Discontinuous Petrov-Galerkin Methods for the Convection-Diffusion Problem

Bachelor Thesis

written by
Philippe Peter

supervised by
Prof. Dr. Ralf Hiptmair
Seminar for Applied Mathematics
ETH Zurich

Abstract

In this thesis the primal and ultraweak DPG formulations of the convection-diffusion problem were analysed. Both methods were implemented in the C++ framework LehrFEM++. The predicted optimal convergence rates were observed empirically in the implementation.

Departement of Mathematics
ETH Zurich
Switzerland
October 5, 2019

Acknowledgement

The author is grateful to Prof. Dr. Ralf Hiptmair for suggesting the bachelor thesis project of discontinuous Petrov-Galerkin methods, and for his continued support in discussions and constructive critiques leading to this thesis.

Contents

1 Introduction

In this thesis we analyse the primal and ultraweak Discontinuous Petrov-Galerkin (DPG) formulations of the convection-diffusion problem. Both formulations are implemented in the C++ library LehrFEM++. LehrFEM++ is a framwork for finite element methods currently developed at ETH Zurich [1]. To achieve this goal, the remainder of this thesis is given as follows.

We start in section 2 by presenting definitions of mesh-related quantities, infinite dimensional spaces and polynomial spaces, which will be used in the derivation and discretization of the two analysed methods.

In section 3 we introduce the ideal and practical DPG method for an abstract linear variational problem (LVP). We discuss assumptions that lead to a computationally efficient method and present the main convergence results. This abstract discussion allows us to derive formulas for the computation of local quantities like element stiffness matrices and element load vectors for general DPG methods. In particular these formulas are used in the implementation of the two analysed methods for the convection-diffusion problem.

Starting from a second order formulation we will derive the primal DPG formulation for the convection-diffusion problem in section 4. We present results that establish the convergence of the method and discuss the structure of the resulting local quantities introduced in section 3.

In section 5 we derive the ultraweak DPG formulation starting from a first order reformulation of the convection-diffusion problem. Again we discuss the convergence of the method as well as the structure of local quantities .

In section 6 we provide details of our implementation in LehrFEM++. In particular we discuss the construction of basis functions, the evaluation of local quantities and describe our approach to enforce boundary conditions. Some of the design choices of our implementations were inspired by the interfaces of the general DPG framework Camellia discussed in [2].

Finally we present numerical experiments in section 7. These experiments examine the convergence orders of both methods in the context of a convex domain. We first numerically establish the optimal convergence orders of our implementation and then analyse the convergence behaviour for an exact solution that develops a boundary layer.

2 Definitions and notation

We present basic definitions and specify the notation used throughout this thesis. In section 2.1 we introduce various mesh-related notions. In section 2.2 we recall the definitions of some standard Sobolev spaces, introduce broken Sobolev spaces and specify spaces for interface variables. We finally introduce polynomial spaces on triangular and quadrilateral meshes in section 2.3.

2.1 Mesh

We consider a bounded, simply connected Lipshitz domain $\Omega \subset \mathbb{R}^2$ with polygonal boundary $\partial\Omega$. We denote by $\mathcal{M} = \{K_i\}_{i=1}^n$ a conforming mesh on Ω with edge set $\mathcal{E}(\mathcal{M})$, vertex set $\mathcal{V}(\mathcal{M})$ and mesh skeleton $\mathcal{S} = \cup_{e\in\mathcal{E}(\mathcal{M})} e$. We denote by $h = \max_{K\in\mathcal{M}} \operatorname{diam}(K)$ the mesh width [3, Section 2.1 & 2.2].

For any domain Ω we denote by $\boldsymbol{n}_\Omega : \partial\Omega \to \mathbb{R}^2$ the unit outer normal vector on $\partial\Omega$. We also introduce prescribed edge normals on the mesh skeleton as a function $\boldsymbol{n}_e : \mathcal{S} \to \mathbb{R}^2$. $\boldsymbol{n}_e$ is a unit vector field normal to the edges and points outwards on $\partial\Omega$ [4, Section 2]. For each mesh element $K \in \mathcal{M}$, the outer and prescribed normals on ∂K have to either coincide or point in opposite directions and can be distinguished by the function $\operatorname{sgn}_K : \partial K \to \{-1, 1\}$ given as [5, Equation 3.22]

$$\operatorname{sgn}_K = \begin{cases} 1 & \text{if } \boldsymbol{n}_K = \boldsymbol{n}_e \\ -1 & \text{if } \boldsymbol{n}_K = -\boldsymbol{n}_e \end{cases} \tag{1}$$

A useful identity which follows directly from this definition is

$$\operatorname{sgn}_K \boldsymbol{n}_e = \boldsymbol{n}_K \quad \text{on } \partial K \tag{2}$$

This identity allows us to switch between outer normals and prescribed normals simply by introducing an additional sgn_K factor.

2.2 Infinite dimensional spaces

We introduce several spaces V by specifying an inner product $(\cdot,\cdot)_V$. If not stated otherwise, these spaces are normed by the induced norm $\|v\|_V = \sqrt{(v,v)_V}$.

We denote by $L^2(\Omega)$ the space of square integrable functions on Ω with standard inner product

$$(u,v)_{L^2(\Omega)} = \int_\Omega uv \,\mathrm{d}x$$

and by $\boldsymbol{L}^2(\Omega) = L^2(\Omega) \times L^2(\Omega)$ the space of vector valued functions with components in $L^2(\Omega)$. We also introduce the Sobolev spaces of square integrable functions with square integrable gradient or divergence as

$$H^1(\Omega) = \{u \in L^2(\Omega) : \boldsymbol{\nabla} u \in \boldsymbol{L}^2(\Omega)\}$$
$$H(\mathrm{div}, \Omega) = \{\boldsymbol{q} \in \boldsymbol{L}^2(\Omega) : \boldsymbol{\nabla} \cdot \boldsymbol{q} \in L^2(\Omega)\}$$

with the corresponding inner products [11, Equation 4,67]

$$(u, v)_{H^1(\Omega)} = \int_\Omega uv + \boldsymbol{\nabla} u \cdot \boldsymbol{\nabla} v \, \mathrm{d}x$$
$$(\boldsymbol{q}, \boldsymbol{\sigma})_{H(\mathrm{div},\Omega)} = \int_\Omega \boldsymbol{q} \cdot \boldsymbol{\sigma} + (\boldsymbol{\nabla} \cdot \boldsymbol{q}) (\boldsymbol{\nabla} \cdot \boldsymbol{\sigma}) \, \mathrm{d}x$$

We furthermore define the space

$$H_0^1(\Omega) = \{u \in H^1(\Omega) : u|_{\partial\Omega} = 0\}$$

and denote by $|u|_{H^1(\Omega)} = \|\boldsymbol{\nabla} u\|_{L^2(\Omega)}$ the H^1 semi norm.

We also introduce broken versions of the Sobolev spaces $H^1(\Omega)$ and $H(\mathrm{div}, \Omega)$ as [6, Section 2],[5, Equation 3.1]

$$H^1(\mathcal{M}) = \{u \in L^2(\Omega) : u|_K \in H^1(K), \forall K \in \mathcal{M}\} = \prod_{K \in \mathcal{M}} H^1(K)$$
$$H(\mathrm{div}, \mathcal{M}) = \left\{\boldsymbol{q} \in \boldsymbol{L}^2(\Omega) : \boldsymbol{q}|_K \in H(\mathrm{div}, K), \forall K \in \mathcal{M}\right\} = \prod_{K \in \mathcal{M}} H(\mathrm{div}, K)$$

These spaces drop inter element continuity constraints present in the corresponding unbroken spaces. Functions in $H^1(\mathcal{M})$ are no longer continuous across element boundaries and functions in $H(\mathrm{div}, \mathcal{M})$ have no longer a continuous normal component across element interfaces. The corresponding differential operators can only be applied element-wise. Therefore the standard inner products on these spaces are also defined element-wise by [5, Equation 3.2] [7, Section 3.1]

$$(u, v)_{H^1(\mathcal{M})} = \sum_{K \in \mathcal{M}} (u|_K, v|_K)_{H^1(K)}$$
$$(\boldsymbol{q}, \boldsymbol{\sigma})_{H(\mathrm{div},\mathcal{M})} = \sum_{K \in \mathcal{M}} (\boldsymbol{q}|_K, \boldsymbol{\sigma}|_K)_{H(\mathrm{div},K)}$$

Furthermore, if we test a partial differential equation (PDE) with a function in $H(\mathrm{div}, \mathcal{M})$ or $H^1(\mathcal{M})$, this leads to the introduction of interface variables that belong to the spaces [4, Section 2] [6, Section 2]

$$H^{1/2}(\mathcal{S}) = \left\{\hat{u} : \exists u \in H^1(\Omega) : \hat{u} = u|_{\mathcal{S}}\right\}$$
$$H^{-1/2}(\mathcal{S}) = \{\eta : \exists \boldsymbol{q} \in H(\mathrm{div}, \Omega) : \eta = (\boldsymbol{q} \cdot \boldsymbol{n}_e)|_{\mathcal{S}}\}$$

We will commonly refer to variables in the above spaces as traces resp. fluxes. Since functions in $H^1(\Omega)$ are continuous across element boundaries and functions in $H(\mathrm{div}, \Omega)$ have a continuous normal component, the above spaces consist of functions that are single valued on the mesh skeleton $\mathcal{S}$. Furthermore, $H^{1/2}(\mathcal{S})$ consists of functions that are continuous across the whole skeleton, while functions in $H^{-1/2}(\mathcal{S})$ are generally discontinuous at vertices. The norms on these spaces are specified as [6, Equation 1.a] [4, Equation 2]

$$\|\hat{u}\|_{H^{1/2}(\mathcal{S})} = \inf_{u \in H^1(\Omega) : u|_{\mathcal{S}} = \hat{u}} \|u\|_{H^1(\Omega)}$$
$$\|\hat{q}_n\|_{H^{-1/2}(\mathcal{S})} = \inf_{\boldsymbol{q} \in H(\mathrm{div}, \Omega) : (\boldsymbol{q} \cdot \boldsymbol{n}_e)|_{\mathcal{S}} = \hat{q}_n} \|q\|_{H(\mathrm{div}, \Omega)}$$

We will also require the space [4, Section 2]

$$H_0^{1/2}(\mathcal{S}) = \left\{\hat{u} : \exists u \in H_0^1(\Omega) : \hat{u} = u|_{\mathcal{S}}\right\}$$

consisting of traces of functions in $H_0^1(\Omega)$. Similarly to $H^{1/2}(\mathcal{S})$ this space is normed by [4, Equation 2]

$$\|\hat{u}\|_{H_0^{1/2}(\mathcal{S})} = \inf_{u \in H_0^1(\Omega) : u|_{\mathcal{S}} = \hat{u}} \|u\|_{H^1(\Omega)}$$

2.3 Polynomial spaces

We denote by $\mathcal{P}_p(K)$ the space of multivariate polynomials of degree up to p on a triangle K and define the spaces [3, Section 2.3]

$$\mathcal{P}_p(\mathcal{M}) = \left\{v \in L^2(\Omega) : v|_K \in \mathcal{P}_p(K), \forall K \in \mathcal{M}\right\}$$
$$\mathcal{P}_p^0(\mathcal{M}) = \mathcal{P}_p(\mathcal{M}) \cap C^0(\bar{\Omega})$$
$$\mathcal{P}_p(\mathcal{S}) = \left\{\eta \in L^2(\mathcal{S}) : \eta|_e \in \mathcal{P}_p(e), \forall e \in \mathcal{E}(\mathcal{M})\right\}$$
$$\mathcal{P}_p^0(\mathcal{S}) = \mathcal{P}_p(\mathcal{S}) \cap C^0(\bar{\mathcal{S}})$$

of discontinuous and continuous polynomials on the mesh resp. mesh-skeleton. The first two definitions can be easily extended to meshes of quadrilateral elements by considering the element-wise restrictions in the

space $\mathcal{Q}_p(K)$ of tensor-product polynomials of degree up to p in both components.

$$\mathcal{Q}_p(\mathcal{M}) = \left\{ v \in L^2(\Omega) : v|_K \in \mathcal{Q}_p(K), \forall K \in \mathcal{M} \right\}$$
$$\mathcal{Q}_p^0(\mathcal{M}) = \mathcal{Q}_p(\mathcal{M}) \cap C^0(\bar{\Omega})$$

We furthermore consider the spaces $\mathcal{P}_{p,0}^0(\mathcal{M})$, $\mathcal{P}_{p,0}^0(\mathcal{S})$ and $\mathcal{Q}_{p,0}^0(\mathcal{M})$ of polynomials with zero traces on the boundary $\partial\Omega$.

Clearly $\mathcal{P}_p(\mathcal{M})$ is a finite dimensional subspace of $L^2(\Omega), H^1(\mathcal{M})$ and $H(\mathrm{div}, \mathcal{M})$. $\mathcal{P}_p^0(\mathcal{M})$ is the space of p-th degree Lagrangian polynomials and a well known subspace of $H^1(\Omega)$. We will use the space $\mathcal{P}_p(\mathcal{S})$ for the discretization of flux variables and $\mathcal{P}_p^0(\mathcal{S})$ for the discretization of trace variables [3, Section 2.3].

3 Abstract results

In this section we discuss the DPG method in an abstract setting. In section 3.1 we introduce the ideal and practical DPG method for a general linear variational problem (LVP). We discuss assumptions on the test space that allow an efficient implementation of the practical DPG method and introduce the DPG error estimator. In section 3.2 we present a theorem, which establishes the convergence of the practical DPG method in this abstract setting. In section 3.3 we finally discuss an approach for the computation of element matrices and vectors for the practical DPG method. These considerations allow an implementation of DPG methods by means of standard local assembly techniques.

3.1 Abstract DPG method

In an abstract setting the DPG method considers an LVP between two Hilbert spaces U and V

$$\begin{cases} \text{Find } u \in U \text{ such that} \\ b(u,v) = \ell(v) \quad \forall v \in V \end{cases} \tag{3}$$

Here $b : U \times V \to \mathbb{R}$ is a continuous bilinear form and $\ell : V \to \mathbb{R}$ is a continuous linear form. We denote by $(\cdot,\cdot)_U$, $(\cdot,\cdot)_V$, $\|\cdot\|_U$ and $\|\cdot\|_V$ the inner products and norms on U and V. We will commonly refer to U as the trial space and to V as the test space of the LVP [5, Equation 2.1][8, Equation 1.1].

We are interested in approximating the solution to (3) on a finite dimensional subspace $U_h \subset U$. The ideal DPG method introduces a trial-to-test operator T which is used to compute a space of optimal test functions $V_h = T(U_h)$. The operator $T : U \to V$ is defined by the property [9, Equation 4], [7, Equation 4]

$$(Tu, v)_V = b(u,v) \quad \forall v \in V \tag{4}$$

The ideal DPG method then solves [7, Equation 5], [5, Equation 2.15]

$$\begin{cases} \text{Find } u_h \in U_h \text{ such that} \\ b(u_h, v_h) = \ell(v_h) \quad \forall v_h \in V_h \end{cases} \tag{5}$$

By the definition of T the computation of the optimal test space involves solving another LVP. If we consider a basis $\{b^j_{u,N}\} \subset U_h$, the computation of

$\bar{b}^j_{v,N} = T(b^j_{u,N})$ requires solving [5, Equation 2.16] [10, Equation 1.12]

$$\begin{cases} \text{Find } \bar{b}^j_{v,N} \in V \text{ such that} \\ (\bar{b}^j_{v,N}, v)_V = b(b^j_{u,N}, v) \quad \forall v \in V \end{cases} \tag{6}$$

In general V is infinite dimensional and the solution to (6) can not be computed in closed form. However one can approximate T by T^r on a finite dimensional subspace $V^r \subset V$. To approximate the optimal test space, for each basis function $b^j_{u,N}$ we now solve [8, Equation 1.2] [7, Equation 1.4]

$$\begin{cases} \text{Find } \tilde{b}^j_{v,N} \in V^r \text{ such that} \\ (\tilde{b}^j_{v,N}, v)_V = b(b^j_{u,N}, v) \quad \forall v \in V^r \end{cases} \tag{7}$$

This leads to the practical DPG method, for which we replace the optimal test space $V_h = T(U_h)$ in (5) by the approximated test space $V^r_h = T^r(U_h)$ [7, Equation 1.5]

$$\begin{cases} \text{Find } u_h \in U_h \text{ such that} \\ b(u_h, v) = \ell(v) \quad \forall v \in V^r_h \end{cases} \tag{8}$$

The practical DPG method is completely discrete. For each basis function of U_h we have to solve (7). However this is computationally expensive for standard test spaces. In the case of $V = H^1(\Omega)$ computing one basis function is as expensive as solving a full boundary value problem [7, Section 1]. To get a computationally efficient method, we can choose V^r as the subspace of a broken energy space, as introduced in section 2.2. Since functions in these spaces do not have to satisfy continuity constraints across element boundaries, LVP (7) decouples into element-wise problems, which can be solved efficiently [5, Section 3]. More precisely we choose a variational formulation so that V has the product structure

$$V = \prod_{K \in \mathcal{M}} V(K) \quad (v, \delta v)_V = \sum_{K \in \mathcal{M}} (v|_K, \delta v|_K)_{V(K)} \tag{9}$$

which is the case for $H^1(\mathcal{M})$ and $H(\mathrm{div}, \mathcal{M})$. The element-wise problems are given as [5, Equation 3.5]

$$\begin{cases} \text{Find } \tilde{b}^j_v \in V^r(K) \text{ such that} \\ (\tilde{b}^j_v, v)_{V(K)} = b_K(b^j_u, v) \quad \forall v \in V^r(K) \end{cases} \tag{10}$$

where b_K denotes the contribution of element K to the bilinear form and b^j_u reps. $\tilde{b}^j_v$ denote local shape functions on K of $b^j_{u,N}$ reps. $\tilde{b}^j_{v,N}$

Furthermore, the method provides a posteriori error estimator. Given any function $\tilde{u}_h \in U_h$ we denote by $\tilde{\epsilon}_h \in V^r$ the solution to [6, Equation 26]

$$\begin{cases} \text{Find } \tilde{\epsilon}_h \in V^r \text{ such that} \\ (\tilde{\epsilon}_h, v)_V = \ell(v) - b(\tilde{u}_h, v) \quad \forall v \in V^r \end{cases} \tag{11}$$

If V^r is a broken space as in (9), solving (11) again decouples into element-wise problems

$$\begin{cases} \text{Find } \tilde{\epsilon}_{h,K} \in V^r(K) \text{ such that} \\ (\tilde{\epsilon}_{h,K}, v)_{V(K)} = \ell_K(v) - b_K(\tilde{u}_{h,K}, v) \quad \forall v \in V^r(K) \end{cases} \tag{12}$$

where the quantities in (12) denote the restrictions of global quantities in (11) to element K. Finally we denote by $\tilde{\eta} = \|\tilde{\epsilon}_h\|_V$ the DPG error estimator associated to $\tilde{u}_h$ [6, Section 4.1]

3.2 Convergence results

We now recall the main convergence results for the practical DPG method in this abstract setting. Convergence of the method relies on the following four assumptions. [8, Equation 1.3a-1.3c],[7, Equation 2.1-2.4], [11, Equation 2.2-2.4]

Assumption 1 (Continuity)**.** *There exists a constant $M \geq 0$ such that $\forall w \in U, v \in V$*

$$b(w, v) \leq M \|w\|_U \|v\|_V$$

Assumption 2 (Inf-Sup condition)**.** *There exists a constant γ such that*

$$\inf_{\|u\|_U=1} \sup_{\|v\|_V=1} |b(u, v)| = \gamma > 0$$

Assumption 3 (Uniqueness)**.** *The following subspace is trivial*

$$Z = \{y \in Y : b(x, y) = 0 \, \forall x \in X\} = \{0\}$$

Assumption 4. *There exists a linear operator $\Pi : V \to V^r$ and a constant $C_\Pi > 0$ such that $\forall w_h \in U_h, \forall v \in V$*

$$b(w_h, v - \Pi v) = 0$$
$$\|\Pi v\|_V \leq C_\Pi \|v\|_V$$

Assumptions 1-3 are required to establish the convergence of the ideal DPG method (5) and do not include our choice of the space V^r. The last assumption finally asserts that we choose $V^r \subset V$ large enough to approximate the optimal test space.

The following main theorem establishes the convergence of the practical DPG method and motivates the quantity $\tilde{\eta}$ as an error estimator. [6, Theorem 4.1] [8, Theorem 1.1] [7, Theorem 2.1].

Theorem 3.1 (Quasi optimality)**.** *Let assumptions 1-4 hold. Then the practical DPG method (8) is uniquely solvable for u_h. If u is the unique exact solution of (3), then u_h satisfies the error estimate*

$$\|u - u_h\|_U \leq \frac{MC_\Pi}{\gamma} \inf_{w_h \in U_h} \|u - w_h\|_U$$

Furthermore any $\tilde{u}_h \in U_h$ and its corresponding DPG error estimator $\tilde{\eta}$ satisfy

$$\gamma \|u - \tilde{u}_h\|_U \leq C_\Pi \tilde{\eta} + osc(l)$$
$$\tilde{\eta} \leq M \|u - \tilde{u}_h\|_U$$

where $osc(l) = \|l \circ (1 - \Pi)\|_{V'}$ denotes the data-approximation error.

3.3 Local computations

In this section we derive explicit formulas for the computation of the element stiffness matrix and the element load vector for the practical DPG method (8). We assume that V^r is the subspace of a broken energy space such that we can use (10) to approximate the optimal basis functions element-wise.

Let $\{b_u^1, \dots, b_u^{Q_U}\}$ be the local shape functions of U_h and $\{b_v^1 \dots, b_v^{Q_V}\}$ be the local shape functions of V^r on an arbitrary element K. We denote the approximated local optimal test shape functions by $\tilde{b}_v^i = T^r(b_u^i)$ The entries of the DPG element stiffness matrix $\boldsymbol{A}^K \in \mathbb{R}^{Q_U, Q_U}$ and the DPG element load vector $\boldsymbol{\phi}^K \in \mathbb{R}^{Q_U}$ are given by

$$\boldsymbol{A}^K_{l,j} = b_K(b_u^j, \tilde{b}_v^l) \quad l, j \in \{1, \dots Q_U\}$$
$$\boldsymbol{\phi}^K_l = \ell_K(\tilde{b}_v^l) \quad l \in \{1, \dots Q_U\}$$

We start by solving (10) and compute the coefficient vectors $\boldsymbol{\mu}^l \in \mathbb{R}^{Q_V}$ of $\tilde{b}_v^l$ in the local basis of $V^r(K)$

$$\boldsymbol{G}^K \boldsymbol{\mu}^l = (\boldsymbol{B}^K)_{:,l}$$

Where $\boldsymbol{G}^K \in \mathbb{R}^{Q_V,Q_V}$ is the local Gram matrix on $V^r(K)$ [5, Equation 3.23] [2, Section 3.2]

$$\boldsymbol{G}^K_{i,j} = \left(b^i_v, b^j_v\right)_{V(K)}, \quad i,j \in \{1 \dots Q_V\}$$

and $\boldsymbol{B}^K \in \mathbb{R}^{Q_V,Q_U}$ denotes the extended element stiffness matrix [5, Equation 3.23] [2, Section 3.2]

$$\boldsymbol{B}^K_{i,j} = b_K(b^j_u, b^i_v), \quad i \in \{1, \dots Q_V\},\ j \in \{1, \dots Q_U\}$$

Since the local Gram matrix $\boldsymbol{G}^K$ is symmetric positive definite, its inverse $\left(\boldsymbol{G}^K\right)^{-1}$ exists and we can compute the entries of $\boldsymbol{\mu}^l$ as

$$\boldsymbol{\mu}^l_i = \sum_{k=1}^{Q_V} \left(\left(\boldsymbol{G}^K\right)^{-1}\right)_{i,k} \boldsymbol{B}^K_{k,l}$$

Our approximation of the optimal test function $\tilde{b}^l_v$ is

$$\tilde{b}^l_v = \sum_{i=1}^{Q_V} \boldsymbol{\mu}^l_i b^i_v = \sum_{i=1}^{Q_V}\sum_{k=1}^{Q_V} \left(\left(\boldsymbol{G}^K\right)^{-1}\right)_{i,k} \boldsymbol{B}^K_{k,l} b^i_v$$

Using the bi-linearity of $b_K(\cdot,\cdot)$ and the definition of the extended element stiffness matrix $\boldsymbol{B}^K$ we can compute the entries of $\boldsymbol{A}^K$ as

$$\begin{aligned}\boldsymbol{A}^K_{l,j} &= b_K\left(b^j_u, \tilde{b}^l_v\right) = \sum_{i=1}^{Q_V}\sum_{k=1}^{Q_V} b_K\left(b^j_u, b^i_v\right) \left(\left(\boldsymbol{G}^K\right)^{-1}\right)_{i,k} \boldsymbol{B}^K_{k,l} \\ &= \sum_{i=1}^{Q_V}\sum_{k=1}^{Q_V} \boldsymbol{B}^K_{i,j} \left(\left(\boldsymbol{G}^K\right)^{-1}\right)_{i,k} \boldsymbol{B}^K_{k,l} = \sum_{i=1}^{Q_V}\sum_{k=1}^{Q_V} \left(\left(\boldsymbol{B}^K\right)^T\right)_{l,k} \left(\left(\boldsymbol{G}^K\right)^{-1}\right)_{k,i} \boldsymbol{B}^K_{i,j}\end{aligned}$$

Where we used the symmetry of the local Gramian $\boldsymbol{G}^K$. The DPG element stiffness matrix is finally given by [5, Equation 3.24] [2, Section 3.2]

$$\boldsymbol{A}^K = \left(\boldsymbol{B}^K\right)^T \left(\boldsymbol{G}^K\right)^{-1} \boldsymbol{B}^K \tag{13}$$

The structure of the above matrix product implies, that $\boldsymbol{A}^K$ is positive semi definite. One can show that the resulting global stiffness matrix $\boldsymbol{A}$ is symmetric positive definite [10, Equation 1.16] [2, section 3.1].

For the computation of the DPG element load vector $\boldsymbol{\phi}^K$, we define the extended element load vector $\boldsymbol{l}^K \in \mathbb{R}^{Q_V}$ as [5, Equation 3.23]

$$\boldsymbol{l}^K_i = \ell_K(b^i_v), \quad i \in \{1, \dots, Q_V\}$$

Using linearity of $\ell_K(\cdot)$ and the formulas for $\tilde{b}_v^l$ we find

$$\begin{aligned}\boldsymbol{\phi}_j^K = \ell_K(\tilde{b}_v^j) &= \sum_{i=1}^{Q_V}\sum_{k=1}^{Q_V} \boldsymbol{l}_i^K \left(\left(\boldsymbol{G}^K\right)^{-1}\right)_{i,k} \boldsymbol{B}_{k,j}^K \\ &= \sum_{i=1}^{Q_V}\sum_{k=1}^{Q_V} \left(\left(\boldsymbol{B}^K\right)^T\right)_{j,k} \left(\left(\boldsymbol{G}^K\right)^{-1}\right)_{k,i} \boldsymbol{l}_i^K\end{aligned}$$

where the symmetry of $\boldsymbol{G}^K$ was used in the last step. The DPG element load vector can thus be computed as [5, Equation 3.24]

$$\boldsymbol{\phi}^K = \left(\boldsymbol{B}^K\right)^T \left(\boldsymbol{G}^K\right)^{-1} \boldsymbol{l}^K \tag{14}$$

We finally derive formulas for the computation of the local error representation function $\tilde{\epsilon}_{h,K}$ and the local DPG error estimator $\tilde{\eta}_K^2 = \|\tilde{\epsilon}_h\|_{V(K)}^2$ which are computed by solving LVP (12). We denote by $\tilde{\boldsymbol{\mu}}^K \in \mathbb{R}^{Q_U}$ the local coefficient vector of the finite element function $\tilde{u}_h$. The entries of the residual vector $\tilde{\boldsymbol{r}}^K \in \mathbb{R}^{Q_V}$ corresponding to the right hand side of (12) are computed by

$$\tilde{\boldsymbol{r}}_i^K = \ell_K(b_v^i) - b_K(\tilde{u}_h, b_v^i) = \boldsymbol{l}_i - \sum_{j=1}^{Q_U} \boldsymbol{B}_{i,j}^K \tilde{\boldsymbol{\mu}}_j^K$$

such that the residual vector is given by

$$\tilde{\boldsymbol{r}}^K = \boldsymbol{l}^K - \boldsymbol{B}^K \tilde{\boldsymbol{\mu}}^K \tag{15}$$

The local coefficient vector $\tilde{\boldsymbol{e}}^K \in \mathbb{R}^{Q_V}$ of $\tilde{\epsilon}_{h,K}$ in $V^r(K)$ then satisfies (12)

$$\boldsymbol{G}^K \tilde{\boldsymbol{e}}^K = \tilde{\boldsymbol{r}}^K$$

Finally, we can compute $\tilde{\eta}_K^2$ by the definition of the local Gram matrix $\boldsymbol{G}^K$ as [2, Section 3.3]

$$\begin{aligned}\tilde{\eta}_K^2 = \|\tilde{\epsilon}_h\|_{V(K)} &= \left(\tilde{\boldsymbol{e}}^K\right)^T \boldsymbol{G}^K \tilde{\boldsymbol{e}}^K = \left(\left(\boldsymbol{G}^K\right)^{-1} \tilde{\boldsymbol{r}}^K\right)^T \boldsymbol{G}^K \left(\left(\boldsymbol{G}^K\right)^{-1} \tilde{\boldsymbol{r}}^K\right) \\ &= \left(\tilde{\boldsymbol{r}}^K\right)^T \left(\boldsymbol{G}^K\right)^{-1} \tilde{\boldsymbol{r}}^K\end{aligned} \tag{16}$$

The global DPG error estimator $\tilde{\eta}^2$ can be computed by summing up the local estimators over the mesh.

$$\tilde{\eta}^2 = \sum_{K \in \mathcal{M}} \tilde{\eta}_K^2$$

4 Primal DPG formulation

In this section we derive the primal DPG formulation starting from a second order formulation of the convection-diffusion problem. In section 4.1 we introduce the model problem, fix the regularity assumptions on the coefficients and define the in- and outflow boundaries. In section 4.2 we provide a detailed derivation of the primal formulation for the convection-diffusion problem and specify the energy setting in the case of pure homogeneous Dirichlet boundary conditions. In section 4.3 we present the main convergence results for the primal formulation. Finally in section 4.4 we give an overview of the structure of the resulting local quantities.

4.1 Model problem

We consider the convection-diffusion model problem. We seek u such that

$$\begin{cases} -\epsilon\Delta u + \boldsymbol{\beta}\cdot\boldsymbol{\nabla} u = f & \text{in } \Omega \\ u = g & \text{on } \Gamma_1 \\ \boldsymbol{n}_\Omega\cdot\boldsymbol{\nabla} u = h & \text{on } \Gamma_2 \end{cases} \tag{17}$$

For the convergence analysis we consider the diffusion coefficient $\epsilon \in \mathbb{R}_{>0}$ and the prescribed advection field $\boldsymbol{\beta} \in \mathbb{R}^2$. The source function $f : \Omega \to \mathbb{R}$ is assumed in $L^2(\Omega)$ [6, Example 3.5]. We impose Dirichlet and Neumann boundary conditions on two distinct parts of the boundary $\Gamma_1, \Gamma_2 \subset \partial\Omega$. A natural choice for Γ_1 and Γ_2 are the in- and outflow boundaries, which are defined as [5, equation 5.2]

$$\Gamma_1 = \Gamma_{in} = \{x \in \partial\Omega : \boldsymbol{\beta}\cdot\boldsymbol{n}_\Omega < 0\}$$
$$\Gamma_2 = \Gamma_{out} = \{x \in \partial\Omega : \boldsymbol{\beta}\cdot\boldsymbol{n}_\Omega \geq 0\}$$

4.2 Variational formulation

In order to apply the DPG method to this problem, we multiply the PDE with a broken test function $v \in H^1(\mathcal{M})$ and integrate over an element $K \in \mathcal{M}$. Since $v|_K \in H^1(K)$, we can apply integration by parts on the element K. We then sum up the contributions from all elements $K \in \mathcal{M}$. [5, Equation 3.11]

$$
\begin{aligned}
-\int_K \epsilon \Delta u v \mathrm{dx} + \int_K \boldsymbol{\beta} \cdot \boldsymbol{\nabla} u v \mathrm{dx} &= \int_K f v \mathrm{dx} \\
\int_K \epsilon \boldsymbol{\nabla} u \cdot \boldsymbol{\nabla} v \mathrm{dx} - \int_{\partial K} \epsilon \boldsymbol{\nabla} u \cdot \boldsymbol{n}_K v \mathrm{dS} + \int_K \boldsymbol{\beta} \cdot \boldsymbol{\nabla} u v \mathrm{dx} &= \int_K f v \mathrm{dx} \\
\sum_{K \in \mathcal{M}} \int_K \epsilon \boldsymbol{\nabla} u \cdot \boldsymbol{\nabla} v \mathrm{dx} - \int_{\partial K} \epsilon \boldsymbol{\nabla} u \cdot \boldsymbol{n}_K v \mathrm{dS} + \int_K \boldsymbol{\beta} \cdot \boldsymbol{\nabla} u v \mathrm{dx} &= \sum_{K \in \mathcal{M}} \int_K f v \mathrm{dx}
\end{aligned}
$$

As described in section 2.2 a function $v \in H^1(\mathcal{M})$ is no longer continuous across the element interfaces and the boundary integrals do not cancel each other on the mesh skeleton $\mathcal{S}$ [5, Section 3].

Since we choose $u \in H^1(\Omega)$, the value of the gradient $\boldsymbol{\nabla} u \in \boldsymbol{L}^2(\Omega)$ is undefined on the element boundaries ∂K. We introduce a new unknown $\hat{q}_n \in H^{-1/2}(\mathcal{S})$, which represents the fluxes on the mesh skeleton $\hat{q}_n = \boldsymbol{\nabla} u \cdot \boldsymbol{n}_e$. To achieve this, we have to replace the outer normals $\boldsymbol{n}_K$ by prescribed edge normals $\boldsymbol{n}_e$. According to identity (2), this simply results in an additional sgn_K factor in the variational formulation [5, Equation 3.17-3.19] [6, Example 3.5].

$$
\begin{aligned}
\sum_{K \in \mathcal{M}} \int_K \epsilon \boldsymbol{\nabla} u \cdot \boldsymbol{\nabla} v \mathrm{dx} - \int_{\partial K} \epsilon \boldsymbol{\nabla} u \cdot \boldsymbol{n}_e \mathrm{sgn}_K v \mathrm{dS} + \int_K \boldsymbol{\beta} \cdot \boldsymbol{\nabla} u v \mathrm{dx} &= \sum_{K \in \mathcal{M}} \int_K f v \mathrm{dx} \\
\sum_{K \in \mathcal{M}} \int_K \epsilon \boldsymbol{\nabla} u \cdot \boldsymbol{\nabla} v \mathrm{dx} - \int_{\partial K} \epsilon \hat{q}_n \mathrm{sgn}_K v \mathrm{dS} + \int_K \boldsymbol{\beta} \cdot \boldsymbol{\nabla} u v \mathrm{dx} &= \sum_{K \in \mathcal{M}} \int_K f v \mathrm{dx}
\end{aligned}
$$

To simplify further notation we define

$$
\begin{aligned}
b((u, \hat{q}_n), v) &= \sum_{K \in \mathcal{M}} \int_K \epsilon \boldsymbol{\nabla} u \cdot \boldsymbol{\nabla} v \mathrm{dx} - \int_{\partial K} \epsilon \hat{q}_n \mathrm{sgn}_K v \mathrm{dS} + \int_K \boldsymbol{\beta} \cdot \boldsymbol{\nabla} u v \mathrm{dx} \\
\ell(v) &= \sum_{K \in \mathcal{M}} \int_K f v \mathrm{dx}
\end{aligned}
$$

Furthermore, we can transform the boundary condition on $\boldsymbol{\nabla} u \cdot \boldsymbol{n}_\Omega$ into a boundary condition for the flux $\hat{q}_n$. By definition the prescribed edge normals $\boldsymbol{n}_e$ point outwards on the boundary $\partial\Omega$. We thus can transform the Neumann boundary condition directly into an essential boundary condition on the flux $\hat{q}_n$ without any sign adjustments.

Our variational formulation finally reads as follows [8, Equation 2.2] [5, Equation 3.19] [12, Equation 2.9]:

Seek $u \in H^1(\Omega)$ and $\hat{q}_n \in H^{-1/2}(\mathcal{S})$ such that

$$\begin{cases} b((u,\hat{q}_n),v) = \ell(v)\,\forall v \in H^1(\mathcal{M}) \\ u = g, \text{ on } \Gamma_1 \\ \hat{q}_n = h, \text{ on } \Gamma_2 \end{cases}$$

For the following convergence analysis we consider homogenous Dirichlet boundary conditions. The general boundary conditions in (17) can be enforced by using offset-function techniques which we describe in section 6.3. We set the trial and test space to [8, Section 2]

$$\begin{aligned} U &= H_0^1(\Omega) \times H^{-1/2}(\mathcal{S}) \\ V &= H^1(\mathcal{M}) \end{aligned}$$

The spaces are normed by

$$\begin{aligned} \|(w,\hat{r}_n)\|_U^2 &= |w|_{H^1(\Omega)} + \|\hat{r}_n\|_{H^{-1/2}(\mathcal{S})}^2 \\ \|v\|_V &= \|v\|_{H^1(\mathcal{M})} \end{aligned}$$

and the variational formulation we are interested to analyse reads as

$$\begin{cases} \text{Find } (u,\hat{q}_n) \in U \text{ such that} \\ b((u,\hat{q}_n),v) = \ell(v) \quad \forall v \in V \end{cases} \tag{18}$$

4.3 Convergence results

We now specify the finite dimensional polynomial subspaces $U_h \subset U$ and $V^r \subset V$ of the practical DPG method (8). For simplicity we consider a triangular mesh. For $k, r \in \mathbb{N}$ we set [8, Equation 4.1]

$$U_h = \{(w,\hat{r}_n) \in U : w|_K \in P_{k+1}(K), \hat{r}_n|_{\partial K} \in P_k(\partial K) \quad \forall K \in \mathcal{M}\}$$

$$V^r = \{v \in V : v|_K \in P_r(K) \quad \forall K \in \mathcal{M}\}$$

In particular by our discussion of polynomial spaces in section 2.3 we have $U_h = \mathcal{P}^0_{k+1,0}(\mathcal{M}) \times \mathcal{P}_k(\mathcal{S})$ and $V^r = \mathcal{P}_r(\mathcal{M})$.

In this setting the following quasi-optimality result holds [8, Theorem 4.1].

Theorem 4.1 (Quasi optimality)**.** *Suppose that $r \geq k+2$ and $\boldsymbol{\beta} = 0$. Let $(u,\hat{q}_n)$ denote the exact and $(u_h,\hat{q}_{n,h})$ the discrete solution for the primal*

DPG method. Let the discretization error $\mathcal{D}$ and the approximation error $\mathcal{A}$ be denoted by

$$\mathcal{D} = \|u - u_h\|_{H^1(\Omega)} + \|\hat{q}_n - \hat{q}_{n,h}\|_{H^{-1/2}(\mathcal{S})}$$
$$\mathcal{A} = \inf_{(w,\hat{r}_{n,h})\in U_h} \left(\|u - w_h\|_{H^1(\Omega)} + \|\hat{q}_n - \hat{r}_{n,h}\|_{H^{-1/2}(\mathcal{S})}\right)$$

Then there is a constant $C > 0$ independent of h such that

$$\mathcal{D} \leq C\mathcal{A}$$

In the case $\boldsymbol{\beta} \neq 0$, the variational formulation remains valid [6, Example 3.5]. We will perform numerical experiments to examine if an enrichment of $r = k + 2$ is still sufficient in this case.

Furthermore, under the assumptions of theorem 4.1 and under additional smoothness assumptions on the exact solution u, the following result regarding convergence rates holds [8, Equation 5.1].

Corollary 4.2 (Convergence rates)**.** *Let the assumptions of theorem 4.1 hold. Let the Discretization error $\mathcal{D}$ be denoted by*

$$\mathcal{D} = \|u - u_h\|_{H^1(\Omega)} + \|\hat{q}_n - \hat{q}_{n,h}\|_{H^{-1/2}(\Gamma_h)}$$

and let $\boldsymbol{q} = -\boldsymbol{\nabla} u$. Then for all $s \leq k + 1$

$$\mathcal{D} \leq Ch^s \left(|u|_{H^{s+1}(\Omega)} + |\boldsymbol{\nabla} \cdot \boldsymbol{q}|_{H^s(\Omega)}\right)$$

In particular, if the solution is smooth enough and if we use polynomials of degree p to approximate the u-part of the solution, we expect algebraic h-convergence of rate p.

4.4 Local quantities

We give explicit formulas for the computation of the local quantities introduced section 3.3. We assume that on an element K for all components of the trial and test space $d \in \{u, \hat{q}_n, v\}$ a set of local shape functions $\{b_d^1, \dots b_d^{Q_d}\}$ is given.

The local Gram matrix $\boldsymbol{G}^K \in \mathbb{R}^{Q_v, Q_v}$ corresponds to the inner product in $V(K)$ and its entries are given by

$$\left(\boldsymbol{G}^K\right)_{i,j} = \left(b_v^i, b_v^j\right)_{V(K)} = \int_K b_v^i b_v^j + \boldsymbol{\nabla} b_v^i \cdot \boldsymbol{\nabla} b_v^j \mathrm{dx}$$

The extended element stiffness matrix has the following structure

$$\boldsymbol{B}^K = \left[\begin{array}{cc} \boldsymbol{B}^{K,u} & \boldsymbol{B}^{K,\hat{q}_n} \end{array} \right] \in \mathbb{R}^{Q_v, Q_u + Q_{\hat{q}n}}$$

and the sub matrices are given by

$$\left(\boldsymbol{B}^{K,u}\right)_{i,j} = b_K((b_u^j, 0), b_v^i) = \int_K \epsilon \boldsymbol{\nabla} b_u^j \cdot \boldsymbol{\nabla} b_v^i + \boldsymbol{\beta} \cdot \boldsymbol{\nabla} b_u^j \; b_v^i \mathrm{dx}$$

$$\left(\boldsymbol{B}^{K,\hat{q}_n}\right)_{i,j} = b_K((0, b_{\hat{q}_n}^j), b_v^i) = \int_{\partial K} -\mathrm{sgn}_K b_{\hat{q}_n}^j \, b_v^i \mathrm{dS}$$

Finally, the extended load vector $\boldsymbol{l}^K \in \mathbb{R}^{Q_v}$ is computed as

$$\left(\boldsymbol{l}^K\right)_i = \ell_K(b_v^i) = \int_K f b_v^i \mathrm{dx}$$

5 Ultraweak DPG formulation

In the previous section we derived the primal DPG formulation by choosing a broken test function $v \in H^1(\mathcal{M})$, while leaving the trial space unbroken by choosing $u \in H^1(\Omega)$. To derive the ultraweak DPG formulation we additionally break the trial space. To achieve this we start from a first order reformulation of the model problem which we introduce in 5.1. We then proceed in the same way as for the primal DPG method in section 4.

5.1 Model problem

We consider a first order reformulation of the model problem (17). We seek u and $\boldsymbol{\sigma}$ satisfying

$$\begin{cases} \dfrac{1}{\epsilon}\boldsymbol{\sigma} - \boldsymbol{\nabla} u = 0 & \text{in } \Omega \\ -\boldsymbol{\nabla}\cdot\boldsymbol{\sigma} + \boldsymbol{\nabla}\cdot(\boldsymbol{\beta} u) = f & \text{in } \Omega \\ \boldsymbol{n}_\Omega \cdot (\boldsymbol{\sigma} - \boldsymbol{\beta} u) = g & \text{on } \Gamma_1 \\ u = u_0 & \text{on } \Gamma_2 \end{cases} \tag{19}$$

Here $\boldsymbol{\sigma}$ represents the scaled gradient $\epsilon\boldsymbol{\nabla} u$. We consider the same conditions on the coefficient functions as in section 4.1. The boundary conditions are the ones proposed in [13, Section 2]

5.2 Variational formulation

We again multiply the equations in (19) with broken test functions $v \in H^1(\mathcal{M})$, $\boldsymbol{\tau} \in H(\mathrm{div}, \mathcal{M})$ and integrate over a single element K.

$$\begin{cases} \displaystyle\int_K \frac{1}{\epsilon}\boldsymbol{\sigma}\cdot\boldsymbol{\tau}\mathrm{dx} - \int_K \boldsymbol{\nabla} u \cdot \boldsymbol{\tau}\mathrm{dx} = 0 \\ \displaystyle -\int_K (\boldsymbol{\nabla}\cdot\boldsymbol{\sigma}) v\mathrm{dx} + \int_K \boldsymbol{\nabla}\cdot(\boldsymbol{\beta} u) v\mathrm{dx} = \int_K f v\mathrm{dx} \end{cases}$$

Our goal is to arrive at a variational formulation, where we can choose $u \in L^2(\Omega)$ and $\boldsymbol{\sigma} \in \boldsymbol{L}^2(\Omega)$. To achieve this we have to shift all derivatives from the trial to the test functions. Since the test functions are locally in $H(\mathrm{div}, K)$ and $H^1(K)$, we can perform integration by parts on K

$$\begin{cases} \displaystyle\int_K \frac{1}{\epsilon}\boldsymbol{\sigma}\cdot\boldsymbol{\tau}\mathrm{dx} + \int_K u\boldsymbol{\nabla}\cdot\tau\mathrm{dx} - \int_{\partial K} u\boldsymbol{\tau}\cdot\boldsymbol{n}_K\mathrm{dS} = 0 \\ \displaystyle\int_K \boldsymbol{\sigma}\cdot\boldsymbol{\nabla} v\mathrm{dx} - \int_{\partial K}\boldsymbol{\sigma}\cdot\boldsymbol{n}_K v\mathrm{dS} - \int_K u\boldsymbol{\beta}\cdot\boldsymbol{\nabla} v\mathrm{dx} + \int_{\partial K} u\boldsymbol{\beta}\cdot\boldsymbol{n}_K v\mathrm{dS} = \int_K f v\mathrm{dx} \end{cases}$$

Summing the two equations we arrive at

$$\int_K \boldsymbol{\sigma}\cdot\boldsymbol{\nabla}v + \frac{1}{\epsilon}\boldsymbol{\sigma}\cdot\boldsymbol{\tau} + u\boldsymbol{\nabla}\cdot\boldsymbol{\tau} - u\boldsymbol{\beta}\cdot\boldsymbol{\nabla}v\mathrm{dx} \\ - \int_{\partial K}(\boldsymbol{\sigma}-u\boldsymbol{\beta})\cdot\boldsymbol{n}_K v + u\boldsymbol{\tau}\cdot\boldsymbol{n}_K\mathrm{dS} = \int_K fv\mathrm{dx}$$

Since we only assume $u \in L^2(\Omega)$, its value on the element boundary is no longer well defined and we represent it with the trace unknown $\hat{u} \in H^{1/2}(\mathcal{S})$. Similarly, since $\boldsymbol{\sigma} \in \boldsymbol{L}^2(\Omega)$, the value of the flux $(\boldsymbol{\sigma}-u\boldsymbol{\beta})\cdot\boldsymbol{n}_K$ is no longer well defined. We again introduce the prescribed normals $\boldsymbol{n}_e$ into the variational formulation via identity (2) and introduce the flux variable $\hat{\sigma}_n \in H^{-1/2}(\mathcal{S})$ representing the total flux across element boundaries $\hat{\sigma}_n = \boldsymbol{n}_e\cdot(\boldsymbol{\sigma}-\boldsymbol{\beta}u)$. Finally summing up the contributions from all elements $K \in \mathcal{M}$, we arrive at a variational formulation that reads as follows [13, Section 2.1] [10, Equation 4.58] [5, Equation 5.3]

$$\sum_{K\in\mathcal{M}}\int_K \boldsymbol{\sigma}\cdot\boldsymbol{\nabla}v + \frac{1}{\epsilon}\boldsymbol{\sigma}\cdot\boldsymbol{\tau} + u\boldsymbol{\nabla}\cdot\tau - u\boldsymbol{\beta}\cdot\boldsymbol{\nabla}v\mathrm{dx} \\ - \sum_{K\in\mathcal{M}}\int_{\partial K}\mathrm{sgn}_K\hat{\sigma}_n v + \hat{u}\boldsymbol{\tau}\cdot\boldsymbol{n}_K\mathrm{dS} = \sum_{K\in\mathcal{M}}\int_K fv\mathrm{dx}$$

To simplify notation we introduce the following bilinear and linear forms

$$\begin{aligned} b\left((u,\boldsymbol{\sigma},\hat{u},\hat{\sigma}_n),(v,\boldsymbol{\tau})\right) &= \sum_{K\in\mathcal{M}}\int_K \boldsymbol{\sigma}\cdot\boldsymbol{\nabla}v + \frac{1}{\epsilon}\boldsymbol{\sigma}\cdot\boldsymbol{\tau} + u\boldsymbol{\nabla}\cdot\tau - u\boldsymbol{\beta}\cdot\boldsymbol{\nabla}v\mathrm{dx} \\ &\quad - \sum_{K\in\mathcal{M}}\int_{\partial K}\mathrm{sgn}_K\hat{\sigma}_n v + \hat{u}\boldsymbol{\tau}\cdot\boldsymbol{n}_K\mathrm{dS} \\ \ell((v,\boldsymbol{\tau})) &= \sum_{K\in\mathcal{M}}\int_K fv\mathrm{dx} \end{aligned}$$

As for the primal formulation, we can transform the boundary conditions in (19) to essential boundary conditions on the flux $\hat{\sigma}_n$ and the trace $\hat{u}$. Our variational formulation thus reads as follows:

Find u, $\boldsymbol{\sigma}$,$\hat{\sigma}_n$ and $\hat{u}$ such that

$$\begin{cases} b\left((u,\boldsymbol{\sigma},\hat{u},\hat{\sigma}_n),(v,\boldsymbol{\tau})\right) = l(v,\boldsymbol{\tau}) \quad \forall(v,\boldsymbol{\tau}) \in H^1(\mathcal{M})\times H(\mathrm{div},\mathcal{M}) \\ \hat{\sigma}_n = g \quad \text{on } \Gamma_1 \\ \hat{u} = u_0 \quad \text{on } \Gamma_2 \end{cases}$$

To simplify the convergence analysis we again consider the case of a problem with pure homogenous Dirichlet boundary conditions. The correct energy setting in this case is given by [7, Section 3.1]

$$\begin{aligned} U &= L^2(\Omega) \times \boldsymbol{L}^2(\Omega) \times H_0^{1/2}(\mathcal{S}) \times H^{-1/2}(\mathcal{S}) \\ V &= H^1(\mathcal{M}) \times H(\mathrm{div}, \mathcal{M}) \end{aligned}$$

The norms on U and V are specified as [7, Section 3.1] [13, Section 2.1-2.2]

$$\begin{aligned} \|(u, \boldsymbol{\sigma}, \hat{u}, \hat{\sigma}_n)\|_U^2 &= \|u\|_{L^2(\Omega)}^2 + \|\boldsymbol{\sigma}\|_{\boldsymbol{L}^2(\Omega)}^2 + \|\hat{u}\|_{H_0^{1/2}(\mathcal{S})}^2 + \|\hat{\sigma}_n\|_{H^{-1/2}(\mathcal{S})}^2 \\ \|(v, \boldsymbol{\tau})\|_V^2 &= \|v\|_{H^1(\mathcal{M})}^2 + \|\boldsymbol{\tau}\|_{H(\mathrm{div}, \mathcal{M})}^2 \end{aligned}$$

A standard choice for the test-space inner product which induces the above norm is [10, Equation 4.66]

$$((v, \boldsymbol{\tau}), (\delta v, \delta \boldsymbol{\tau}))_V = (v, \delta v)_{H^1(\mathcal{M})} + (\boldsymbol{\tau}, \delta \boldsymbol{\tau})_{H(\mathrm{div}, \mathcal{M})}$$

Finally, the variational formulation, we are interested to analyse, reads as

$$\begin{cases} \text{Find } (u, \boldsymbol{\sigma}, \hat{u}, \hat{\sigma}_n) \in U \text{ such that} \\ b\left((u, \boldsymbol{\sigma}, \hat{u}, \hat{\sigma}_n), (v, \boldsymbol{\tau})\right) = l(v, \boldsymbol{\tau}) \quad \forall (v, \boldsymbol{\tau}) \in V \end{cases} \tag{20}$$

5.3 Convergence results

We again specify the finite dimensional polynomial subspaces $U_h \subset U$ and $V^r \subset V$ needed in the practical DPG method (8). For $p, r \in \mathbb{N}$ we set [7, Section 3.3]

$$\begin{aligned} U_h = \{(u, \boldsymbol{\sigma}, \hat{u}, \hat{\sigma}_n) \in U : & u|_K \in \mathcal{P}_p(K), \\ & \boldsymbol{\sigma}|_K \in \boldsymbol{\mathcal{P}}_p(K), \\ & \hat{u}|_{\partial K} \in \mathcal{P}_{p+1}^0(\partial K), \\ & \hat{\sigma}_n|_{\partial K} \in \mathcal{P}_p(\partial K) \quad \forall K \in \mathcal{M}\} \\ V^r = \{(v, \boldsymbol{\tau}) \in V : & v|_K \in \mathcal{P}_r(K) \\ & \boldsymbol{\tau}|_K \in \boldsymbol{\mathcal{P}}_r(K) \quad \forall K \in \mathcal{M}\} \end{aligned}$$

By our discussion in section 2.2 the finite dimensional spaces can be directly specified as $U_h = \mathcal{P}_p(\mathcal{M}) \times \boldsymbol{\mathcal{P}}_p(\mathcal{M}) \times \mathcal{P}_{p+1,0}^0(\mathcal{S}) \times \mathcal{P}_p(\mathcal{S})$ and $V^r = \mathcal{P}_p(\mathcal{M}) \times \boldsymbol{\mathcal{P}}_p(\mathcal{M})$ [3, Section 2.5].

In this setting the following quasi-optimality result holds [7, Theorem 3.4]

Theorem 5.1 (Quasi optimality). *Suppose that $r \geq p+2$ and $\boldsymbol{\beta} = 0$. Let $(u, \boldsymbol{\sigma}, \hat{u}, \hat{\sigma}_n)$ denote the exact and $(u_h, \boldsymbol{\sigma}_h, \hat{u}_h, \hat{\sigma}_{n,h})$ the discrete solution for the ultraweak DPG method. Let the discretization error $\mathcal{D}$ and approximation error $\mathcal{A}$ be denoted by*

$$\mathcal{D} = \|u - u_h\|^2_{L^2(\Omega)} + \|\boldsymbol{\sigma} - \boldsymbol{\sigma}_h\|^2_{\boldsymbol{L}^2(\Omega)} + \|\hat{u} - \hat{u}_h\|^2_{H_0^{1/2}(\mathcal{S})} + \|\hat{\sigma}_n - \hat{\sigma}_{n,h}\|^2_{H^{-1/2}(\mathcal{S})}$$

$$\mathcal{A} = \inf_{(w_h, \boldsymbol{\rho}_h, \hat{\omega}_h, \hat{\eta}_{h,n}) \in U_h} \left(\|u - w_h\|^2_{L^2(\Omega)} + \|\boldsymbol{\sigma} - \boldsymbol{\rho}_h\|^2_{\boldsymbol{L}^2(\Omega)} + \|\hat{u} - \hat{\omega}_h\|^2_{H_0^{1/2}(\mathcal{S})} + \|\hat{\sigma}_n - \hat{\eta}_{h,n}\|^2_{H^{-1/2}(\mathcal{S})} \right)$$

Then there is a constant $C \geq 0$ independent of h such that

$$\mathcal{D} \leq C\mathcal{A}$$

In the case of $\boldsymbol{\beta} \neq 0$, the variational formulation remains valid [6, Example 3.5]. We will again perform numerical experiments to examine the effect of the additional convection term.

Regarding the expected convergence rates, the following result holds under the assumption of theorem 5.1 and under additional smoothness assumptions on the exact solution u [7, Corollary 3.6].

Corollary 5.2 (Convergence rates). *Let the assumptions of theorem 5.1 hold. Let the discretization error $\mathcal{D}$ be denoted by*

$$\mathcal{D} = \|u - u_h\|^2_{L^2(\Omega)} + \|\boldsymbol{\sigma} - \boldsymbol{\sigma}_h\|^2_{\boldsymbol{L}^2(\Omega)} + \|\hat{u} - \hat{u}_h\|^2_{H_0^{1/2}(\mathcal{S})} + \|\hat{\sigma}_n - \hat{\sigma}_{n,h}\|^2_{H^{-1/2}(\mathcal{S})}$$

Then for all $s \in (1/2, p+1]$

$$\mathcal{D} \leq Ch^s \left(\|u\|_{H^{s+1}(\Omega)} + \|\boldsymbol{\sigma}\|_{H^{s+1}(\Omega)} \right)$$

In particular, for smooth enough solutions, we expect algebraic h-convergence of order $p+1$, if we approximate the u and $\boldsymbol{\sigma}$ component of our solution with polynomials of degree p.

5.4 Local quantities

We again present explicit formulas for the computation of the local quantities introduced in section 3.3. By our choice of trial and test space in section 5.3, it is clear that the components of the vector-valued functions $\boldsymbol{\sigma}$ and $\boldsymbol{\tau}$ can be described by independent shape functions. So we assume, that for each

component of the trial and test space $d \in \{u, \sigma_x, \sigma_y, \hat{u}, \hat{q}_n, v, \tau_x, \tau_y\}$ on an element K a set of local shape functions $\{b_d^1, \cdots b_d^{Q_d}\}$ is given.

The local Gram matrix has the following structure

$$\boldsymbol{G}^K = \begin{bmatrix} \boldsymbol{G}^{K,v,v} & 0 & 0 \\ 0 & \boldsymbol{G}^{K,\tau_x,\tau_x} & \boldsymbol{G}^{K,\tau_x,\tau_y} \\ 0 & \boldsymbol{G}^{K,\tau_y,\tau_x} & \boldsymbol{G}^{K,\tau_y,\tau_y} \end{bmatrix} \in \mathbb{R}^{Q_V,Q_V}$$

where the sub matrices are computed as

$$\begin{aligned}
\left(\boldsymbol{G}^{K,v,v}\right)_{i,j} &= \int_K b_v^i b_v^j + \boldsymbol{\nabla} b_v^i \cdot \boldsymbol{\nabla} b_v^j \, \mathrm{dx} \\
\left(\boldsymbol{G}^{K,\tau_x,\tau_x}\right)_{i,j} &= \int_K b_{\tau_x}^i b_{\tau_x}^j + \frac{\partial b_{\tau_x}^i}{\partial x} \frac{\partial b_{\tau_x}^j}{\partial x} \mathrm{dx} \\
\left(\boldsymbol{G}^{K,\tau_y,\tau_y}\right)_{i,j} &= \int_K b_{\tau_y}^i b_{\tau_y}^j + \frac{\partial b_{\tau_y}^i}{\partial y} \frac{\partial b_{\tau_y}^j}{\partial y} \mathrm{dx} \\
\left(\boldsymbol{G}^{K,\tau_x,\tau_y}\right)_{i,j} &= \int_K \frac{\partial b_{\tau_x}^i}{\partial x} \frac{\partial b_{\tau_y}^j}{\partial y} \mathrm{dx} \\
\left(\boldsymbol{G}^{K,\tau_y,\tau_x}\right)_{i,j} &= \int_K \frac{\partial b_{\tau_y}^i}{\partial y} \frac{\partial b_{\tau_x}^j}{\partial x} \mathrm{dx}
\end{aligned}$$

Similarly the extended element stiffness matrix will have the following structure

$$\boldsymbol{B}^K = \begin{bmatrix} \boldsymbol{B}^{K,v,u} & \boldsymbol{B}^{K,v,\sigma_x} & \boldsymbol{B}^{K,v,\sigma_y} & 0 & \boldsymbol{B}^{K,v,\hat{\sigma}_n} \\ \boldsymbol{B}^{K,\tau_x,u} & \boldsymbol{B}^{K,\tau_x,\sigma_x} & 0 & \boldsymbol{B}^{K,\tau_x,\hat{u}} & 0 \\ \boldsymbol{B}^{K,\tau_y,u} & 0 & \boldsymbol{B}^{K,\tau_y,\sigma_y} & \boldsymbol{B}^{K,\tau_y,\hat{u}} & 0 \end{bmatrix} \in \mathbb{R}^{Q_V,Q_U}$$

and the entries of the sub-matrices are given by

$$\begin{aligned}
\left(\boldsymbol{B}^{K,v,u}\right)_{i,j} &= -\int_K b_u^j \boldsymbol{\beta} \cdot \boldsymbol{\nabla} b_v^i \mathrm{dx} \\
\left(\boldsymbol{B}^{K,v,\sigma_x}\right)_{i,j} &= \int_K b_{\sigma_x}^j \frac{\partial b_v^i}{\partial x} \mathrm{dx} \\
\left(\boldsymbol{B}^{K,v,\sigma_y}\right)_{i,j} &= \int_K b_{\sigma_y}^j \frac{\partial b_v^i}{\partial y} \mathrm{dx} \\
\left(\boldsymbol{B}^{K,v,\hat{\sigma}_n}\right)_{i,j} &= -\int_{\partial K} \mathrm{sgn}_K b_{\hat{\sigma}_n}^j b_v^i \mathrm{dS}
\end{aligned}$$

$$
\begin{aligned}
\left(\boldsymbol{B}^{K,\tau_x,u}\right)_{i,j} &= \int_K b_u^j \frac{\partial b_{\tau_x}^i}{\partial x} \mathrm{dx} \\
\left(\boldsymbol{B}^{K,\tau_y,u}\right)_{i,j} &= \int_K b_u^j \frac{\partial b_{\tau_y}^i}{\partial y} \mathrm{dx} \\
\left(\boldsymbol{B}^{K,\tau_x,\sigma_x}\right)_{i,j} &= \int_K \frac{1}{\epsilon} b_{\sigma_x}^j b_{\tau_x}^i \mathrm{dx} \\
\left(\boldsymbol{B}^{K,\tau_y,\sigma_y}\right)_{i,j} &= \int_K \frac{1}{\epsilon} b_{\sigma_y}^j b_{\tau_y}^i \mathrm{dx} \\
\left(\boldsymbol{B}^{K,\tau_x,\hat{u}}\right)_{i,j} &= -\int_{\partial K} b_{\hat{u}}^j b_{\tau_x}^i n_{K,x} \mathrm{dS} \\
\left(\boldsymbol{B}^{K,\tau_y,\hat{u}}\right)_{i,j} &= -\int_{\partial K} b_{\hat{u}}^j b_{\tau_y}^i n_{K,y} \mathrm{dS}
\end{aligned}
$$

Finally the structure of the extended element load vector is given as

$$
\boldsymbol{l}^K = \begin{bmatrix} \boldsymbol{l}^{K,v} \\ 0 \\ 0 \end{bmatrix} \in \mathbb{R}^{Q_V}
$$

and the sub vector $\boldsymbol{l}^{K,v}$ is computed as

$$
\left(\boldsymbol{l}^{K,v}\right)_i = \int_K f b_v^i \mathrm{dx}
$$

6 Implementation

We present details of our implementation of the two DPG methods in the context of the C++ library LehrFEM++. LehrFEM++ is an open source finite element library currently developed at ETH Zurich [1]. Our implementation was partly inspired by the description of the general DPG framework Camellia in [2].

6.1 Shape functions

We provide a detailed description of the implementation of basis functions for the polynomial spaces introduced in section 2.3. Since LehrFEM++ relies heavily on parametric mapping techniques, this task is reduced to the specification of local shape functions on a reference element. In particular we have to provide evaluation methods and specify a set of local interpolation nodes to which the shape functions are associated. These interpolation nodes can either be associated to the reference element itself, one of its edges or one of its vertices. This implicitly defines a set of global shape functions, which result from "glueing" together local shape functions associated to the same global interpolation nodes.

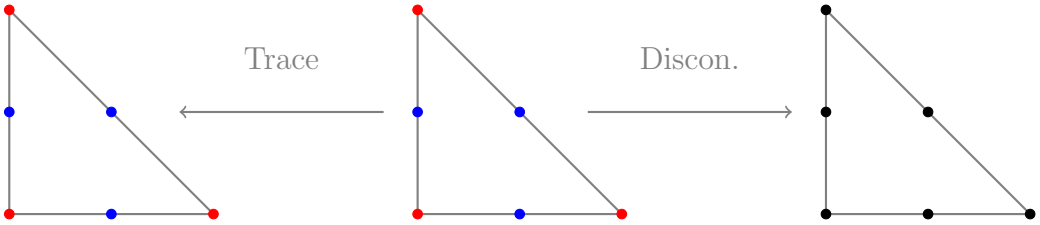
6.1.1 Continuous polynomials ($\mathcal{P}_p^0(\mathcal{M})$)

A first order implementation of Lagrangian finite elements is already provided in LehrFEM++ in the class `FeLagrangeO1Tria`[1]. We provide an implementation of second and third order Lagrangian finite elements in the classes `FeLagrangeO2Tria` and `FeLagrangeO3Tria`. We use the interpolation nodes shown in figure 1. The shape functions are computed such that they satisfy the cardinal basis property on these interpolation nodes. Explicit formulas for the quadratic case can be found in [14, Section 2.6.1]. The formulas in the third order case are similar. Furthermore, one can show, that by this choice of local shape functions, the resulting global shape functions are indeed continuous [14, Section 2.6.1].

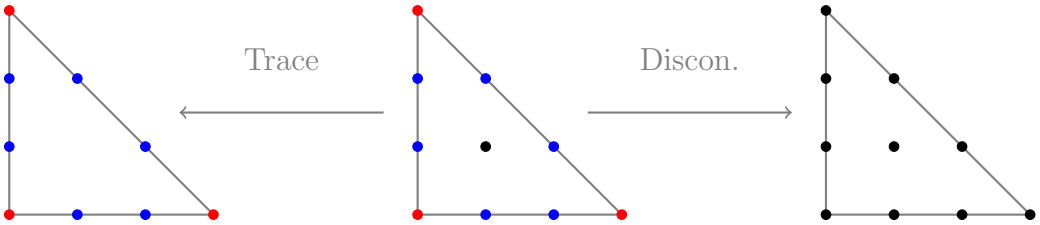
6.1.2 Discontinuous polynomials ($\mathcal{P}_p(\mathcal{M})$)

The continuity of the global shape functions of $\mathcal{P}_p^0(\mathcal{M})$ is the result of an implicit glueing of shape functions associated to the same global interpolation

[1] We ommit the specification of namespaces and template arguments throughout this thesis for readability. All LehrFEM++ classes are located in the default namespace `lf`. The classes of our own implementation are located in the namespace `projects::dpg`



(a) Quadratic interpolation nodes



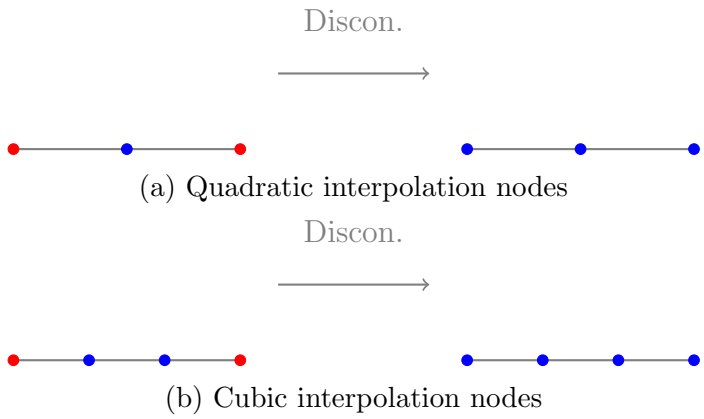
(b) Cubic interpolation nodes

Figure 1: Interpolation nodes on the reference triangle. Red nodes are associated to vertices, blue nodes to edges and black nodes to the triangle

node. Our approach with discontinuous polynomials is to avoid this glueing and treat all local shape functions as independent global shape functions, locally supported on exactly one element. This is achieved by the decorator pattern implemented in the class `DiscontinuousScalarReferenceFinite-Element`. This class decorates any `ScalarReferenceFiniteElement` and associates all its interpolation nodes to the underlying reference element, independent on where they lie. Figure 1 shows the effect of the decorator to the interpolation nodes of the second and third order Lagrangian shape functions. In the case of $p = 0$ we directly construct discontinuous shape functions in the class `FeDiscontinuousO0Tria`.

6.1.3 Trace polynomials ($\mathcal{P}_p^0(\mathcal{S})$)

In order to construct a basis for $\mathcal{P}_p^0(\mathcal{S})$, we simply use the traces of the local shape functions of $\mathcal{P}_p^0(\mathcal{M})$ on the element boundary. By the cardinal basis property, traces of functions associated to interior interpolation nodes are zero. In order to arrive at a basis for $\mathcal{P}_p^0(\mathcal{S})$, these shape functions have to be dropped. This is achieved by the decorator pattern implemented in the class `TraceScalarReferenceFiniteElement`. The class decorates any `ScalarReferenceFiniteElement` and drops all its interior shape functions and interpolation nodes. Figure 1 illustrates the effect of the decorator to the interpolation nodes of the second and third order Lagrangian shape functions.



(a) Quadratic interpolation nodes

(b) Cubic interpolation nodes

Figure 2: Interpolation nodes on the reference line segment. Red nodes are associated to vertices, blue nodes to the segment

6.1.4 Flux polynomials ($\mathcal{P}_p(\mathcal{S})$)

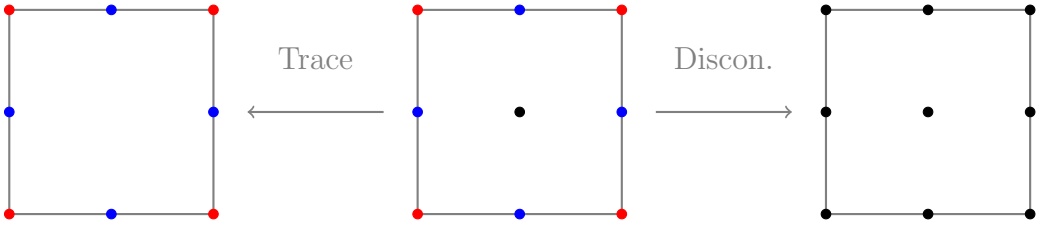
To construct a basis for $\mathcal{P}_p(\mathcal{S})$ we use an edge based approach. We proceed similarly to the construction of a basis for $\mathcal{P}_p(\mathcal{M})$. We first construct local shape functions for a basis of the space of continuous polynomials on the mesh skeleton in the classes `FeLagrangeO2Segment` and `FeLagrangeO3Segment`. We then apply the discontinuous decorator to arrive at a basis for $\mathcal{P}_p(\mathcal{S})$. Figure 2 shows the corresponding interpolation nodes and the effect of applying the discontinuous decorator. In the case of $p = 0$ we directly construct discontinuous shape functions in the class `FeDiscontinuousO0Segment`.

6.1.5 Quadrilateral meshes

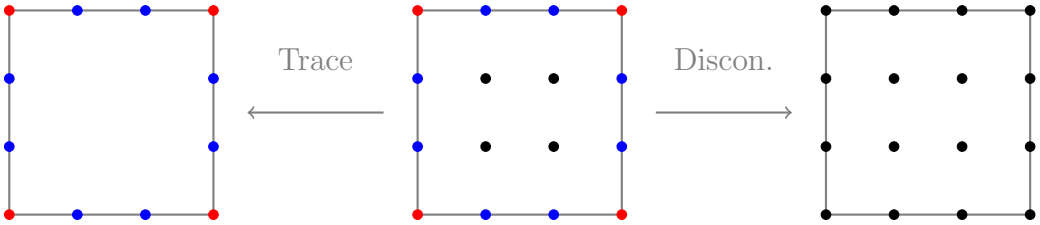
The construction in the case of quadrilateral elements is similar. A first order implementation of Lagrangian finite elements on quadrilaterals is provided by LehrFEM++ in the class `FeLagrangeQuadO1` and we provide an implementation for the second and third order bases in `FeLagrangeQuadO2` and `FeLagrangeQuadO3`. Figure 3 shows the interpolation nodes on the reference quadrilateral as well as the effect of the two decorators.

6.2 Local computations

We give a description of the approach used to evaluate local quantities such as the DPG element matrices and vectors.



(a) Quadratic interpolation nodes



(b) Cubic interpolation nodes

Figure 3: Interpolation nodes on the reference quadrilateral. Red nodes are associated to vertices, blue nodes to edges and black nodes to the quadrilateral.

6.2.1 Sub element matrices

We provide a set of `SubElementMatrixProviders` that allow the evaluation of the sub matrices, introduced in the discussion of the extended element stiffness matrices $\boldsymbol{B}^K$ and the local Gram matrices $\boldsymbol{G}^K$ in section 4.4 and 5.4. In particular they allow the evaluation of element matrices for the following bilinear forms.

$$(u, v) \to \int_K \mathcal{L}_1(u)\boldsymbol{\alpha}(x)\mathcal{L}_2(v)\mathrm{dx}$$
$$(\hat{u}, v) \to \int_{\partial K} \hat{u}v\,\boldsymbol{n}_K \cdot \boldsymbol{\alpha}(x)\mathrm{dS}$$
$$(\hat{q}_n, v) \to \int_{\partial K} \mathrm{sgn}_K \hat{q}_n v\alpha(x)\mathrm{dS}$$

The first bilinear form deals with all sub matrices that involve integration over an element K. $\mathcal{L}_{1,2}$ are first order linear differential operators and $\boldsymbol{\alpha}(x)$ is a coefficient function of suitable dimensions. Our current implementation supports the operators $\mathcal{L}_{1,2}(u) \in \{u, \boldsymbol{\nabla} u, \frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}\}$ by a suitable choice of parameters in the classes `DiffusionElementMatrixProvider`, `Reaction-ElementMatrixProvider` and `ConvectionElementMatrixProvider`.

The second bilinear form deals with sub matrices that are associated to trace variables and is implemented in the class `TraceElementMatrix-`

`Provider`. $\boldsymbol{\alpha}(x)$ is a vector-valued coefficient function.

Finally the third bilinear form is used for the evaluation of sub matrices associated to flux variables. Here the coefficient function $\alpha(x)$ is scalar valued. The corresponding provider is implemented in the class `FluxElementMatrixProvider`.

We provide similar functionality for the evaluation of the sub vectors of the extended element load vector $\boldsymbol{l}^K$. Here the only linear form of interest is

$$v \to \int_K v f \mathrm{dx}$$

where f is the scalar valued source function. This functionality is implemented in the class `LoadElementVectorProvider`.

In all implementations, the computation of the integrals is based on quadrature-based parametric mapping techniques [14, Section 2.8.3].

6.2.2 Product element matrices

The class `ProductElementMatrixProvider` allows the construction of more complex element matrices, using sub matrices provided by `SubElementMatrixProviders`.

For two spaces U and V with the following product structure

$$U = U_1 \times U_2 \times \cdots \times U_n$$
$$V = V_1 \times V_2 \times \cdots \times V_m$$

the class allows the evaluation of element matrices for bilinear forms $b : U \times V \to \mathbb{R}$ that are given as

$$b\left((u_1, u_2, \ldots u_n), (v_1, v_2, \ldots v_m)\right) = \sum_{k=1}^{N} b_k(u_{i_k}, v_{j_k})$$

where $i_k \in \{1, \ldots, n\}$, $j_k \in \{1, \ldots m\}$ and each $b_k : U_{i_k} \times V_{j_k} \to \mathbb{R}$ is a bilinear form between two components of the spaces U and V. The class evaluates the element matrices for the bilinear forms b_k that are provided by `SubElementMatrixProviders` and stacks them together to the element matrix of the bilinear form b. In particular the class can evaluate the extended element stiffness matrices $\boldsymbol{B}^K$ and the local Gramian matrices $\boldsymbol{G}^K$ of the primal and ultraweak DPG method.

We provide similar functionality to construct more complex element vectors, based on `SubElementVectorProviders` in the class `ProductElementVectorProvider`. This class allows the evaluation of an element vector for a

linear form $\ell : V \to \mathbb{R}$ that has the following structure

$$\ell\left((v_1, v_2, \dots v_m)\right) = \sum_{k=1}^{M} \ell_k(v_{j_k})$$

$j_k \in \{1, \dots m\}$ and each $\ell_k : V_{j_k} \to \mathbb{R}$ is a linear form of a component of the space V. In particular the class can evaluate the extended element load vector $\boldsymbol{l}^K$ of the primal and ultraweak DPG method.

6.2.3 DPG element matrices

The classes `DpgElementMatrixProvider` and `DpgElementVectorProvider` deal with the evaluation of the DPG element stiffness matrices and DPG element load vectors. They require a specification of the extended element stiffness matrix $\boldsymbol{B}^K$, local Gram matrix $\boldsymbol{G}^K$ and the extended load vector $\boldsymbol{l}^K$. They then use the formulas derived in section 3.3 to evaluate the DPG element stiffness matrix $\boldsymbol{A}^K$ and DPG element load vector $\boldsymbol{\phi}^K$ as

$$\boldsymbol{A}^K = \left(\boldsymbol{B}^K\right)^T \left(\boldsymbol{G}^K\right)^{-1} \boldsymbol{B}^K$$
$$\boldsymbol{\phi}^K = \left(\boldsymbol{B}^K\right)^T \left(\boldsymbol{G}^K\right)^{-1} \boldsymbol{l}^K$$

Standard assembly techniques are then used to construct the full stiffness matrix $\boldsymbol{A}$ and load vector $\boldsymbol{\phi}$ [5, Remark 3.1]

6.3 Boundary conditions

In the ultraweak formulation the boundary conditions can be transformed into essential conditions for two different variables: The flux variable $\hat{\sigma}_n$ and the trace variable $\hat{u}$. We use the offset function technique to enforce the essential boundary conditions. In particular we follow the approach in [14, Section 2.7.6] and approximate the offset function as a linear combination of relevant global shape functions. These consist of the global shape functions of $\hat{\sigma}_n$ associated to entities on Γ_1 and of the global shape functions of $\hat{u}$ associated to entities on Γ_2. The boundary conditions can then be introduced by a simple modification of the system matrix $\boldsymbol{A}$ and the load vector $\boldsymbol{\phi}$. We extend current LehrFEM++ functionality to deal with these essential boundary conditions in the function `InitEssentialConditionsFromFunctions`.

For the primal DPG method we proceed similarly by considering u as the trace-like variable and $\hat{q}_n$ as the flux variable.

6.4 DPG error estimator

Given a coefficient vector of a finite element function $\tilde{\boldsymbol{\mu}}$, the function `ElementErrorEstimators` evaluates the local DPG error estimators. First the local coefficient vector $\tilde{\boldsymbol{\mu}}^K$ is extracted and then the formulas derived in section 3.3 are used to compute the local error estimators $\tilde{\eta}_K$

$$\tilde{\boldsymbol{r}}^K = \boldsymbol{l}^K - \boldsymbol{B}^K \tilde{\boldsymbol{\mu}}^K$$
$$\tilde{\eta}_K^2 = \left(\tilde{\boldsymbol{r}}^K\right)^T \left(\boldsymbol{G}^K\right)^{-1} \tilde{\boldsymbol{r}}^K$$

Finally the global DPG error estimate $\tilde{\eta}$ can be computed from the local estimators by the function `EvalPosteriorError`.

(a) Triangular mesh (b) Quadrilateral mesh

Figure 4: Initial tensor product meshes on the unit square

7 Numerical experiments

In this section we present some numerical experiments. All computations were performed on the unit square domain $\Omega = (0,1)^2$. The examples use regular refinement on a tensor-product triangular or quadrilateral mesh. The corresponding initial meshes are shown in figure 4. In all examples we plot various error quantities with respect to $N = \dim(U_h)$, the number of degrees of freedom of the trial space. Since we consider 2D examples and regular refinement we have $N = \mathcal{O}(h^{-2})$. An algebraic convergence rate of order α in h will correspond to a line of slope $-\alpha/2$ in the corresponding log-log plot [3, Section 6].

7.1 Smooth solution

We consider two Dirichlet problems on the unit square with smooth exact solution $u(x,y) = \sin(\pi x)\sin(\pi y)$. The first problem is a pure Poisson equation (BVP 1). The second problem contains an additional constant advection field (BVP 2). The parameters of the two BVPs are set to

$$\text{BVP 1} \begin{cases} \epsilon_1 = 1 \\ \beta_1 = (0,0)^T \\ g_1 = 0 \end{cases}$$

$$\text{BVP 2} \begin{cases} \epsilon_2 = 1 \\ \beta_2 = (1,1)^T \\ g_2 = 0 \end{cases}$$

7.1.1 Primal formulation

We recall the polynomial approximation spaces for the primal formulation introduced in section 4. For given values of p and r we set

$$U_h = \mathcal{P}^0_{p,0}(\mathcal{M}) \times \mathcal{P}_{p-1}(\mathcal{S})$$
$$V^r = \mathcal{P}_r(\mathcal{M})$$

We examine the effect of different choices for the enrichment degree Δp and the influence of the additional advection field on the convergence behaviour of the primal DPG method. We consider $p = 1, 2$ and examine the two enrichment choices $\Delta p = 1, 2$ by setting $r = p + \Delta p$.

The error in the U norm is hard to compute and we only report the error in the $H^1(\Omega)$ norm. For a given approximation $(u_h, \hat{q}_{n,h})$ this is

$$e_h = \|u - u_h\|_{H^1(\Omega)}$$

Furthermore we report the DPG error estimator η of our approximation. Since the manufactured solution is smooth, we expect algebraic h-convergence of rate p by corollary 4.2.

Figure 5 reports our results on a triangular mesh. In all configurations we measured the predicted optimal convergence rates. As described in theorem 4.1, an enrichment degree of $\Delta p = 1$ was sufficient for the pure Poisson equation. Increasing the enrichment degree to $\Delta p = 2$, resulted in a qualitatively and quantitatively equivalent convergence behaviour. The additional convection term also had no effect on the convergence behaviour of the method. In particular an enrichment degree of $\Delta p = 1$ was enough for this example. In all cases the DPG error estimator provided an accurate estimate of the true error.

We perform analogous convergence studies on quadrilateral meshes and for given values of p and r consider the spaces [8, Section 5]

$$U_h = \mathcal{Q}^0_{p,0}(\mathcal{M}) \times \mathcal{P}_{p-1}(\mathcal{S})$$
$$V^r = \mathcal{Q}_r(\mathcal{M})$$

Figure 6 shows our results on quadrilateral meshes. Again we observed the optimal convergence rates and the convergence behaviour remained qualitatively equivalent to the results on triangular meshes.

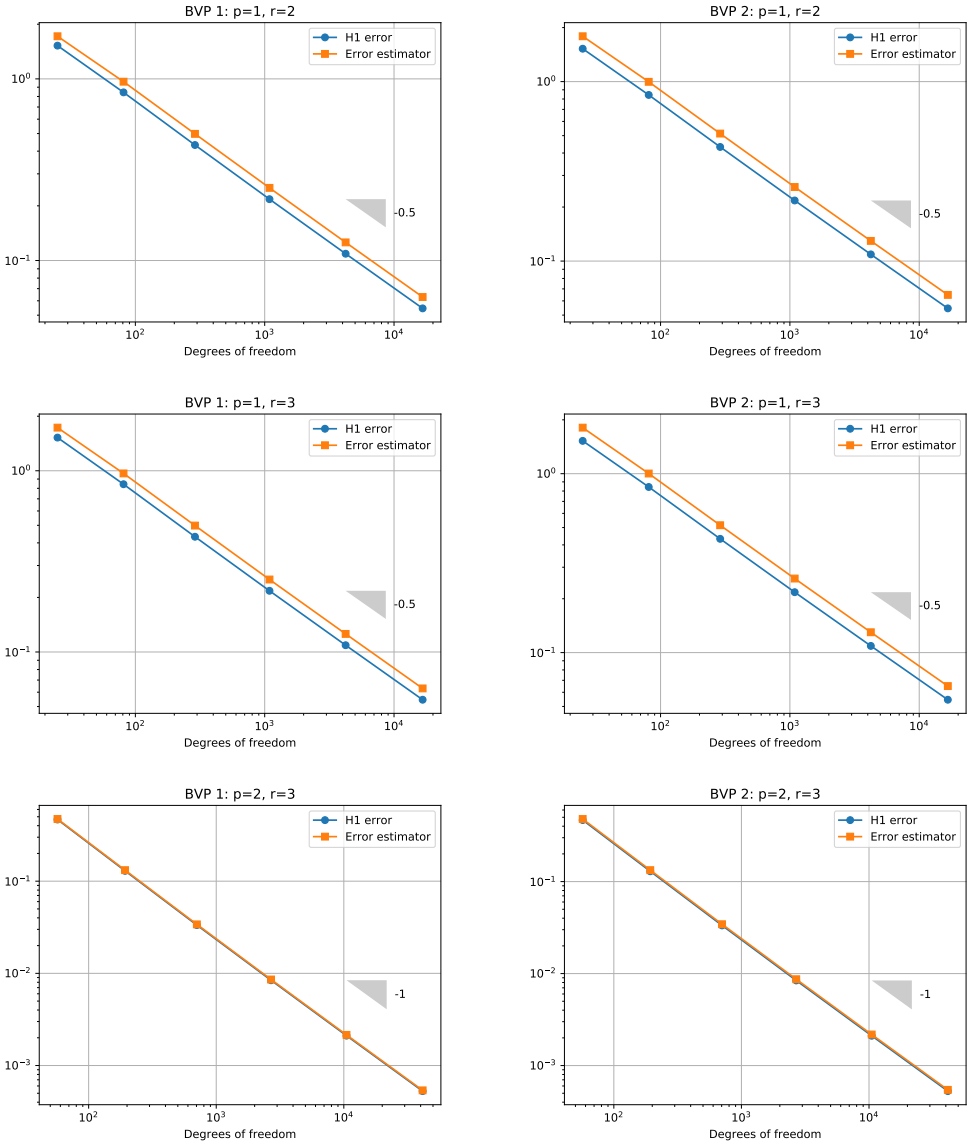


Figure 5: Primal DPG method for BVP1 and BVP2 on a triangular mesh

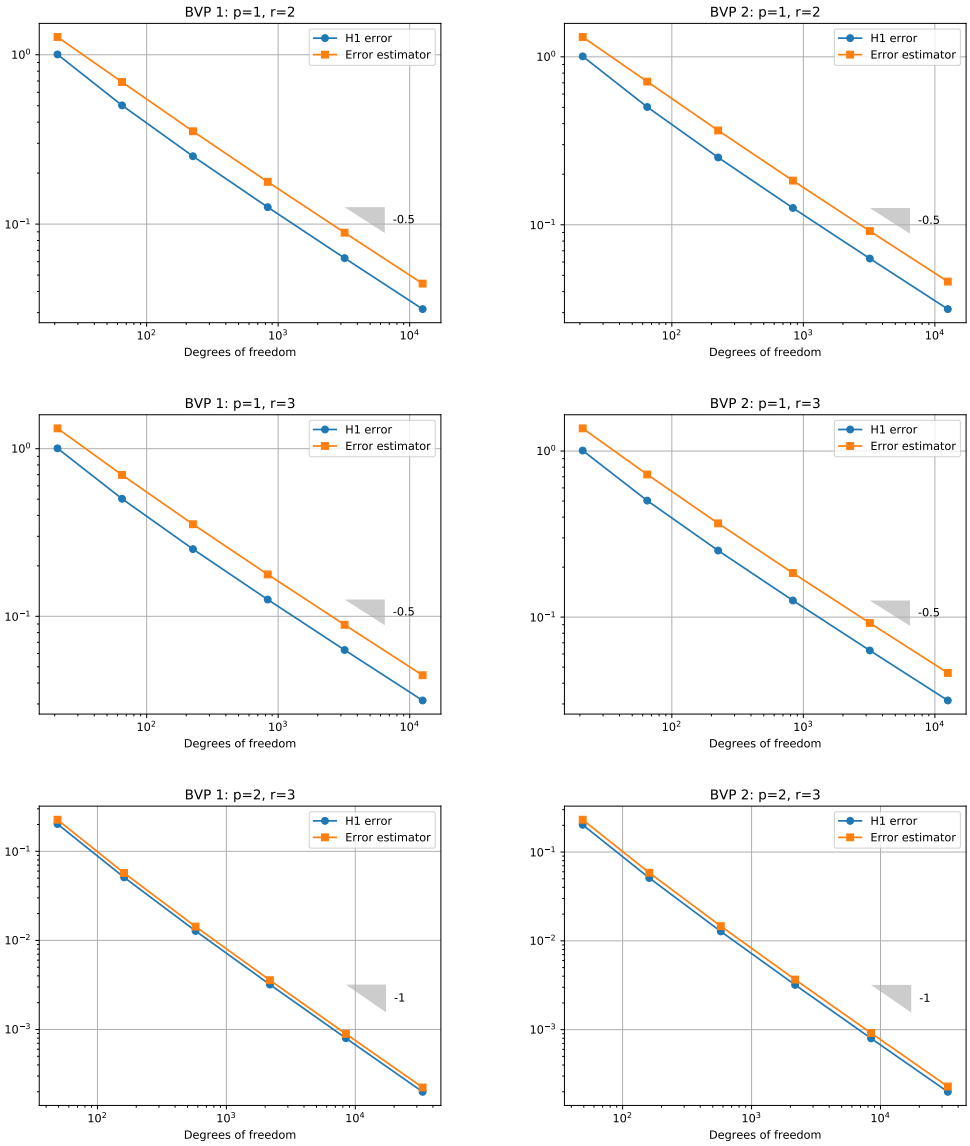


Figure 6: Primal DPG method for BVP 1 and BVP 2 on a quadrilateral mesh.

7.1.2 Ultraweak formulation

We recall the polynomial approximation spaces of the ultraweak formulation introduced in section 5. For given values of p and r we set

$$U_h = \mathcal{P}_p(\mathcal{M}) \times \boldsymbol{\mathcal{P}}_p(\mathcal{M}) \times \mathcal{P}^0_{p+1,0}(\mathcal{S}) \times \mathcal{P}_p(\mathcal{S})$$
$$V^r = \mathcal{P}_r(\mathcal{M}) \times \boldsymbol{\mathcal{P}}_r(\mathcal{M})$$

We again study the effect of the additional convection term on the convergence behaviour. By theorem 5.1 our standard enrichment choice is $\Delta p = 2$ and we set $r = p + 2$. We consider the cases $p = 0, 1$. The error in the U norm is hard to compute and we define the following error quantities [3, Section 6].

$$\mathrm{err}(u_h) = \|u - u_h\|_{L^2(\Omega)}$$
$$\mathrm{err}(\boldsymbol{\sigma}_h) = \|\boldsymbol{\sigma} - \boldsymbol{\sigma}_h\|_{\boldsymbol{L}^2(\Omega)}$$

These are simply the L^2-errors in the u and $\boldsymbol{\sigma}$ components. We furthermore report η, the value of the DPG error estimator. Since our exact solution is smooth, we expect h-convergence of rate $p + 1$ by corollary 5.2.

Figure 7 reports our results on a triangular mesh. In all configurations we observed the predicted optimal convergence rates. The plots indicate, that the DPG error estimator is indeed accurate. The convergence behaviour between the two boundary value problems remained qualitatively comparable.

We perform analogous convergence studies on quadrilateral meshes. Following the recommendation in [10, Section 4] we approximate both components of $\boldsymbol{\tau} \in H(\mathrm{div}, \mathcal{M})$ by polynomials of equal degree. For given values of p and r we consider the spaces

$$U_h = \mathcal{Q}_p(\mathcal{M}) \times \boldsymbol{\mathcal{Q}}_p(\mathcal{M}) \times \mathcal{P}^0_{p+1,0}(\mathcal{S}) \times \mathcal{P}_p(\mathcal{S})$$
$$V^r = \mathcal{Q}_r(\mathcal{M}) \times \boldsymbol{\mathcal{Q}}_r(\mathcal{M})$$

As expected, the convergence behaviour shown in figure 8 remained qualitatively similar to the triangular case, and we observed the optimal convergence rates.

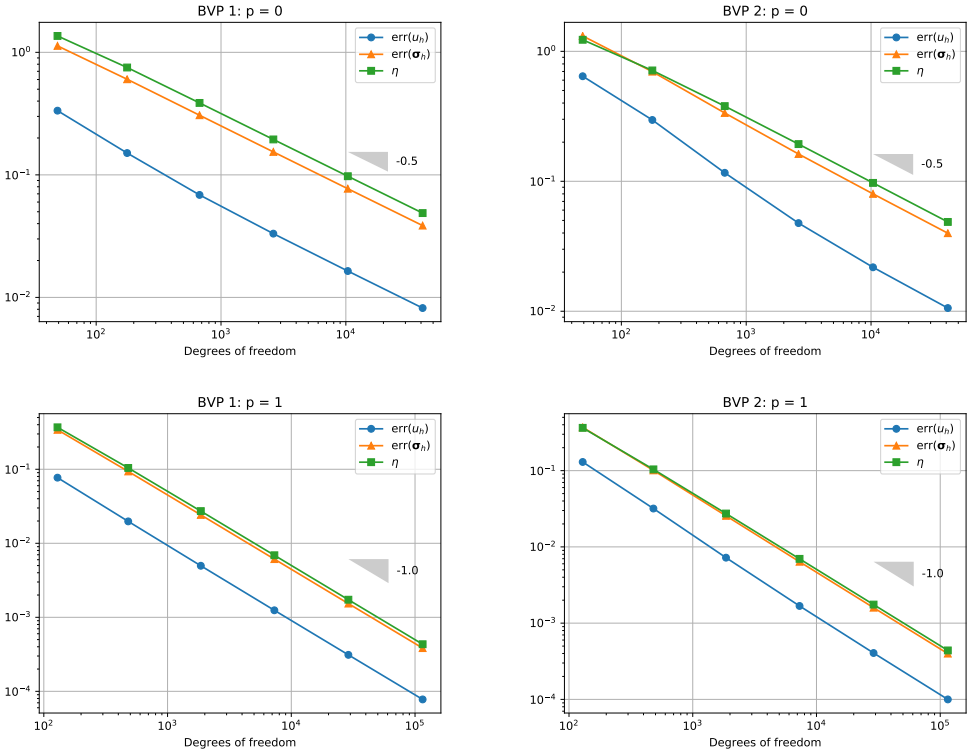


Figure 7: Ultraweak DPG method for BVP1 and BVP2 on a triangular mesh.

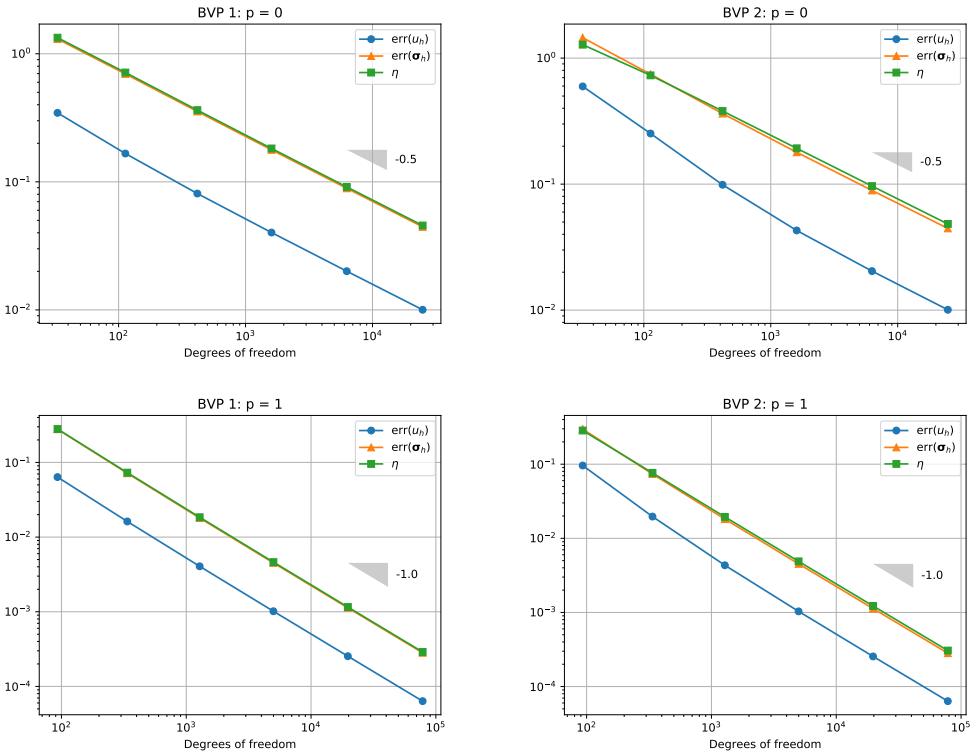


Figure 8: Ultraweak DPG method for BVP 1 and BVP 2 on a quadrilateral mesh.

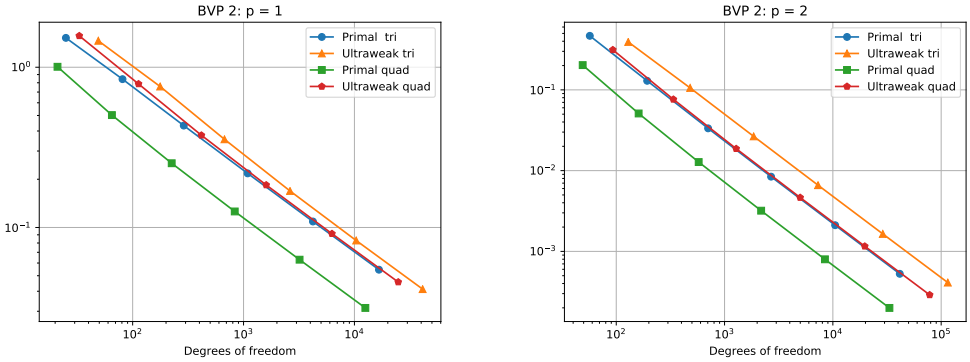


Figure 9: Quantitative comparison between the primal and ultraweak DPG method for BVP 2.

7.1.3 Quantitative comparison

We report some results, comparing the quantitative efficiency of the two methods. In order to achieve comparable convergence rates, we consider the primal DPG method of degree p and compare it to the ultraweak DPG method of degree $p-1$. For the primal formulation we report the H^1 norm

$$\mathrm{err}(u_h) = \|u - u_h\|_{H^1(\Omega)}$$

and for the ultraweak formulation we report a weak analogue given by

$$\mathrm{err}(u_h, \boldsymbol{\sigma_h}) = \sqrt{\|u - u_h\|^2_{L^2(\Omega)} + \|\boldsymbol{\sigma} - \boldsymbol{\sigma}_h\|^2_{\boldsymbol{L}^2(\Omega)}}$$

We limit our considerations to BVP 2 and chose $p = 1, 2$. Figure 9 shows our results. If we consider the number of degrees of freedom as a measure for the computational cost of our code, then this example shows:

1. The primal method performed generally better than the ultraweak method.

2. Both methods were more efficient on quadrilateral meshes than on triangular ones.

7.2 Boundary layer

For $\epsilon > 0$ and $\beta = (2, 1)^T \in \mathbb{R}^2$ we consider the exact solution [11, Section 5.1]

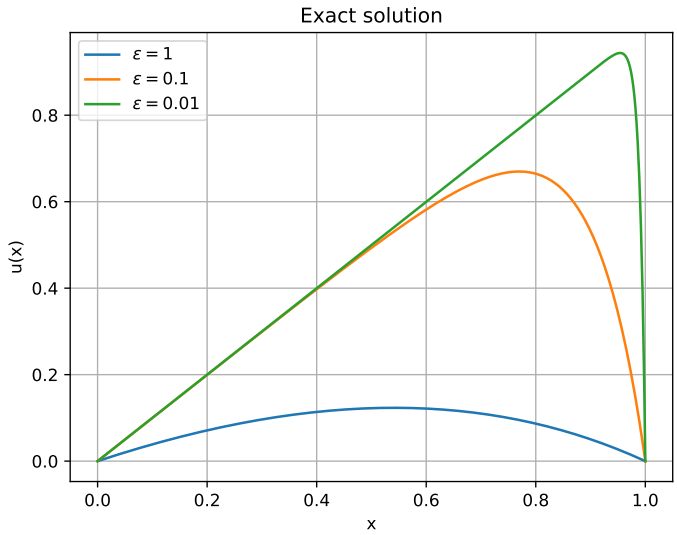


Figure 10: Boundary layer of the exact 1D solution for different values of ϵ

$$u(x,y) = \left(x + \frac{e^{\frac{\beta_1 x}{\epsilon}} - 1}{1 - e^{\frac{\beta_1}{\epsilon}}}\right)\left(y + \frac{e^{\frac{\beta_2 y}{\epsilon}} - 1}{1 - e^{\frac{\beta_2}{\epsilon}}}\right)$$

This function satisfies homogeneous Dirichlet boundary conditions on the unit square. Figure 10 shows the 1D analogue of u. For small values of ϵ this function develops a steep boundary layer along the top and right boundary of the unit square.

We analyse the convergence behaviour for different choices of the diffusion parameter ϵ. We restrict our analysis to a triangular tensor product mesh and compare the cases $\epsilon \in \{1, 0.5, 0.1, 0.05, 0.01\}$. For even smaller values of ϵ, our approach of using regular refinement is infeasible, since an adequate resolution of the boundary layer becomes computationally expensive. We consider both the primal and ultraweak DPG method and report the same error quantities as introduced in section 7.1.

Figure 11 shows our results in the case of the primal DPG method of degree $p = 2$ and figure 12 shows our results in the case of the ultraweak DPG formulation of degree $p = 1$.

In both cases for big values of ϵ, we observed optimal convergence rates similar to the smooth solution analysed in section 7.1. For smaller values of ϵ, we observed a region of pre-asymptotic noise, since the boundary layer could not be adequately resolved on the coarser meshes. As soon as the

mesh width was small enough, the boundary layer could be resolved and we approached optimal convergence rates. The behaviour of the two methods for a given value of ϵ was qualitatively similar.

Regarding the ultraweak formulation, the DPG error estimator was accurate on fine meshes for all values of ϵ. For the primal DPG method the estimator reported the true error scaled by a factor of around ϵ. We considered some different choices of test space inner products. In particular a choice of

$$(v, \delta v)_V = \sum_{K \in \mathcal{M}} \int_K v \delta v \mathrm{dx} + \epsilon^2 \int_K \boldsymbol{\nabla} v \cdot \boldsymbol{\nabla} \delta v \mathrm{dx}$$

resulted in the convergence behaviour shown in figure 13, where the DPG error estimator remained more accurate for small values of ϵ on fine meshes.

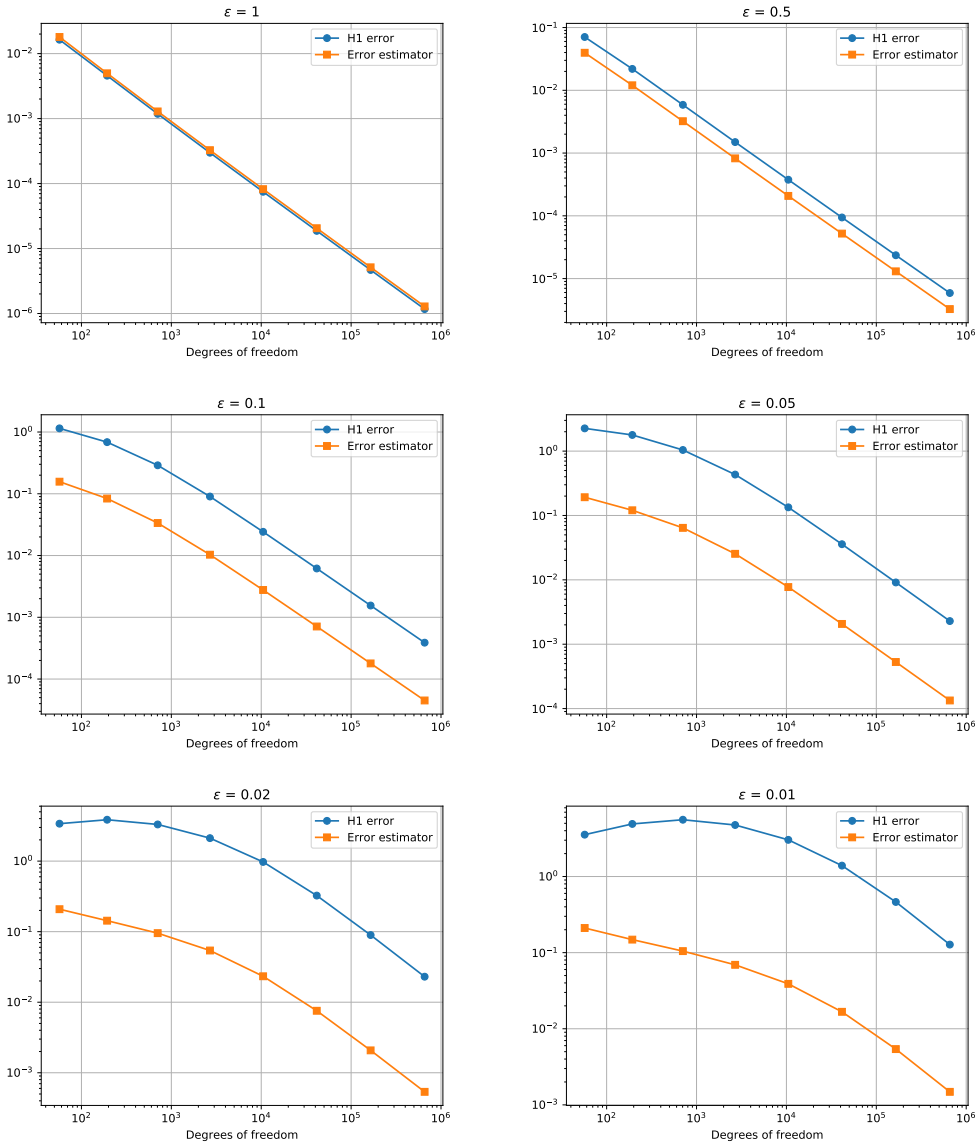


Figure 11: Primal DPG method for the BVP with boundary layer for different values of ϵ

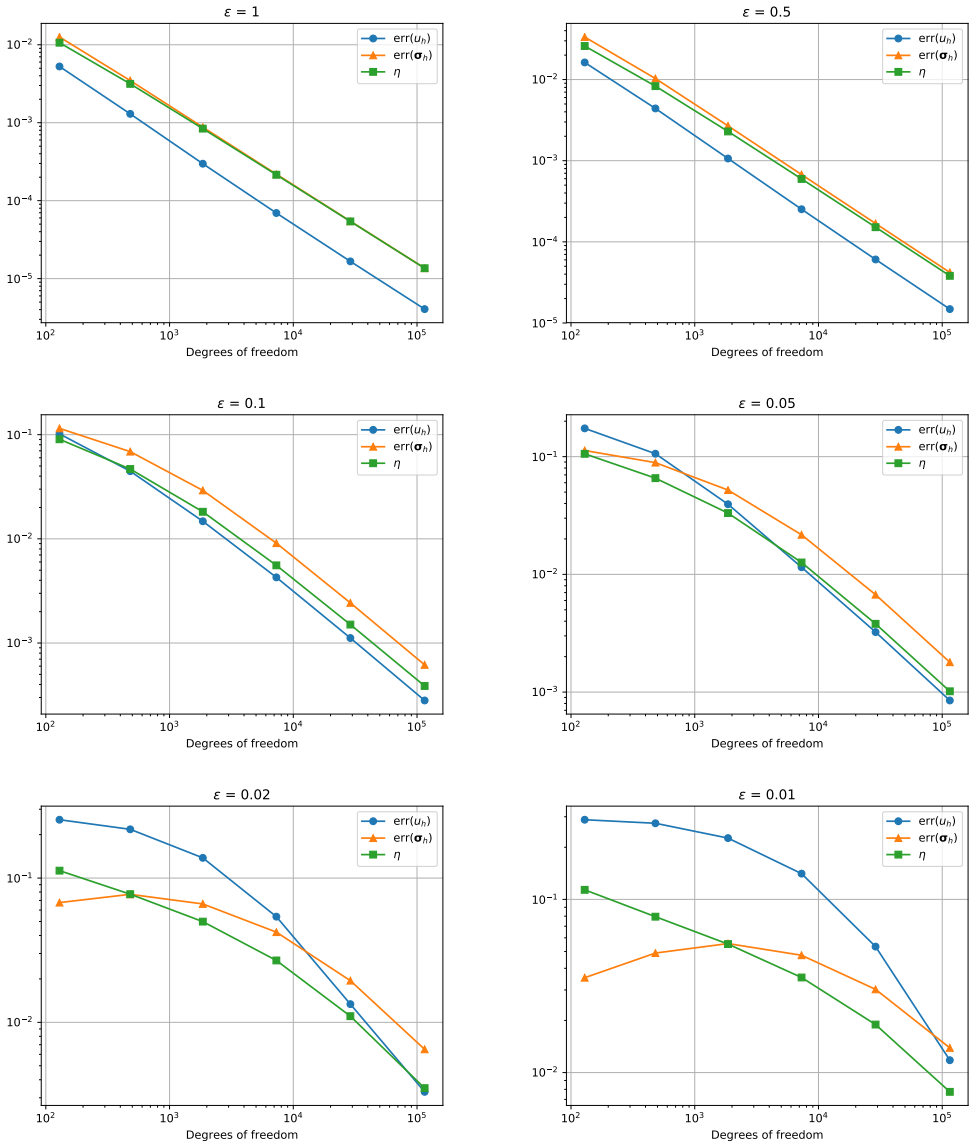


Figure 12: Ultraweak DPG method for the BVP with boundary layer for different values of ϵ

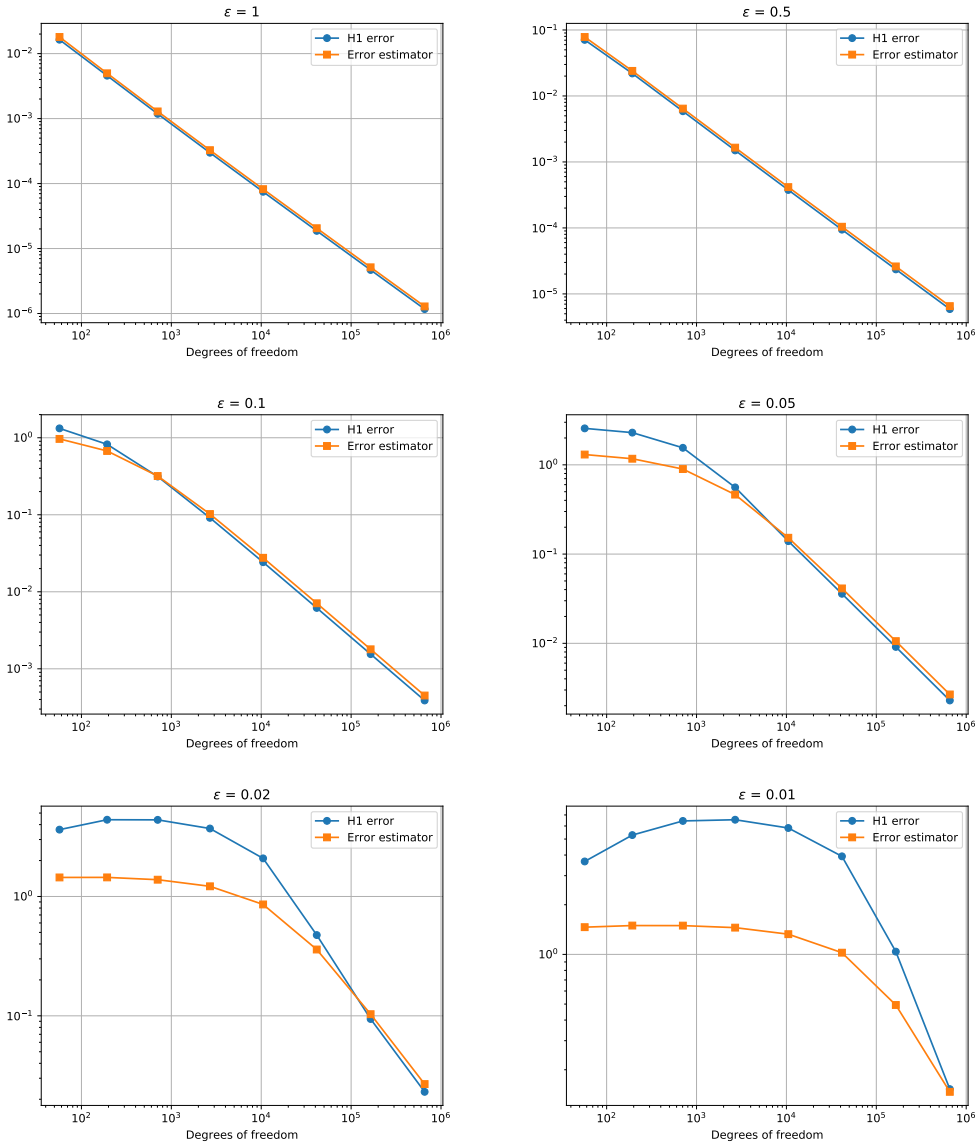


Figure 13: Primal DPG method with adapted test space inner product for the BVP with boundary layer for different values of ϵ

8 Conclusion

We derived the variational formulations for both the primal and ultraweak DPG method of the convection-diffsuion model probelm and presented the main convergence results. Both formulations were successfully implemented using the C++ library LehrFEM++. Our implementation was based on the local computation of DPG element stiffness matrices and DPG element load vectors. We performed numerical experiments and observed the optimal convergence rates predicted by the theory.

We discussed the convergence behaviour of our implementation for a solution that develops a steep boundary layer in section 7.2. In particular, we managed to get decent results for values of $\epsilon \geq 0.01$, but for smaller values of ϵ it became computationally very expensive to adequately resolve the boundary layer. An adaptive approach, based on the DPG error estimator, would be highly desirable in this case, to allow a fine resolution of the boundary layer without refining the whole mesh [10].

References

[1] "LehrFEM++." `https://github.com/craffael/lehrfempp`, 2019.

[2] N. V. Roberts, "Camellia: A software framework for discontinuous petrov–galerkin methods," *Computers and Mathematics with Applications*, vol. 68, no. 11, pp. 1581 – 1604, 2014. Minimum Residual and Least Squares Finite Element Methods.

[3] T. Führer, "Superconvergence in a dpg method for an ultra-weak formulation," *Computers and Mathematics with Applications*, vol. 75, no. 5, pp. 1705 – 1718, 2018.

[4] N. Heuer, M. Karkulik, and F.-J. Sayas, "Note on discontinuous trace approximation in the practical dpg method," *Computers Mathematics with Applications*, vol. 68, no. 11, pp. 1562 – 1568, 2014. Minimum Residual and Least Squares Finite Element Methods.

[5] L. F. Demkowicz and J. Gopalakrishnan, *An Overview of the Discontinuous Petrov Galerkin Method*, pp. 149–180. Cham: Springer International Publishing, 2014.

[6] C. Carstensen, L. Demkowicz, and J. Gopalakrishnan, "Breaking spaces and forms for the dpg method and applications including maxwell equations," *Computers and Mathematics with Applications*, vol. 72, no. 3, pp. 494 – 522, 2016.

[7] J. Gopalakrishnan and W. Qiu, "An analysis of the practical dpg method," *Math. Comput.*, vol. 83, pp. 537–552, 2014.

[8] L. Demkowicz and J. Gopalakrishnan, "A primal dpg method without a first-order reformulation," *Computers and Mathematics with Applications*, vol. 66, no. 6, pp. 1058 – 1064, 2013.

[9] L. Demkowicz and J. Gopalakrishnan, "Analysis of the dpg method for the poisson equation," *SIAM Journal on Numerical Analysis*, vol. 49, no. 5, pp. 1788–1809, 2011.

[10] L. Demkowicz, J. Gopalakrishnan, and A. H. Niemi, "A class of discontinuous petrov–galerkin methods. part iii: Adaptivity," *Applied Numerical Mathematics*, vol. 62, no. 4, pp. 396 – 427, 2012.

[11] L. Demkowicz and J. Gopalakrishnan, "A class of discontinuous petrov-galerkin methods. part ii: Optimal test functions," *Numerical Methods for Partial Differential Equations*, vol. 27, no. 1, pp. 70–105, 2011.

[12] S. Nagaraj, S. Petrides, and L. F. Demkowicz, "Construction of dpg fortin operators for second order problems," *Computers Mathematics with Applications*, vol. 74, no. 8, pp. 1964 – 1980, 2017.

[13] J. Chan, N. Heuer, T. Bui-Thanh, and L. Demkowicz, "A robust dpg method for convection-dominated diffusion problems ii: Adjoint boundary conditions and mesh-dependent test norms," *Computers Mathematics with Applications*, vol. 67, no. 4, pp. 771 – 795, 2014. High-order Finite Element Approximation for Partial Differential Equations.

[14] R. Hiptmair, "Numerical methods for partial differential equations," 2019. `https://www.sam.math.ethz.ch/~grsam/NUMPDEFL/NUMPDE19.pdf`.