E)} \leqslant \frac{k(k-1)h_E}{2|T_+|}\|\,\mathrm{D}^2v_{\mathrm{IP}}\|^2_{L^2(T_+)}.$$

Consequently, the multiplication with $h_E/(\varepsilon\sigma_{\mathrm{IP},E})$ and the definition of the penalty parameter $\sigma_{\mathrm{IP},E}$ result in the analogous estimate (3.6) for $E \in \mathcal{E}(\partial\Omega)$.

The sum of (3.6) over all edges $E \in \mathcal{E}$ and the finite overlap of the edge patches $(\omega(E) : E \in \mathcal{E})$ lead to

$$\sum_{E\in\mathcal{E}}\frac{h_E}{\varepsilon\sigma_{\mathrm{IP},E}}\|\langle(\mathrm{D}^2_{\mathrm{pw}}v_{\mathrm{IP}}\,\nu_E)\cdot\nu_E\rangle_E\|^2_{L^2(E)} \leqslant \frac{1}{a\varepsilon}|||v_{\mathrm{IP}}|||^2_{\mathrm{pw}}. \tag{3.7}$$

The third step concludes the proof with a combination of the lower bound (3.4), and the estimate (3.7) for

$$\left(1-\varkappa-\frac{1}{a\varepsilon}\right)|||v_{\mathrm{IP}}|||^2_{\mathrm{pw}} + (1-\varkappa-\varepsilon)\sum_{E\in\mathcal{E}}\frac{\sigma_{\mathrm{IP},E}}{h_E}\|[\nabla v_{\mathrm{IP}}\cdot\nu_E]_E\|^2_{L^2(E)} \leqslant A_h(v_{\mathrm{IP}},v_{\mathrm{IP}}) - \varkappa\|v_{\mathrm{IP}}\|^2_h. \tag{3.8}$$

Every choice of $0 < \varkappa \leqslant \min\{1-1/(a\varepsilon), 1-\varepsilon\}$ leads to a nonnegative lower bound in (3.8), and so proves the claim (3.1); $\varepsilon = 1/\sqrt{a}$ results in $\varkappa = 1-1/\sqrt{a}$. □

4 Numerical experiments

This section investigates an implementation of the C0IP method from (1.2) using an unpublished in-house software package AFEM2 in MATLAB version 9.9.0.1718557 (R2020b) Update 6. The investigation considers four computational benchmark examples with domains displayed in Fig. 1. If not mentioned otherwise, the numerical experiments employ $a = 4$.

4.1 Adaptive mesh refinement

The experiments include a comparison of uniform and adaptive mesh refinement driven by the a posteriori error estimator $\eta^2(\mathcal{T}) := \sum_{T\in\mathcal{T}}\eta^2(T)$ with, for all $T \in \mathcal{T}$,

$$\begin{aligned}\eta^2(T) := \|h_T^2(f-\Delta^2u_{\mathrm{IP}})\|^2_{L^2(T)} &+ \sum_{E\in\mathcal{E}(T)}\frac{\sigma^2_{\mathrm{IP},E}}{h_E}\|[\nabla u_{\mathrm{IP}}]_E\cdot\nu_E\|^2_{L^2(E)}\\ &+ \sum_{E\in\mathcal{E}(T)\cap\mathcal{E}(\Omega)}h_E\|[(\mathrm{D}^2_{\mathrm{pw}}u_{\mathrm{IP}}\,\nu_E)\cdot\nu_E]_E\|^2_{L^2(E)} + \sum_{E\in\mathcal{E}(T)\cap\mathcal{E}(\Omega)}h_E^3\|[\partial(\Delta u_{\mathrm{IP}})/\partial\nu_E]_E\|^2_{L^2(E)}\end{aligned} \tag{4.1}$$

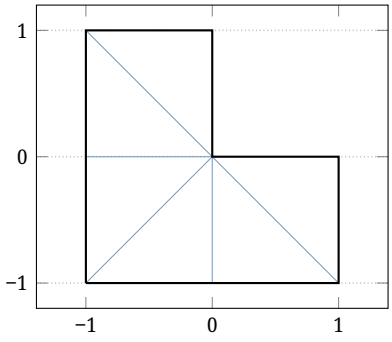
from [12, Sect. 7.1]. The software employs the Dörfler marking strategy [29] and newest-vertex bisection according to [43] in the adaptive Algorithm 1 to compute a sequence of nested triangulations $(\mathcal{T}_\ell)_{\ell\in\mathbb{N}_0}$. The marking strategy with minimal cardinality of the set $\mathcal{M}_\ell$ of marked triangles can be realized in linear computational complexity [40].

Another main aspect of the numerical investigation concerns the influence of the parameter $a > 1$ in (1.5) to the stability of the scheme. To this end, the principal eigenvalue $\lambda_{1,\ell}$ to the following discrete eigenvalue problem is computed: Seek $\Phi_{\mathrm{IP}} \in V_\ell := S^k_0(\mathcal{T}_\ell)$ with $\Phi_{\mathrm{IP}} \neq 0$ and $\mu \in \mathbb{R}$ such that

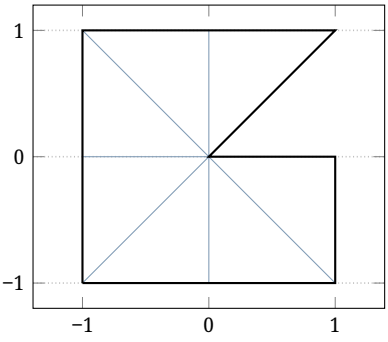
$$A_h(\Phi_{\mathrm{IP}},v_{\mathrm{IP}}) = \mu(a_{\mathrm{pw}}(\Phi_{\mathrm{IP}},v_{\mathrm{IP}}) + c_{\mathrm{IP}}(\Phi_{\mathrm{IP}},v_{\mathrm{IP}})) \quad \forall v_{\mathrm{IP}} \in V_\ell. \tag{4.2}$$

The Rayleigh–Ritz principle shows that $\varkappa = 1-1/\sqrt{a}$ from Theorem 3.1 provides a lower bound for the principal eigenvalue $\lambda_{1,\ell}$. Given a basis $\varphi_0,\dots,\varphi_J$ of V_ℓ, the practical realization of (4.2) employs the matrices $B(\mathcal{T}), N(\mathcal{T}) \in \mathbb{R}^{J\times J}$ given, for $j,k = 1,\dots,J$, by

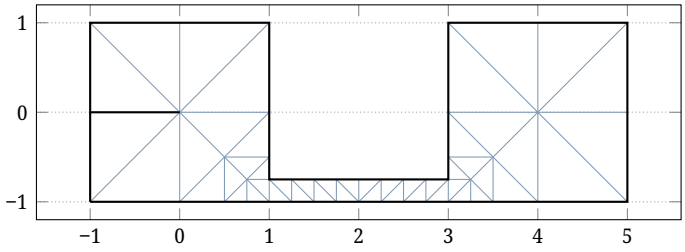
$$B_{jk}(\mathcal{T}) := A_h(\varphi_j,\varphi_k), \qquad N_{jk}(\mathcal{T}) := a_{\mathrm{pw}}(\varphi_j,\varphi_k) + c_{\mathrm{IP}}(\varphi_j,\varphi_k). \tag{4.3}$$



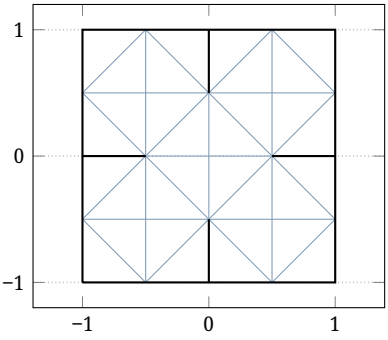
(a) L-shaped domain



(b) 1/8 cusp domain



(c) Dumbbell-slit domain



(d) Four-slit domain

Fig. 1: Visualization of the initial triangulations $\mathcal{T}_0$ for the four benchmark problems.

Algorithm 1 Adaptive C^0 interior penalty method

Input: regular triangulation $\mathcal{T}_0$ and bulk parameter $0 < \vartheta \leqslant 1$.

for $\ell = 0, 1, 2, \dots$ **do**

Solve (1.2) with respect to triangulation $\mathcal{T}_\ell$ for solution $u_\ell \in S^k_0(\mathcal{T}_\ell)$.

Compute refinement indicator $\eta^2(\mathcal{T}_\ell, T)$ from (4.1) for all $T \in \mathcal{T}_\ell$.

Mark a minimal subset $\mathcal{M}_\ell \subseteq \mathcal{T}_\ell$ by the Dörfler criterion

$$\vartheta\, \eta^2(\mathcal{T}_\ell) \leqslant \sum_{T\in\mathcal{M}_\ell} \eta^2(\mathcal{T}_\ell, T).$$

Refine $\mathcal{T}_\ell$ to $\mathcal{T}_{\ell+1}$ by newest-vertex bisection such that $\mathcal{M}_\ell \subseteq \mathcal{T}_\ell \setminus \mathcal{T}_{\ell+1}$.

end for

Output: sequence of triangulations $\mathcal{T}_\ell$ with u_ℓ and $\eta(\mathcal{T}_\ell)$ for $\ell \in \mathbb{N}_0$.

4.2 Singular solution on L-shaped domain

This benchmark considers the L-shaped domain $\Omega = (-1, 1)^2 \setminus [0, 1)^2$ with interior angle $\omega = 3\pi/2$ at the reentrant corner. This determines the noncharacteristic solution $\alpha := 0.5444837$ to $\sin^2(\alpha\omega) = \alpha^2 \sin^2(\omega)$ in

$$\begin{aligned} g_{\alpha,\omega}(\varphi) = &\left(\frac{\sin((\alpha-1)\omega)}{\alpha-1} - \frac{\sin((\alpha+1)\omega)}{\alpha+1}\right)\big(\cos((\alpha-1)\varphi) - \cos((\alpha+1)\varphi)\big) \\ &- \left(\frac{\sin((\alpha-1)\varphi)}{\alpha-1} - \frac{\sin((\alpha+1)\varphi)}{\alpha-1}\right)\big(\cos((\alpha-1)\omega) - \cos((\alpha+1)\omega)\big). \end{aligned} \tag{4.4}$$

The exact singular solution in polar coordinates from [32, p. 107] reads

$$u(r, \varphi) = (1 - r^2\cos^2\varphi)^2(1 - r^2\sin^2\varphi)^2 r^{1+\alpha} g_{\alpha,\omega}\left(\varphi - \frac{\pi}{2}\right). \tag{4.5}$$

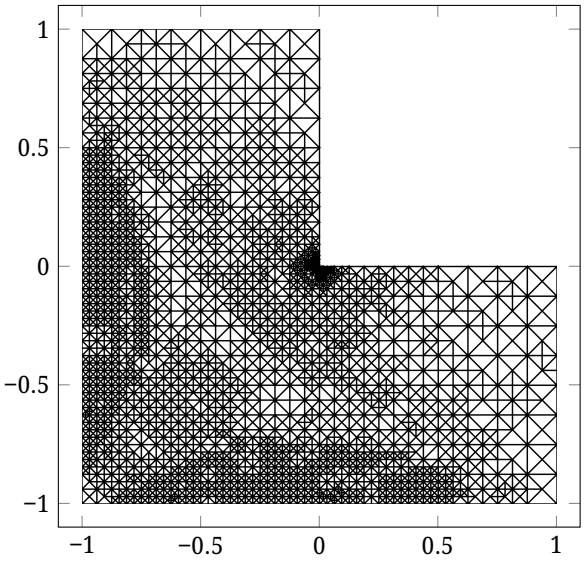
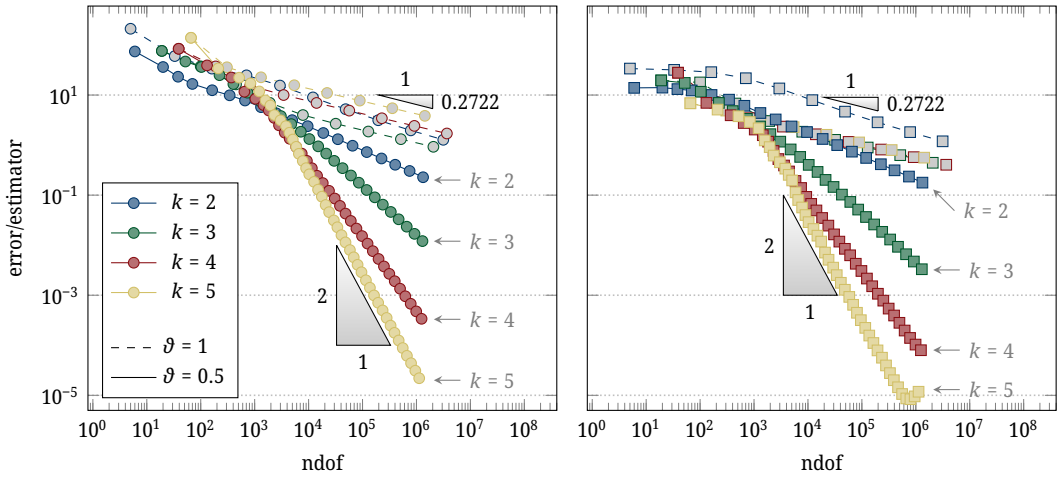


Fig. 2: Adaptively refined mesh ($\vartheta = 0.5$) with polynomial degree $k = 2$ and 10406 degrees of freedom for the L-shaped domain in Section 4.2.



(a) Error estimator $\eta(\mathcal{T}_\ell)$ **(b)** Exact error $\|u - u_h\|_h$

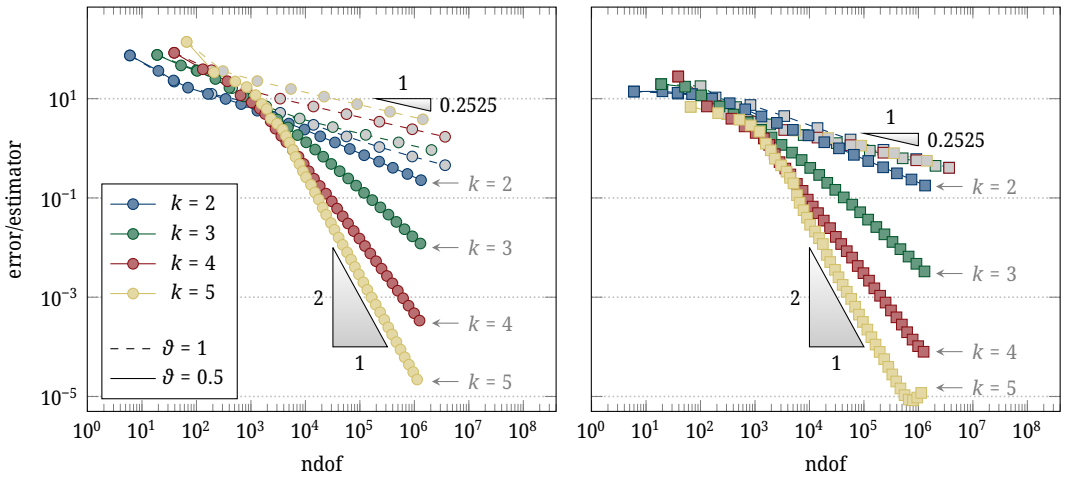
Fig. 3: Convergence history plots for the L-shaped domain in Section 4.2.

The right-hand side $f := \Delta^2 u$ is computed accordingly. The reduced regularity of the exact solution u leads to the empirical convergence rate $\alpha/2$ for uniform mesh refinement displayed by dashed lines in Fig. 3. The adaptive refinement strategy results in local mesh refining at the reentrant corner in Fig. 2. For all polynomial degrees $k = 2, \dots, 5$, the adaptive algorithm recovers the optimal convergence rates with respect to the number of degrees of freedom (ndof) as displayed by the solid lines in Fig. 3.

4.3 Singular solution on 1/8 cusp domain

This benchmark problem investigates the 1/8 cusp domain $\Omega := (-1, 1)^2 \setminus \operatorname{conv}\{(0, 0), (1, -1), (1, 0)\}$ with interior angle $\omega = 7\pi/4$ as depicted in Fig. 1b. The right-hand side $f := \Delta^2 u$ is given by the exact singular solution in polar coordinates

$$u(r, \varphi) = (1 - r^2 \cos^2 \varphi)^2 (1 - r^2 \sin^2 \varphi)^2 r^{1+\alpha} g_{\alpha,\omega}\left(\varphi - \frac{\pi}{4}\right) \tag{4.6}$$



(a) Error estimator $\eta(\mathcal{T}_\ell)$ **(b)** Exact error $\|u - u_h\|_h$

Fig. 4: Convergence history plots for the 1/8 cusp domain in Section 4.3.

analogously to (4.5) with $g_{\alpha,\omega}$ from (4.4) for the parameter $\alpha = 0.50500969$. The singularity causes an empirical convergence rate $\alpha/2$ for uniform mesh refinement in Fig. 4, while the adaptive algorithm recovers the optimal rates for the considered polynomial degrees. In the case $k = 5$, the highly adapted meshes lead to nearly singular matrices close to the number of 10^6 degrees of freedom. Undisplayed triangulation plots confirm the increased adaptive refinement towards the reentrant corner.

4.4 Dumbbell-slit domain

The benchmark problem considers the uniform force $f \equiv 1$ on the dumbbell-slit domain

$$\Omega = \big((-1,1)\times(-1,5)\setminus[1,3]\times[-0.75,1)\big)\setminus(-1,0]\times\{0\}$$

depicted in Fig. 1c. Although Ω is no Lipschitz domain, Figures 5 confirm the optimal convergence rates with adaptive mesh refinement. Undisplayed plots of the adaptive triangulations show an increased refinement towards the slit as well as the two reentrant corners at $(1, -0.75)$ and $(3, -0.75)$.

4.5 Four-slit domain

Let

$$\Omega = \big((-1,1)^2\setminus\big((-1,-0.5]\cup[0.5,1)\big)\times\{0\}\big)\setminus\{0\}\times\big((-1,-0.5]\cup[0.5,1)\big)$$

denote the square domain with one slit at each edge as depicted in Fig. 1d. This benchmark also considers the uniform load $f \equiv 1$. Similarly to the previous benchmarks, optimal convergence rates can be observed in the case the adaptive mesh refinement for the polynomial degrees $k = 2, \dots, 5$ in Fig. 6. Undisplayed triangulation plots indicate that the adaptive algorithm detects the singularities of the unknown exact solution at tips of the four slits.

4.6 Investigation of parameter α

Given the triangulation $\mathcal{T}_\ell$ on the level $\ell \in \mathbb{N}_0$ from the adaptive Algorithm 1, the principle eigenvalue $\lambda_{1,\ell}$ of the discrete eigenvalue problem (4.2) with respect to $\mathcal{T}_\ell$ is the stability constant in (3.1). The computation applies MATLAB's `eigs` function to the eigenvalue problem of the matrices $B(\mathcal{T}_\ell)$ and $N(\mathcal{T}_\ell)$ from (4.3).

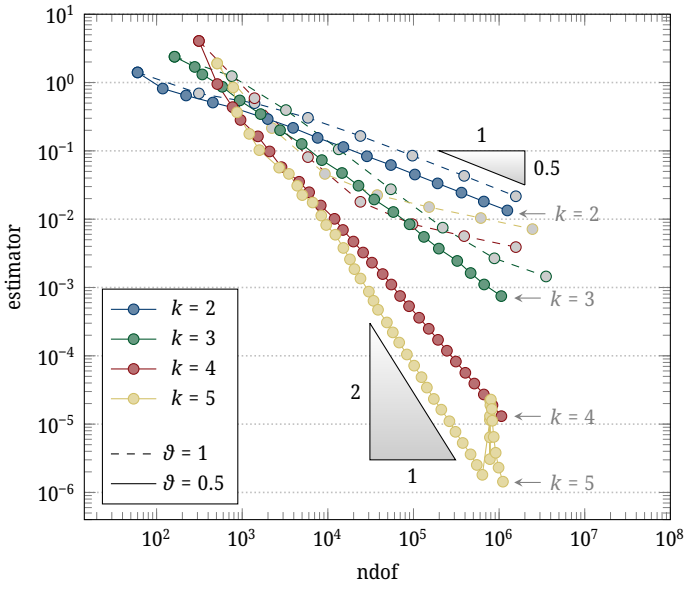


Fig. 5: Convergence history plots for the dumbbell domain in Section 4.4.

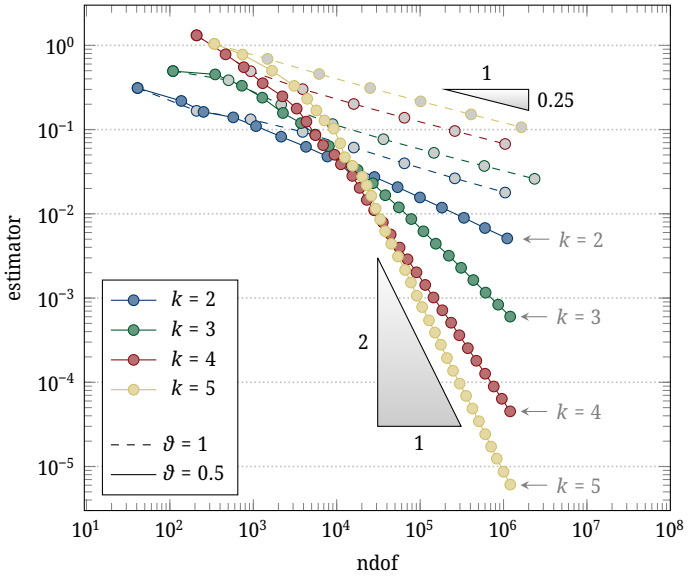
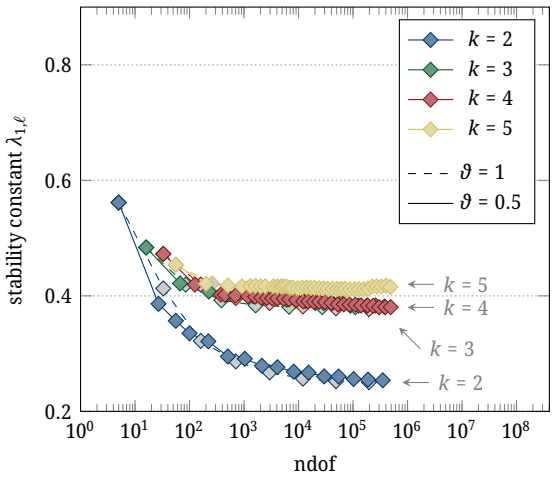


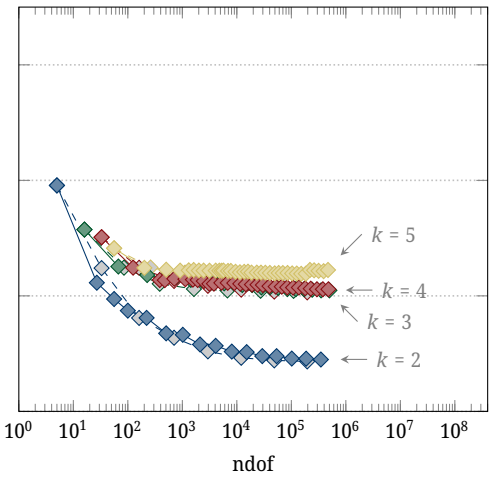
Fig. 6: Convergence history plots for the fourslit domain in Section 4.5.

Figure 7 displays $\lambda_{1,\ell}$ in dependence of the number of degrees of freedom for different choices of $a > 1$ using adaptive and uniform mesh refinement on the L-shaped domain in Section 4.2; a large parameter a results in a larger stability constant tending towards 1. The guaranteed lower bound $\varkappa = 1 - 1/\sqrt{a}$ from Theorem 3.1 is always fulfilled. It is remarkable that even the unjustified choice $a = 1$ leads to a seemingly stable discretization with stability constants between 0.2 and 0.5 depicted in Fig. 7a. Figure 9a displays the relation of the parameter a and the stability constant $\lambda_{1,\ell}$ in more detail. The computed values for $\lambda_{1,\ell} < 1$ are slightly larger than the guaranteed lower bounds. This indicates little over-stabilization of the automated choice of $\sigma_{\mathrm{IP},E}$. Undisplayed numerical experiments do not reveal any significant difference between the computed stability constants of the four domains in Fig. 1.

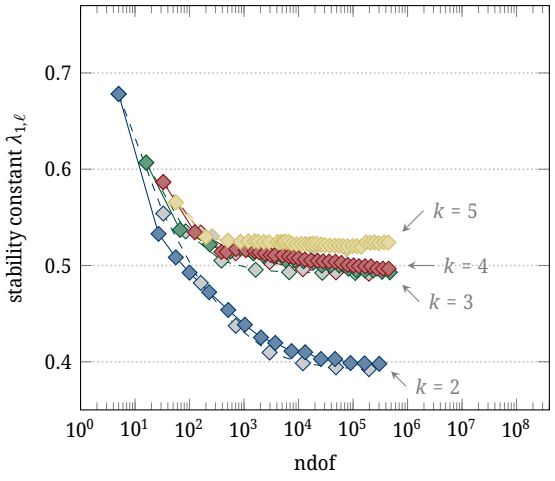
An interesting empirical observation is that adaptive mesh refinement and higher-order polynomials even lead to slightly more stable discretizations. An over-stabilization as in the theoretical analysis from [9] is *not* necessary in the computational benchmarks. This supports the advantage of the automated choice of the penalization with $\sigma_{\mathrm{IP},E}$ for guaranteed stability.



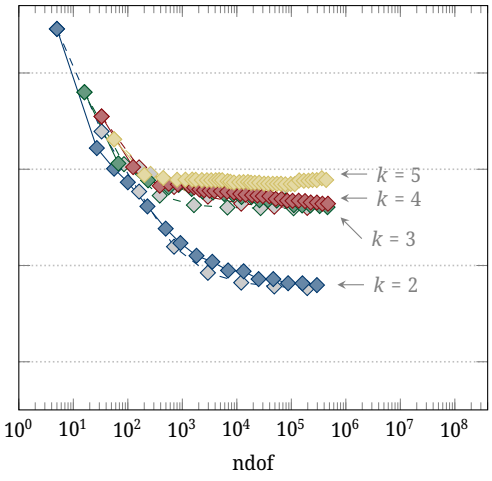
(a) $a = 1.0$ (lower bound $\varkappa = 0$)



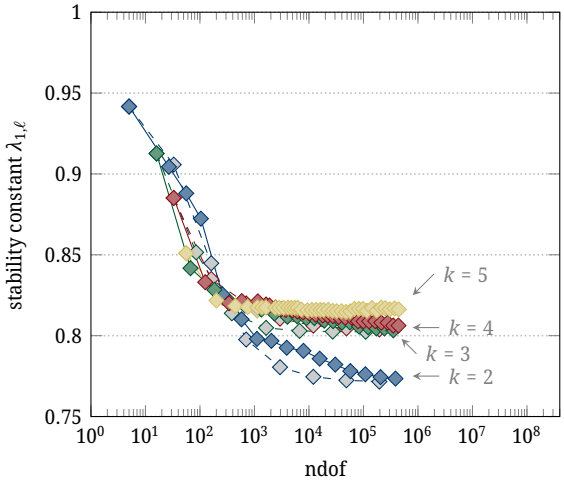
(b) $a = 1.1$ (lower bound $\varkappa = 0.0465$)



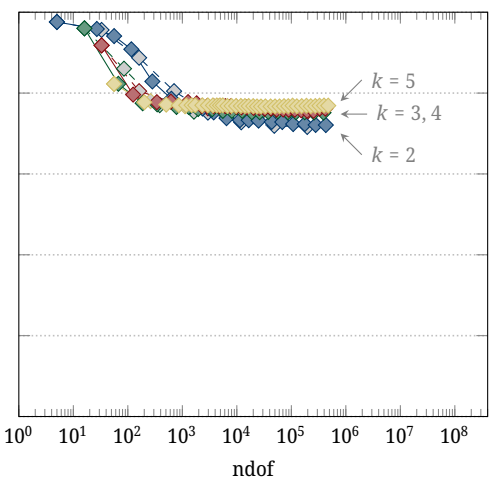
(c) $a = 1.5$ (lower bound $\varkappa = 0.1835$)



(d) $a = 2.0$ (lower bound $\varkappa = 0.2929$)

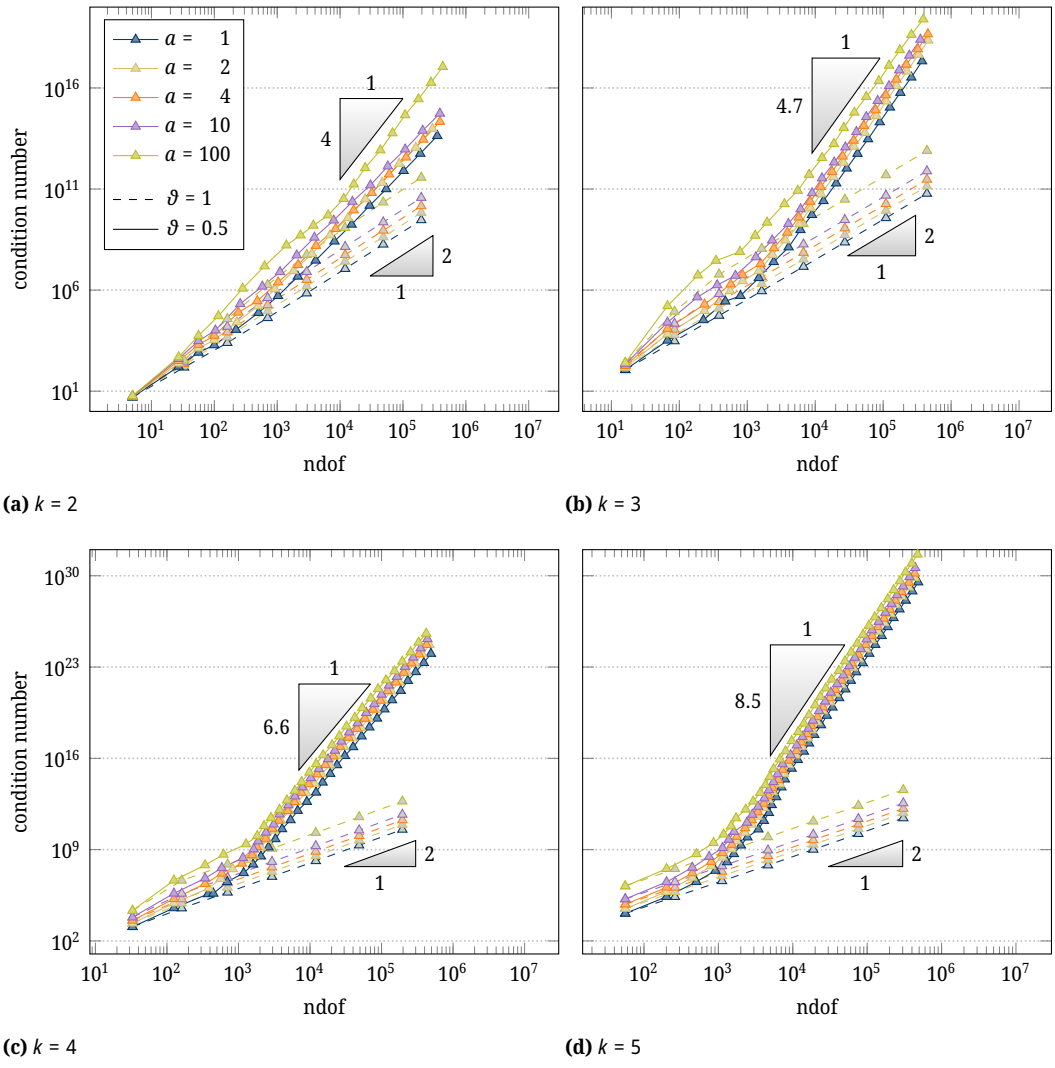


(e) $a = 10.0$ (lower bound $\varkappa = 0.6838$)



(f) $a = 100.0$ (lower bound $\varkappa = 0.9$)

Fig. 7: Plot of stability constant $\lambda_{1,\ell}$ with various selections of parameter a for the L-shaped domain in Subsection 4.2.



(a) $k = 2$ **(b)** $k = 3$

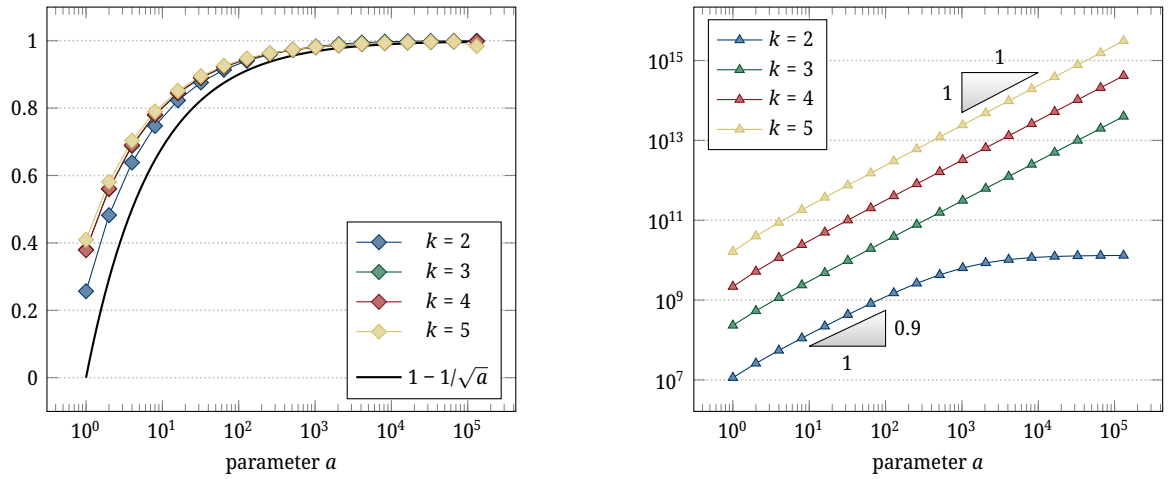
(c) $k = 4$ **(d)** $k = 5$

Fig. 8: Plot of condition number $\mathrm{cond}_1(B(\mathcal{T}_\ell))$ of the system matrix $B(\mathcal{T}_\ell)$ from (4.3) with various selections of a for the L-shaped domain in Subsection 4.2.

The condition number $\mathrm{cond}_1(B(\mathcal{T}_\ell))$ with respect to the 1-norm of the system matrix $B(\mathcal{T}_\ell)$ from (4.3) depends on a as displayed in Fig. 8. The presented lower bounds for $\mathrm{cond}_1(B(\mathcal{T}_\ell))$ are computed using MATLAB's `condest` function. Naturally, the number of degrees of freedom has the biggest influence and the rate of the increase depends on the polynomial degree k. While the importance of a seems negligible, the difference between the condition number for $a = 1$ and $a = 100$ is still about two orders of magnitude. Figure 9b illustrates this linear relation between the parameter a and the condition number on a fixed uniform mesh of the L-shaped domain. This suggests that a small a is advantageous.

4.7 Conclusions

The numerical experiments validate the theoretically established stability result with the proposed penalty parameter from (1.5) for polynomial degrees $k = 2, 3, 4, 5$. Figures 3–6 exhibit rate-optimal convergence of the adaptive C^0 interior penalty method.



(a) Stability constant $\lambda_{1,\ell} > 1 - a^{-1/2}$

(b) Condition number $\mathrm{cond}_1(B(\mathcal{T}_\ell))$

Fig. 9: Plots of a-dependence of quantities on a uniform mesh with 6 144 triangles of the L-shaped domain in Subsection 4.2.

A detailed investigation of the influence of the parameter $a > 1$ recommends the choice of a as small as possible in order to avoid large condition numbers of the linear system of equations. We suggest $a = 2$ with $\varkappa = 1 - 1/\sqrt{2} \approx 0.293$.

Funding: This paper has been supported by SPARC project (ID 235) 'The Mathematics and Computation of Plates' and by the Austrian Science Fund (FWF) standalone project P33216 'Computational nonlinear PDEs'.

Acknowledgment: It is our pleasure to thank Zhaonan Dong (of Inria, Paris) for fruitful discussions on the discrete trace inequality.

References

[1] S. Agmon, *Lectures on Elliptic Boundary Value Problems*, AMS Chelsea Publishing, Providence, RI, 2010.

[2] M. S. Alnæs, A. Logg, K. B. Ølgaard, M. E. Rognes, and G. N. Wells, Unified form language: a domain-specific language for weak formulations and partial differential equations, *ACM Trans. Math. Software* **40** (2014), No. 2, Art. 9, 37.

[3] M. Alnæs, J. Blechta, J. Hake, A. Johansson, B. Kehlet, A. Logg, C. Richardson, J. Ring, M. E. Rognes, and G. N. Wells, The FEniCS project, Version 1.5, *Archive of Numerical Software* **3** (2015), 9–23.

[4] J. H. Argyris, I. Fried, and D. W. Scharpf, The TUBA family of plate elements for the matrix displacement method, *Aeronautical J.* **72** (1968), 701–709.

[5] D. Arndt, W. Bangerth, M. Feder, M. Fehling, R. Gassmöller, T. Heister, L. Heltai, M. Kronbichler, M. Maier, P. Munch, J.-P. Pelteret, S. Sticko, B. Turcksin, and D. Wells, The `deal.II` Library, Version 9.4, *J. Numer. Math.* **30** (2022), No. 3, 231–246.

[6] G. A. Baker, Finite element methods for elliptic equations using nonconforming elements, *Math. Comp.* **31** (1977), No. 137, 45–59.

[7] P. Bastian, M. Blatt, A. Dedner, N.-A. Dreier, C. Engwer, R. Fritz, C. Grüninger, D. Kempf, R. Klöfkorn, M. Ohlberger, and O. Sander, The Dune framework: basic concepts and recent developments, *Comput. Math. Appl.* **81** (2021), 75–112.

[8] H. Blum and R. Rannacher, On the boundary value problem of the biharmonic operator on domains with angular corners, *Math. Methods Appl. Sci.* **2** (1980), No. 4, 556–581.

[9] A. Bonito and R. H. Nochetto, Quasi-optimal convergence rate of an adaptive discontinuous Galerkin method, *SIAM J. Numer. Anal.* **48** (2010), No. 2, 734–771.

[10] D. Braess, *Finite Elements*, 3rd ed., Cambridge University Press, Cambridge, 2007.

[11] S. C. Brenner, J. Cui, T. Gudi, and L.-Y. Sung, Multigrid algorithms for symmetric discontinuous Galerkin methods on graded meshes, *Numer. Math.* **119** (2011), No. 1, 21–47.

[12] S. C. Brenner, C^0 interior penalty methods, In: *Frontiers in Numerical Analysis — Durham 2010*, Lect. Notes Comput. Sci. Vol. 85, Springer, Heidelberg, 2012, pp. 79–147.

[13] S. C. Brenner, C. B. Davis, and L.-Y. Sung, Additive Schwarz preconditioners for the obstacle problem of clamped Kirchhoff plates, *Electron. Trans. Numer. Anal.* **49** (2018), 274–290.
[14] S. C. Brenner, T. Gudi, and L.-Y. Sung, An a posteriori error estimator for a quadratic C^0-interior penalty method for the biharmonic problem, *IMA J. Numer. Anal.* **30** (2010), No. 3, 777–798.
[15] S. C. Brenner and L. R. Scott, *The Mathematical Theory of Finite Element Methods*, 3rd ed., Texts in Applied Mathematics, Vol. 15, Springer, New York, 2008.
[16] S. C. Brenner and L.-Y. Sung, C^0 interior penalty methods for fourth order elliptic boundary value problems on polygonal domains, *J. Sci. Comput.* **22-23** (2005), 83–118.
[17] S. C. Brenner and L.-Y. Sung, Multigrid algorithms for C^0 interior penalty methods, *SIAM J. Numer. Anal.* **44** (2006), No. 1, 199–223.
[18] S. C. Brenner, L.-Y. Sung, and K. Wang, Additive Schwarz preconditioners for C^0 interior penalty methods for the obstacle problem of clamped Kirchhoff plates, *Numer. Methods Partial Differential Equations* **38** (2022), No. 1, 102–117.
[19] S. C. Brenner, L.-Y. Sung, H. Zhang, and Y. Zhang, A Morley finite element method for the displacement obstacle problem of clamped Kirchhoff plates, *J. Comput. Appl. Math.* **254** (2013), 31–42.
[20] S. C. Brenner and K. Wang, Two-level additive Schwarz preconditioners for C^0 interior penalty methods, *Numer. Math.* **102** (2005), No. 2, 231–255.
[21] S. C. Brenner and J. Zhao, Convergence of multigrid algorithms for interior penalty methods, *Appl. Numer. Anal. Comput. Math.* **2** (2005), No. 1, 3–18.
[22] P. Bringmann, C. Carstensen, and J. Streitberger, Local parameter selection in the C^0 interior penalty method for the biharmonic equation, *arXiv:2209.05221*, 2023.
[23] C. Carstensen and J. Hu, Hierarchical Argyris finite element method for adaptive and multigrid algorithms, *Comput. Methods Appl. Math.* **21** (2021), No. 3, 529–556.
[24] C. Carstensen, G. Mallik, and N. Nataraj, A priori and a posteriori error control of discontinuous Galerkin finite element methods for the von Kármán equations, *IMA J. Numer. Anal.* **39** (2019), No. 1, 167–200.
[25] C. Carstensen and N. Nataraj, Adaptive Morley FEM for the von Kármán equations with optimal convergence rates, *SIAM J. Numer. Anal.* **59** (2021), No. 2, 696–719.
[26] C. Carstensen and N. Nataraj, A priori and a posteriori error analysis of the Crouzeix–Raviart and Morley FEM with original and modified right-hand sides, *Comput. Methods Appl. Math.* **21** (2021), No. 2, 289–315.
[27] D. A. Di Pietro and A. Ern, *Mathematical Aspects of Discontinuous Galerkin Methods*, Vol. 69, Springer, 2011.
[28] V. Dominguez and F.-J. Sayas, Algorithm 884: A Simple Matlab implementation of the Argyris element, *ACM Trans. Math. Softw.* **35** (2008), No. 2, 1–11.
[29] W. Dörfler, A convergent adaptive algorithm for Poisson's equation, *SIAM J. Numer. Anal.* **33** (1996), No. 3, 1106–1124.
[30] D. Gallistl, Morley finite element method for the eigenvalues of the biharmonic operator, *IMA J. Numer. Anal.* **35** (2015), No. 4, 1779–1811.
[31] D. Gilbarg and N. S. Trudinger, *Elliptic Partial Differential Equations of Second Order*, Classics in Mathematics, Springer-Verlag, Berlin, 2001.
[32] P. Grisvard, *Singularities in Boundary Value Problems*, Recherches en Mathématiques Appliquées, Vol. 22, Masson, Paris; Springer-Verlag, Berlin, 1992.
[33] B. Gräßle, Optimal multilevel adaptive FEM for the Argyris element, *Comput. Methods Appl. Mech. Engrg.* **399** (2022), 115352.
[34] K. Hillewaert, *Development of the discontinuous Galerkin method for high-resultion, large scale CFD and acoustics in industrial geometries*, Ph.D. thesis, Université Catholique de Louvain, 2013.
[35] R. C. Kirby, A general approach to transforming finite elements, *SMAI J. Comput. Math.* **4** (2018), 197–224.
[36] R. C. Kirby and L. Mitchell, Code generation for generally mapped finite elements, *ACM Trans. Math. Softw.* **45** (2019), No. 4, 1–23.
[37] J. Nečas, *Direct Methods in the Theory of Elliptic Equations*, Springer Monographs in Mathematics, Springer, Heidelberg, 2012.
[38] K. B. Ølgaard, A. Logg, and G. N. Wells, Automated code generation for discontinuous Galerkin methods, *SIAM J. Sci. Comput.* **31** (2008/09), No. 2, 849–864.
[39] K. B. Ølgaard, A. Logg, and G. N. Wells, Automated code generation for discontinuous Galerkin methods, *SIAM J. Sci. Comput.* **31** (2009), No. 2, 849–864.
[40] C.-M. Pfeiler and D. Praetorius, Dörfler marking with minimal cardinality is a linear complexity problem, *Math. Comp.* **89** (2020), No. 326, 2735–2752.
[41] F. Rathgeber, D. A. Ham, L. Mitchell, M. Lange, F. Luporini, A. T. T. McRae, G.-T. Bercea, G. R. Markall, and P. H. J. Kelly, Firedrake: automating the finite element method by composing abstractions, *ACM Trans. Math. Software* **43** (2016), No. 3, 1–27.
[42] J. Schöberl, C++11 implementation of finite elements in NGSolve, *Institute for Analysis and Scientific Computing, TU Wien*, **30** (2014).
[43] R. Stevenson, The completion of locally refined simplicial partitions created by bisection, *Math. Comp.* **77** (2008), No. 261, 227–241.
[44] E. Süli and I. Mozolevski, *hp*-version interior penalty DGFEMs for the biharmonic equation, *Comput. Math. Appl.* **196** (2007), No. 13-16, 1851–1863.
[45] T. Warburton and J. S. Hesthaven, On the constants in *hp*-finite element trace inverse inequalities, *Comput. Methods Appl. Mech. Engrg.* **192** (2003), No. 25, 2765–2773.

A Remarks on the numerical realization

The following three subsections illustrate the numerical realization of the C0IP method with the proposed local parameter selection in established software packages such as FEniCS (using the unified form language), deal.II, and NGSolve.

A.1 Realization using the unified form language (UFL)

The unified form language [2] provides a standardized way of implementing variational formulations for PDEs. It supports discontinuous Galerkin discretizations such as the C0IP method [38].

The first step defines the shape of the element domain and several geometric quantities required for the bilinear form A_h:

```
cell = triangle
n = FacetNormal(cell)
h = FacetArea(cell)
vol = CellVolume(cell)
```

Given a polynomial degree $k \in \mathbb{N}$ with $k \geqslant 2$, these variables allow to define the penalty parameter $\sigma_{\mathrm{IP},E}$ according to the formula (1.5):

```
a = 4.0
sigma = 3.0 * a * k * (k - 1) / 8.0 * h("+")**2 * avg(1 / vol)
sigma_boundary = 3.0 * a * k * (k - 1) * h**2 / vol
```

The following lines specify the Lagrange finite element and the trial and test functions:

```
element = FiniteElement("Lagrange", cell, k)
u = TrialFunction(element)
v = TestFunction(element)
```

Abbreviate the Hessian matrix of finite element functions via

```
def hess(v):
    return grad(grad(v))
```

The realization of the bilinear form A_h from (1.3) reads

```
A = inner(hess(u), hess(v)) * dx \
    - inner(avg(dot(hess(u) * n, n)), jump(grad(v), n)) * dS \
    - inner(dot(grad(u), n), dot(hess(v) * n, n)) * ds \
    - inner(jump(grad(u), n), avg(dot(hess(v) * n, n))) * dS \
    - inner(dot(hess(u) * n, n), dot(grad(v), n)) * ds \
    + sigma / h("+") * inner(jump(grad(u), n), jump(grad(v), n)) * dS \
    + sigma_boundary / h * inner(dot(grad(u), n), dot(grad(v), n)) * ds
```

Finally, the right-hand side may be defined as follows:

```
f = Coefficient(element)
F = f * v * dx
```

Many software packages employ the UFL such as DUNE [7], FEniCS [3], and Firedrake [41]. With slight adaptions of the above UFL code, the C0IP method with automatic penalty selection can be realized within these three software frameworks for arbitrary polynomial degree.

A.2 Realization in deal.II

The C++ software package deal.II [5] (version 9.4) provides a finite element implementation for rectangular grids. An example implementation of the C0IP method in the file `examples/step-47.cc` employs the following heuristic choice of the penalty parameter (adapted to the notation in this paper) for an edge $E \in \mathcal{E}$ with $d_\pm := 2|T_\pm|/h_E$:

$$\frac{\widetilde{\sigma}_{\mathrm{IP},E}}{h_E} = \begin{cases} k(k+1)/\min\{d_+, d_-\} & \text{if } E = \partial T_+ \cap \partial T_- \in \mathcal{E}(\Omega) \\ k(k+1)/d_+ & \text{if } E \in \mathcal{E}(T_+) \cap \mathcal{E}(\partial\Omega). \end{cases}$$

This leads to a slight over-penalization compared to the parameter suggested in (3.2):

$$\frac{\sigma_{\mathrm{IP},E}}{h_E} = \begin{cases} a(k-1)^2\left(\frac{1}{d_+} + \frac{1}{d_-}\right) & \text{if } E = \partial T_+ \cap \partial T_- \in \mathcal{E}(\Omega) \\ \frac{4a(k-1)^2}{d_+} & \text{if } E \in \mathcal{E}(T_+) \cap \mathcal{E}(\partial\Omega). \end{cases}$$

The application of the automatic parameter selection with guaranteed stability solely requires the replacement of the lines 512–518 in the file `step-47.cc` by

```
const double a = 4.0;
const double sigma_over_h =
  2 * a * pow(k - 1, 2.0) * (
      1 / cell->extent_in_direction(
          GeometryInfo<dim>::unit_normal_direction[f]) +
      1 / ncell->extent_in_direction(
          GeometryInfo<dim>::unit_normal_direction[nf]));
```

for the parameter on the interior edges and the lines 622–625 by

```
const double a = 2.0;
const double sigma_over_h =
    2 * a * pow(k - 1, 2) /
    cell->extent_in_direction(
    GeometryInfo<dim>::unit_normal_direction[face_no]);
```

for the parameter on the boundary edges.

A.3 Realization in Netgen/NGSolve

The C++ software package NGSolve [42] with an easy-to-use python interface provides an implementation of a hybridized C0IP method. Straight-forward modifications of the existing code enable the C0IP method presented in this paper. Given an object `mesh` of the `Mesh` class representing a regular triangulation into simplices, the Lagrange finite element of order $k \in \mathbb{N}$ with $2 \leqslant k$ can be defined as follows:

```
k = 2
fes = H1(mesh, order=k, dirichlet="left|right|bottom|top", dgjumps=True)
```

This allows to define variables for trial and test functions:

```
u = fes.TrialFunction()
v = fes.TestFunction()
```

For an edge $E \in \mathcal{E}$ with adjacent triangles $T_\pm \in \mathcal{T}$, recall the notation $d_\pm := 2|T_\pm|/h_E$ for the altitude of $T_\pm$. This quantity and the normal vector ν_E can be defined by special coefficient functions in NGsolve:

```
d = specialcf.mesh_size
n = specialcf.normal(2)
```

Thus, the automatic choice of the penalty parameter $\sigma_{\mathrm{IP},E}$ reads, for interior and boundary edges,

```
a = 4
sigma_over_h = 3 * a * k * (k - 1) / 4 * (1/d + 1/ d.Other())
sigma_over_h_bdr = 3 * a * k * (k - 1) / 2 * 1/d
```

Abbreviating the Hessian of a function v

```
def hesse(v):
    return v.Operator("hesse")
```

allows for the following variables for the average of the binormal part of the Hessian in the jump term $\mathcal{J}$ from (2.3):

```
mean_d2udn2 = 0.5*InnerProduct(n, (hesse(u)+hesse(u.Other()))*n)
mean_d2vdn2 = 0.5*InnerProduct(n, (hesse(u)+hesse(u.Other()))*n)
```

This and the definition of the normal jump

```
def jumpdn(v):
    return n*(grad(v)-grad(v.Other()))
```

lead to the bilinear form A_h from (1.3) via

```
A = BilinearForm(fes)
A += SymbolicBFI(InnerProduct (hesse(u), hesse(v)) )
A += SymbolicBFI(-mean_d2udn2 * jumpdn(v), skeleton=True)
A += SymbolicBFI(-mean_d2vdn2 * jumpdn(u), skeleton=True)
A += SymbolicBFI(sigma_over_h * jumpdn(u) * jumpdn(v), VOL, skeleton=True)
A += SymbolicBFI(sigma_over_h_bdr * jumpdn(u) * jumpdn(v), BND, skeleton=True)
```