Weierstraß-Institut
für Angewandte Analysis und Stochastik

Leibniz-Institut im Forschungsverbund Berlin e. V.

Preprint ISSN 2198-5855

On reducing spurious oscillations in discontinuous Galerkin (DG) methods for steady-state convection-diffusion-reaction equations

Derk Frerichs[1], Volker John[1,2]

submitted: October 5, 2020

[1] Weierstrass Institute
Mohrenstr. 39
10117 Berlin
Germany
E-Mail: derk.frerichs@wias-berlin.de
volker.john@wias-berlin.de

[2] Freie Universität Berlin
Department of Mathematics and Computer Science
Arnimallee 6
14195 Berlin
Germany

No. 2769
Berlin 2020

2010 *Mathematics Subject Classification.* 65N30.

Key words and phrases. Steady-state convection-diffusion-reaction equations, convection-dominated regime, discontinuous Galerkin finite element method, reduction of spurious oscillations, post-processing approaches, slope limiters.

Edited by
Weierstraß-Institut für Angewandte Analysis und Stochastik (WIAS)
Leibniz-Institut im Forschungsverbund Berlin e. V.
Mohrenstraße 39
10117 Berlin
Germany

Fax: +49 30 20372-303
E-Mail: preprint@wias-berlin.de
World Wide Web: http://www.wias-berlin.de/

On reducing spurious oscillations in discontinuous Galerkin (DG) methods for steady-state convection-diffusion-reaction equations

Derk Frerichs, Volker John

Abstract

A standard discontinuous Galerkin (DG) finite element method for discretizing steady-state convection-diffusion-reaction equations is known to be stable and to compute sharp layers in the convection-dominated regime, but also to show large spurious oscillations. This paper studies post-processing methods for reducing the spurious oscillations, which replace the DG solution in a vicinity of layers by a constant or linear approximation. Three methods from the literature are considered and several generalizations and modifications are proposed. Numerical studies with the post-processing methods are performed at two-dimensional examples.

1 Introduction

Convection-diffusion-reaction equations model the transport of a physical species like temperature (energy balance) or concentration (mass balance). In practice, the convective transport by the velocity field is usually much stronger than the diffusive transport. This situation is called convection-dominated regime. Characteristic features of the solution of a convection-diffusion-reaction equation in this regime are layers, which are thin regions where the gradient of the solution possesses a very large norm. These small spatial scales are present for both, solutions of time-dependent and steady-state convection-diffusion-reaction equations. Since solutions of the time-dependent equation often do not possess small scales with respect to time, such that the major feature are the small scales with respect to space, we will concentrate here on the discussion of the steady-state problem.

The steady-state convection-diffusion-reaction equation is given by

$$\begin{aligned} -\varepsilon\Delta u + \boldsymbol{b}\cdot\nabla u + cu &= f \quad \text{in } \Omega, \\ u &= g \quad \text{on } \Gamma_{\mathrm{D}}, \\ \varepsilon\nabla u\cdot\boldsymbol{n} &= 0 \quad \text{on } \Gamma_{\mathrm{N}}, \end{aligned} \tag{1}$$

where $\Omega \subset \mathbb{R}^d$, $d \in \{2,3\}$, is a bounded domain with polyhedral Lipschitz boundary $\Gamma = \Gamma_{\mathrm{D}} \cup \Gamma_{\mathrm{N}}$ with $\Gamma_{\mathrm{D}} \cap \Gamma_{\mathrm{N}} = \emptyset$. The coefficient $\varepsilon \in \mathbb{R}$, $\varepsilon > 0$, is the diffusion coefficient, $\boldsymbol{b}$ is the convection field, c the reaction coefficient, and f models sources. The prescribed boundary conditions on the Dirichlet boundary Γ_{D} are denoted by g and $\boldsymbol{n}$ is the outward pointing unit normal vector on the boundary of Ω. From the physical point of view, one has to prescribe Dirichlet boundary conditions at the inflow boundary, i.e., $\Gamma_{-} = \{\boldsymbol{x} \in \Gamma \,:\, \boldsymbol{b}(\boldsymbol{x})\cdot\boldsymbol{n}(\boldsymbol{x}) < 0\} \subset \Gamma_{\mathrm{D}}$. Mathematically, the convection-dominated regime is described by $\varepsilon \ll L\|\boldsymbol{b}\|_{L^\infty(\Omega)}$, where L is a characteristic length scale. Note that this inequality is correct with respect to the physical units.

Many discretizations are based on an underlying mesh. In the convection-dominated regime, the size of the layers is much smaller than the affordable mesh width. It is well known that standard discretizations, like the central finite difference method or the Galerkin finite element method cannot cope with

DOI 10.20347/WIAS.PREPRINT.2769 Berlin 2020

this situation. They try to compute all important scales of the solution, which is not possible since the layers cannot even be represented on affordable grids. Numerical solutions computed with these schemes are globally polluted by spurious oscillations, i.e., unphysical values. One has to introduce some stabilizing component in the discretization, leading to so-called stabilized discretizations. A survey on stabilized methods, in particular in the framework of finite element methods, can be found in the monograph [29]. Since the publication of this monograph, several methods and their numerical analysis have been developed further, e.g., the analysis of algebraic flux correction (AFC) schemes, e.g., see [3].

Screening the literature on finite element methods for steady-state convection-diffusion-reaction equations, one finds that by far most publications are for conforming finite elements with Lagrangian basis functions, often of first order. But even for stabilized methods using such basis functions, there are a number of important unresolved questions, which are formulated in [23]. The most popular stabilization for conforming finite elements is the SUGP (streamline-upwind Petrov–Galerkin) or streamline-diffusion method from [20, 5].

For discontinuous Galerkin (DG) methods, one can find, in comparison with conforming finite element methods, only rather few contributions for convection-diffusion equations. The first proposal of using discontinuous finite element functions, for first order hyperbolic problems, dates back to [27]. During the last decades, DG approaches gained also popularity for discretizing second order elliptic equations, e.g., see the monographs [28, 11, 12]. Big advantages of DG methods, in comparison with conforming finite element methods, are that they allow comparatively easily to use hp-adaptivity, e.g., see [15], even for polygonal or polyhedral meshes, [7]. Concerning convection-diffusion-reaction equations, error analysis can be found in [19, 17, 6, 25, 2, 11], which will be discussed in some detail at the end of Section 2. In the competitive numerical study [1], a DG method was included. On the one hand, this method computed numerical solutions with very sharp layers. But on the other hand, the solutions possessed very large over- and undershoots in a vicinity of layers.

The goal of this paper consists in studying approaches for reducing these spurious oscillations. Of course, the ideal situation in practice is a numerical solution without such oscillations, but often small spurious oscillations can be tolerated. For conforming finite elements, there are many proposals of methods for reducing spurious oscillations. A large number of these methods takes the SUPG method as basic stabilized discretization and then adds an additional term to reduce the spurious oscillations of the SUPG method, e.g., see [22] for a survey of these so-called SOLD (spurious oscillations at layers diminishing) methods and [30] for a more recent proposal. Usually, SOLD methods are nonlinear and they are used for lowest order finite elements. Another idea consists in optimizing the stabilization parameter of the SUPG method in order to reduce spurious oscillations, e.g., see [24, 26]. For DG methods, the numerical analysis in [17, 6, 25, 2, 11] shows that there is a control of the error of the streamline derivative without introducing a stabilization term of streamline-diffusion type. This situation is of advantage in practice since there is no need to choose a stabilization parameter. Because of this advantage, we will not consider the SUPG stabilization for DG methods, although it is possible to utilize it, e.g., see [2, Remark 3.1]. Thus, the optimization of a stabilization parameter is not possible. There would be still the way of adding terms like in SOLD methods, but we decided not to pursue this approach for the following reasons. First, we liked to consider only linear methods. It has been observed for many SOLD methods that the solution of the nonlinear problems requires often many iterations and it is time-consuming, e.g., see [22]. And second, we liked to study also methods with higher polynomial degree. A high polynomial degree is a good choice away from layers, but in a vicinity of layers a low polynomial degree is more appropriate, since Sobolev norms of the analytic solution at layers, which have an impact on the error, scale with inverse powers of the diffusion parameter and the power increases with the order. For these reasons, post-processing approaches will be considered

that replace in a vicinity of layers the higher order polynomial by a low order one, whose definition utilizes a slope limiter. Such methods were proposed for constant replacements in [13, 14] and for (at most) linear replacements in [9, 28]. Note that such an easy local change of the polynomial degree is not possible for conforming finite elements. A main contribution of this paper consists in presenting some generalizations and modifications of the post-processing methods. A second main contribution is the first step of a systematic numerical investigation of these methods for steady-state convection-diffusion-reaction equations.

The paper is organized as follows. Section 2 introduces the DG method that is studied. The approaches for reducing spurious oscillations are described in Section 3. Numerical studies of these approaches at two standard problems are presented in Section 4. Finally, a summary and an outlook are provided.

2 DG Methods for Convection-Diffusion-Reaction Equations

Throughout the paper, standard notations will be used for Lebesgue and Sobolev spaces and their norms. A norm of a space X is denoted by $\|\cdot\|_X$, a seminorm by $|\cdot|_X$, and the inner product in $L^2(\Omega)$ is denoted by $(\cdot,\cdot)$.

Starting point of a DG method is the weak formulation of (1): Find $u \in H^1_{\mathrm{D},g}(\Omega)$ such that

$$(\varepsilon\nabla u, \nabla v) + (\boldsymbol{b}\cdot\nabla u + cu, v) = (f, v) \quad \forall\, v \in H^1_{\mathrm{D},0}(\Omega), \tag{2}$$

where

$$H_{\mathrm{D},g} = \left\{ v \in H^1(\Omega) \,:\, v|_{\Gamma_\mathrm{D}} = g \right\}, \quad H_{\mathrm{D},0} = \left\{ v \in H^1(\Omega) \,:\, v|_{\Gamma_\mathrm{D}} = 0 \right\}.$$

The Lax–Milgram theorem shows that under the conditions

$$(\mu(x))^2 = c - \frac{1}{2}\nabla\cdot\boldsymbol{b} \geq \mu_0 > 0, \quad \Gamma_\mathrm{D} \neq \emptyset, \quad \boldsymbol{b}\cdot\boldsymbol{n} \geq 0 \text{ on } \Gamma_\mathrm{N},$$

problem (2) possesses a unique solution, e.g., see [29, Section III.1.1].

Let $\mathcal{T}_h$ be a decomposition of $\overline{\Omega}$ into simplicial or quadrilateral/hexahedral mesh cells $\{K\}$ with pairwise disjoint interiors such that $\overline{\Omega} = \cup_{K\in\mathcal{T}_h} K$. The triangulation should be admissible in the usual sense, see the definition in [8, p. 38, p. 51]. Regular families of triangulations will be considered, e.g., see [4, Def. 4.4.13], such that each (open) facet of a mesh cell which lies on Γ is either contained in Γ_D or in Γ_N. The area of a mesh cell K is denoted by $|K|$, its diameter by h_K, and it is $h := \max_{K\in\mathcal{T}_h} h_K$. For each mesh cell $K \in \mathcal{T}_h$, the set of all facets $E \subset \partial K$ is denoted by $\mathcal{E}_h(K)$. Then, the set of all facets is $\mathcal{E}_h := \cup_{K\in\mathcal{T}_h}\mathcal{E}_h(K)$, such that $\mathcal{E}_h = \mathcal{E}_h^\mathrm{I} \cup \partial\mathcal{E}_h$, where $\mathcal{E}_h^\mathrm{I}$ denotes the set of all interior facets and $\partial\mathcal{E}_h := \mathcal{E}_h \cap \partial\Omega$ the set of facets on the boundary. Furthermore, the set of the facets on the Dirichlet boundary is denoted by $\mathcal{E}_h^\mathrm{D} := \Gamma_\mathrm{D} \cap \mathcal{E}_h$ and the notation $\mathcal{E}_h^\mathrm{ID} := \mathcal{E}_h^\mathrm{I} \cup \mathcal{E}_h^\mathrm{D}$ is used. The inflow boundary edges are denoted by $\mathcal{E}_h^- := \Gamma_- \cap \mathcal{E}_h$. Let h_E be the diameter of a facet E. Because of the regularity of the families of triangulations, there exists a constant $C > 0$ such that for all $\mathcal{T}_h$ and all $K \in \mathcal{T}_h$ it holds that $h_E \leq h_K \leq C h_E$.

Two mesh cells $K_i, K_j \in \mathcal{T}_h$ are called neighbors along a facet $E \in \mathcal{E}_h$ if $E = K_i \cap K_j$. For $K \in \mathcal{T}_h$, let $\boldsymbol{n}_K$ denote the unit outer normal vector on ∂K. Given a fixed numbering of the mesh cells $K_0, K_1, K_2, \ldots$, the unit normal vector $\boldsymbol{n}_E$ on a facet $E \in \mathcal{E}_h$ is defined as follows

$$\boldsymbol{n}_E := \begin{cases} \boldsymbol{n}_K, & \text{if } E \in \partial\mathcal{E}_h \cap \mathcal{E}_h(K) \text{ for a } K \in \mathcal{T}_h, \\ \boldsymbol{n}_{K_i}, & \text{if } K_i \text{ and } K_j \text{ are neighbors along facet } E \text{ and } i < j. \end{cases}$$

The space of polynomials of at most degree r on simplicial mesh cells K is denoted by $P_r(K)$ and the space of tensor products of polynomials of degree at most r in each coordinate direction on quadrilateral/hexahedral mesh cells by $Q_r(K)$. For DG methods, the broken Sobolev space

$$H^s(\mathcal{T}_h) = \left\{ v \in L^2(\Omega) \,:\, v|_K \in H^s(K) \text{ for any } K \in \mathcal{T}_h \right\} \supset H^s(\Omega), \quad s \geq 0,$$

together with the norm and seminorm

$$\|v\|^2_{H^s(\mathcal{T}_h)} := \sum_{K \in \mathcal{T}_h} \|v\|_{H^s(K)}, \quad |v|^2_{H^s(\mathcal{T}_h)} := \sum_{K \in \mathcal{T}_h} |v|_{H^s(K)}$$

is defined. Then, the finite element space with $\mathcal{R}$ either P (simplices) or Q (quadrilaterals/hexahedra) is given by

$$\mathcal{R}_{h,r} := \left\{ v_h \in L^2(\Omega) \,:\, v_h|_K \in \mathcal{R}_r(K) \text{ for any } K \in \mathcal{T}_h \right\} \subset H^s(\mathcal{T}_h).$$

This space contains functions that are discontinuous along interior facets. The jump along a facet E, whose sign depends on the numbering of the mesh cells, is defined by

$$[\![v]\!]_E := \begin{cases} v|_{\partial K_i \cap E} - |_{\partial K_j \cap E}, & \text{if } K_i \text{ and } K_j \text{ are neighbors along facet } E \text{ and} \\ & i < j, \\ v|_{\partial K \cap E}, & \text{if } E \in \partial\mathcal{E}_h \cap \mathcal{E}_h(K) \text{ for a } K \in \mathcal{T}_h, \end{cases}$$

and the average of a function on E by

$$\langle v \rangle_E := \begin{cases} \frac{1}{2}(v|_{\partial K_i \cap E} + v|_{\partial K_j \cap E}), & \text{if } K_i \text{ and } K_j \text{ are neighbors along facet } E \\ & \text{and } i < j, \\ v|_{\partial K \cap E}, & \text{if } E \in \partial\mathcal{E}_h \cap \mathcal{E}_h(K) \text{ for a } K \in \mathcal{T}_h. \end{cases}$$

The used DG discretization of (1) reads as follows: Find $u_h \in \mathcal{R}_{h,r}$ such that

$$a_{\mathrm{DG}}(u_h, v_h) = f_{\mathrm{DG}}(v_h) \quad \forall\, v_h \in \mathcal{R}_{h,r}, \tag{3}$$

where the bilinear form $a_{\mathrm{DG}} : H^1(\mathcal{T}_h) \times H^1(\mathcal{T}_h) \to \mathbb{R}$ is defined as $a_{\mathrm{DG}}(v,w) = a_\varepsilon(v,w) + a_{bc}(v,w)$ with

$$\begin{aligned} a_\varepsilon(v,w) = & \sum_{K \in \mathcal{T}_h} \int_K \varepsilon \nabla v \cdot \nabla w \, \mathrm{d}\boldsymbol{x} \\ & - \sum_{E \in \mathcal{E}_h^{\mathrm{ID}}} \varepsilon \int_E \Big(\langle \nabla v \cdot \boldsymbol{n}_E \rangle_E [\![w]\!]_E + \kappa \langle \nabla w \cdot \boldsymbol{n}_E \rangle_E [\![v]\!]_E \Big) \, \mathrm{d}s \\ & + \sum_{E \in \mathcal{E}_h^{\mathrm{I}}} \frac{\sigma}{h_E} \int_E [\![v]\!]_E [\![w]\!]_E \, \mathrm{d}s + \sum_{E \in \mathcal{E}_h^{\mathrm{D}}} \frac{2\sigma}{h_E} \int_E vw \, \mathrm{d}s \end{aligned} \tag{4}$$

and

$$\begin{aligned} a_{bc}(v,w) = & \sum_{K \in \mathcal{T}_h} \int_K \big(\boldsymbol{b} \cdot \nabla v w + cvw \big) \, \mathrm{d}\boldsymbol{x} - \sum_{E \in \mathcal{E}_h^{\mathrm{I}}} \int_E \boldsymbol{b} \cdot \boldsymbol{n}_E [\![v]\!]_E \langle w \rangle_E \, \mathrm{d}s \\ & + \sum_{E \in \mathcal{E}_h^{\mathrm{I}}} \int_E \frac{\eta}{2} |\boldsymbol{b} \cdot \boldsymbol{n}_E| [\![v]\!]_E [\![w]\!]_E \, \mathrm{d}s - \sum_{E \in \mathcal{E}_h^{-}} \int_E \boldsymbol{b} \cdot \boldsymbol{n}_E vw \, \mathrm{d}s. \end{aligned} \tag{5}$$

The right-hand side $f_{\mathrm{DG}} : H^1(\mathcal{T}_h) \to \mathbb{R}$ of (3) is given by

$$\begin{aligned} f_{\mathrm{DG}}(w) = &\sum_{K\in\mathcal{T}_h} \int_K fw \ \mathrm{d}\boldsymbol{x} - \sum_{E\in\mathcal{E}_h^-} \int_E \boldsymbol{b}\cdot\boldsymbol{n}_E g w \ \mathrm{d}s \\ &- \sum_{E\in\mathcal{E}_h^{\mathrm{D}}} \varepsilon\kappa \int_E \nabla w\cdot\boldsymbol{n}_E g \ \mathrm{d}s + \sum_{E\in\mathcal{E}_h^{\mathrm{D}}} \frac{2\sigma}{h_E}\int_E gw \ \mathrm{d}s. \end{aligned} \tag{6}$$

Method (3) contains three user-chosen parameters. The discretization (4) of the Laplacian is also called interior penalty (IP) method. The parameter κ in (4) determines the symmetry properties of the discretization of the Laplacian: $\kappa = 1$ gives the symmetric (SIP) method, $\kappa = 0$ the incomplete (IIP) method, and $\kappa = -1$ the non-symmetric (NIP) method. It is well known, that a_ε is coercive for NIP and any $\sigma > 0$ and for SIP and IIP if σ is sufficiently large, where the necessary magnitude of σ depends on ε. e.g., see [28, Chapter 2.7.1]. Note that κ appears also on the right-hand side (6). The parameter σ in (4) and (6) is a positive penalty parameter. The stabilization parameter σ is incorporated in the way as proposed in [25, Section 2.2], since the analysis in this work shows that it is more equilibrated in this form compared with other forms that can be found in the literature. Finally, the stabilization parameter $\eta \geq 0$ appears in (5), where $\eta = 0$ refers to a central flux and $\eta = 1$ refers to an upwind flux across the facet E. In our simulations, always $\eta = 1$ was utilized.

A convergence analysis for the DG method (3)–(6) for the case $\kappa = -1$ (NIP) was developed in [19]. In particular, a robust error estimate was derived, i.e., the constant of the error bound does not blow up as $\varepsilon \to 0$. As usual for convection-diffusion equations, the norm for which the error bound is proved contains contributions from the bilinear form. For functions that are piecewise sufficiently smooth, with respect to the underlying grid, it is $|||v|||_{\mathrm{DG}}^2 = a_{\mathrm{DG}}(v,v)$. An estimate of the form

$$|||u-u_h|||_{\mathrm{DG}} \leq Ch^r \left(\frac{\varepsilon}{r^{2r-1}} + \frac{h\|\boldsymbol{b}\|_{L^\infty}}{r^{2r+1}} + C_1(\nabla\cdot\boldsymbol{b}, c, \mu)\frac{h^2\|\mu\|_{L^\infty}^2}{r^{2r+2}} \right)^{1/2} \|u\|_{H^{r+1}},$$

was proved, provided that $\sigma = \varepsilon r^2$, where the constant C is independent of ε. Thus, in the convection-dominated regime, where $\varepsilon \leq h\|\boldsymbol{b}\|_{L^\infty}$, the order of error reduction is $r + 1/2$. All error bounds that will be mentioned below have the same order of convergence r and the same order of error reduction $r + 1/2$ if $\varepsilon \leq h\|\boldsymbol{b}\|_{L^\infty}$ as the bound from [19].

A convergence analysis for the SIP method of the diffusive term and for a different norm is presented in [25, Chapter 5.1], see also [17]. This norm contains explicitly a term with streamline derivative

$$\left(\sum_{K\in\mathcal{T}_h} h\|\boldsymbol{b}\cdot\nabla v\|_{L^2(K)}^2 \right)^{1/2}.$$

The derived error bound is robust, which shows that the DG method controls the streamline derivative even without the presence of a special stabilization term for this derivative. Another robust error analysis for the SIP method of the diffusive term, even for heterogeneous diffusion, can be found in [11, Chapter 4.6.3.2]. This analysis considers a slightly different norm than the analysis in [17, 25], where besides a term for the streamline derivative also terms appear that contain the normal component of the convection field on facets. In [6], the error analysis of a so-called multiscale DG method is presented. A class of DG methods, which is derived with the so-called weighted-residual approach, was analyzed in [2]. The methods from [19] and [6] belong to this class. A robust estimate for a norm containing the streamline derivative is proved in [2].

3 Approaches for Reducing Spurious Oscillations in Numerical Solutions of Convection-Diffusion-Reaction Equations

This paper considers post-processing techniques for reducing spurious oscillations. After having computed the discrete solution with the DG method (3)–(6), the idea consists in identifying those subregions where unphysical oscillations might occur and then to reduce or clip the degree of the polynomial approximation in these subregions, thereby utilizing slope limiters. Thus, the post-processing techniques consist of the following two steps:

1 Identify and mark cells where the numerical solution might possess spurious oscillations.

2 Approximate the solution on the marked cells by a polynomial of lower degree by utilizing a slope limiter.

Post-processing approaches of this kind, which are also included in the numerical studies, are proposed in [9, 13, 14]. In this section, these approaches are described and several generalizations and modifications will be proposed. The presentation will be restricted to the two-dimensional case. An extension to three dimensions is generally straightforward.

3.1 Post-Processing Based on Linear Reconstructions Across Faces of Mesh Cells

This post-processing approach was proposed in [9], see also [28, Chapter 4.3.1] for a presentation. It is formulated for triangles, it is derived from the assumption that spurious oscillations in the discrete solution only arise if they occur in its linear part, which is its $L^2(\Omega)$ projection into the space of piecewise linear functions, and it uses a (at most) linear approximation in a vicinity of layers.

Let K be an interior triangle, thus possessing three neighbors $\widehat{K}_{K,0}$, $\widehat{K}_{K_1}$, $\widehat{K}_{K,2}$, where the numbering is in accordance to the local edge numbering of K. The barycenters of the triangles are denoted by b_K and $b_{\widehat{K}_{K,i}}$, $i = 0, 1, 2$, respectively, and the midpoints of the edges are labeled with $m_{K,i}$, $i = 0, 1, 2$, see Figure 1. Using the notation $\overline{u}_{h,K} := \int_K u_h \, \mathrm{d}\mathbf{x}/|K|$, it is checked whether $u_h|_K(m_{K,i})$ is between $\overline{u}_{h,K}$ and $\overline{u}_{h,\widehat{K}_{K,i}}$ for $i = 0, 1, 2$. If for at least one i, the value at the edge midpoint is not between the cell averages of the adjacent cells, the cell K is marked.

For all marked cells three affine functions $L_j(x, y) := a^j_{0,K} + a^j_{1,K}x + a^j_{2,K}y$, $j = 0, 1, 2$, are constructed. They are defined by

$$L_j(b_K) = \overline{u}_{h,K}, \quad L_j(b_{K,j+1}) = \overline{u}_{h,\widehat{K}_{K,j+1}} \quad L_j(b_{K,j+2}) = \overline{u}_{h,\widehat{K}_{K,j+2}},$$

where $b_{K,3} := b_{K,0}$, $b_{K,4} := b_{K,1}$, and $\widehat{K}_{K,3} := \widehat{K}_{K,0}$, $\widehat{K}_{K,4} := \widehat{K}_{K,1}$. Afterwards the three affine functions are ordered by decreasing values $\sqrt{(a^j_{1,K})^2 + (a^j_{2,K})^2}$.

Starting with the largest of these values, the affine functions are now tested whether $L_j(m_i)$ lies between $\overline{u}_{h,K}$ and $\overline{u}_{h,\widehat{K}_{K,i}}$ for $i = 0, 1, 2$. If for a j the affine function satisfies this condition, then u_h is locally replaced by this function and the remaining limiters are discarded. Otherwise, i.e., if none of the three affine function fulfills the condition, u_h gets replaced by $\overline{u}_{h,K}$. An advantage of this approach is that it does not contain user-chosen parameters. Below, it will be called *LinTriaReco* (linear approximation on triangles based on a reconstruction).

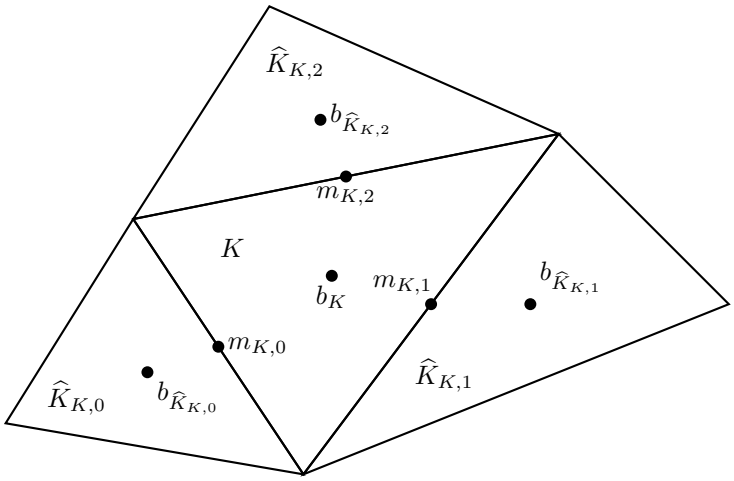

Figure 1: Notations for an interior triangle K having three neighbors $\widehat{K}_{K,i}$ with barycenters $b_{\widehat{K}_{K,i}}$ and edge midpoints $m_{K,i}$, $i = 0, 1, 2$.

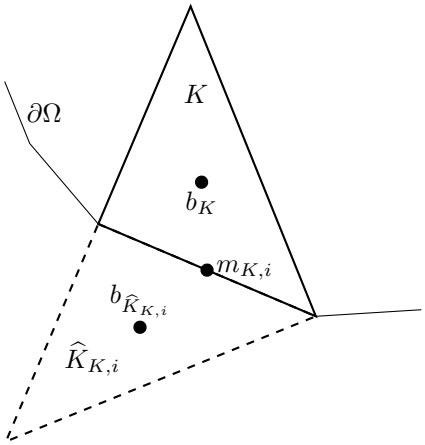

Figure 2: Construction of a 'virtual' triangle $\widehat{K}_{K,i}$ by reflecting the original triangle K across the boundary edge E_i.

Algorithm *LinTriaReco* is only defined if the triangles possess three neighbors. Since layers appear also on boundary triangles, it is desirable to extend the algorithm to those cells. A possible approach for boundary triangles is to construct 'virtual' neighbors, define a solution on them and then apply the algorithm as before. This is done in the following way. For a boundary edge $E_i \in \mathcal{E}_h(K) \cap \partial\mathcal{E}_h$ of K, a 'virtual' triangle $\widehat{K}_{K,i}$ is constructed by reflecting the original triangle across E_i, see Figure 2. In addition to that, the algorithm requires a mean value of some function u_h on the 'virtual' triangles. For such a triangle $\widehat{K}_{K,i}$, the 'virtual' solution $u_h|_{\widehat{K}_{K,i}}$ is defined to be the continuation of u_h from K to $\widehat{K}_{K,i}$, which is well defined, since u_h is a polynomial on K.

Since undershoots and overshoots can occur not only at the edge midpoint but also at a vertex of a cell, it might be a good idea to include those points to the decision whether the finite element function on a cell shall be limited or not. A possible generalization of the original indicator is, instead of considering the value of u_h at the edge midpoint, it is examined whether the integral mean along the edge lies between the cell averages of the two adjacent cells.

Another crucial point of the original algorithm is the fact that it may decrease locally the maximum in a cell but at the same time decreases also the minimum, or vice versa. In the worst case, it can happen

that the total amount of overshoot is decreased but at the price that the total amount of undershoot is increased. Altogether, it might be a safer choice, with respect to the size of the spurious oscillations, to always replace the solution on marked cells by a constant, namely by its local integral mean.

The modified post-processing algorithm that uses the approach for boundary cells described above, the integral mean values along the edges, and a locally constant approximation of marked cells is denoted by *ConstTriaReco*.

Next, an extension of the post-processing approaches from triangles to quadrilaterals will be proposed. Taking three out of the four edges, one can perform the same methods as for triangles. There are four possibilities for choosing three edges and for the method *LinQuadReco* the post-processing is computed for all of them. This gives four admissible affine functions, where admissible has to be understood in the sense as explained for triangles. From these functions, this is chosen as local approximation with the smallest Euclidean norm $\sqrt{(a_{1,K})^2 + (a_{2,K})^2}$. The method *ConstQuadReco* applies the ideas of *ConstTriaReco* to the quadrilateral. If in this method the cell is marked by one combination of edges, it is not longer necessary to consider the other combinations since the locally constant approximation does not depend on the neighbor cells.

3.2 Post-Processing Based on Weighted Mean Derivatives

In [9], this approach is proposed only for axis-parallel rectangular grids, see [28, Chapter 4.3.1] for another presentation. Here, a generalization to quadrilaterals being the image of a reference cell under an affine transform is presented.

In the code used in the numerical simulations presented in Section 4, so-called mapped finite elements are implemented, i.e., the basis functions and nodal functionals of a finite element are defined on a reference cell $\hat{K}$ and the functions and functionals on the physical cell K are given by the reference transform $F_K : \hat{K} \to K$, where $\hat{K} := [-1, 1]^2$.

Since mapped finite elements are used, it is near at hand to base the definition of the post-processing technique on the reference cell and the reference transform. Consider from now on the case that the transform is affine. Then, on K, the functions

$$\psi(x, y) = \frac{F_{K,1}^{-1}(x, y)}{2}, \qquad \xi(x, y) = \frac{F_{K,2}^{-1}(x, y)}{2}$$

are defined, where $F_{K,1}^{-1}$ and $F_{K,2}^{-1}$ are the first and second component of the inverse of the transform F_K, respectively. Note that for axis-parallel rectangular cells both functions coincide with their respective definition given in [28, Chapter 4.3.1] and differ from [9] by a factor of one half, if the reference transform does not rotate or reflect the vertices of the reference cell.

Locally on K, the discrete function u_h can be expanded by

$$u_h|_K(x, y) = a_{0,K} + a_{1,K}\psi(x, y) + a_{2,K}\xi(x, y) + \text{ higher order terms}, \tag{7}$$

where $a_{0,K} := \overline{u}_{h,K}$ is defined to be the integral mean of u_h over K. Defining the following nodal functionals

$$N_0(v_h) := \frac{1}{|K|}\int_K v_h \, \mathrm{d}\mathbf{x}, \quad N_1(v_h) := C_1 \int_{\widehat{K}} v_h(F_K(\widehat{x}, \widehat{y}))\, \widehat{x} \, \mathrm{d}\widehat{\mathbf{x}},$$
$$N_2(v_h) := C_2 \int_{\widehat{K}} v_h(F_K(\widehat{x}, \widehat{y}))\, \widehat{y} \, \mathrm{d}\widehat{\mathbf{x}},$$

then the N_K-dimensional finite element space on K is equipped with a so-called local basis $\{\varphi_i\}_{i=0}^{N_K-1}$, i.e., it holds $N_i(\varphi_j) = \delta_{ij}$, $i = 0, \ldots, N_K - 1$, where δ_{ij} is the Kronecker symbol, and $\varphi_0 = 1$, $\varphi_1 = \psi$, $\varphi_2 = \xi$. Note that the nodal functionals and the local basis functions for $i \geq 3$ are not needed for computing the coefficients in (7).

Altogether, the terms with $a_{1,K}$ and $a_{2,K}$ in (7) can be interpreted as the collection of all the linear parts of u_h. Neglecting the higher order terms in (7), denoting the resulting affine function with $\tilde{u}$, and denoting by B_K the matrix of the affine transform, one finds that

$$\nabla \tilde{u}_h = \frac{B_K^{-1}}{2} \begin{pmatrix} a_{1,K} \\ a_{2,K} \end{pmatrix}.$$

The first factor on the right-hand side can be considered to be a weight. Since $\tilde{u}_h$ comprises only the linear part of u_h, $a_{1,K}$ and $a_{2,K}$ can be thought of providing information on a weighted mean derivative of u_h in K.

If $|a_{1,K}|$ or $|a_{2,K}|$ are too large, then slope limiting is applied by considering the jumps of the mean values across the edges in the respective directions. Let K_l be the neighbor of K across the image of $\hat{E}_l$, and use analogously the notations K_r, K_b, and K_t. It is assumed that $\hat{E}_l := \overline{(-1,1)(-1,-1)}$, $\hat{E}_r := \overline{(1,-1)(1,1)}$, $\hat{E}_t := \overline{(1,1)(-1,1)}$ and $\hat{E}_b = \overline{(-1,-1)(1,-1)}$. Now, for two user-chosen constants $M_{\text{lim}} \geq 0$ and $\gamma \geq 0$, modified coefficients

$$\overline{a}_{1,K} := \begin{cases} a_{1,K}, & \text{if } |a_{1,K}| \leq M_{\text{lim}}, \\ \operatorname{minmod}\left(a_{1,K}, \gamma(a_{0,K_r} - a_{0,K}), \gamma(a_{0,K} - a_{0,K_l})\right), & \text{else,} \end{cases}$$

$$\overline{a}_{2,K} := \begin{cases} a_{2,K}, & \text{if } |a_{2,K}| \leq M_{\text{lim}}, \\ \operatorname{minmod}\left(a_{2,K}, \gamma(a_{0,K_t} - a_{0,K}), \gamma(a_{0,K} - a_{0,K_b})\right), & \text{else} \end{cases}$$

are computed. The minmod function is defined by

$$\operatorname{minmod}(a_0, a_1, a_2) := \begin{cases} s \, \min_{i=0,1,2} |a_i|, & \text{if } s := \operatorname{sign}(a_0) = \operatorname{sign}(a_1) = \operatorname{sign}(a_2), \\ 0, & \text{else.} \end{cases}$$

For K owing boundary edges and therefore having less than four neighbors, the minmod function is called without the entry that corresponds to the non-existing neighbor(s).

The indicator tests whether $\overline{a}_{1,K} \neq a_{1,K}$ or $\overline{a}_{2,K} \neq a_{2,K}$. If this is the case, then the cell is marked and the solution is replaced by the (at most) affine function

$$a_{0,K} + \overline{a}_{1,K}\psi(x,y) + \overline{a}_{2,K}\xi(x,y). \tag{8}$$

In the numerical simulations, we used the same parameters as proposed in [28, p. 104]: $M_{\text{lim}} = 0$ and $\gamma = 1$. This method will be called *LinQuadDeriv*, because of the connection to the weighted mean derivative explained above.

With the same arguments as for triangles, the linear approximation (8) might not sufficiently reduce the spurious oscillations. For this reason, we studied also a slight variation of the method that uses locally the constant approximation $a_{0,K}$ instead of (8), which is called *ConstQuadDeriv*.

A generalization of this approach to d-linear reference transforms will be subject to future research.

3.3 Post-Processing Based on Evaluating Jumps Across Facets

This section deals with a post-processing technique that reconstructs the solution on all marked cells by a piecewise constant function. It was first introduced in [13] and further analyzed in [14].

The indicator is based on the observation from numerical studies with DG methods with $r = 1$ that on mesh cells K where the numerical solution is smooth, it holds

$$\sum_{E \in \mathcal{E}_h(K) \cap \mathcal{E}_h^{\mathrm{I}}} \int_E \frac{[\![u_h]\!]_E^2}{h_K^5} \, \mathrm{d}s \approx \sum_{E \in \mathcal{E}_h(K) \cap \mathcal{E}_h^{\mathrm{I}}} \int_E \frac{(\mathcal{O}(h_K^2))^2}{h_K^5} \, \mathrm{d}s \approx \mathcal{O}(1), \tag{9}$$

whereas in the vicinity of layers, it is

$$\sum_{E \in \mathcal{E}_h(K) \cap \mathcal{E}_h^{\mathrm{I}}} \int_E \frac{[\![u_h]\!]_E^2}{h_K} \, \mathrm{d}s \approx \sum_{E \in \mathcal{E}_h(K) \cap \mathcal{E}_h^{\mathrm{I}}} \int_E \frac{(\mathcal{O}(1))^2)^2}{h_K} \, \mathrm{d}s \approx \mathcal{O}(1). \tag{10}$$

Hence, for $\alpha \in (1, 5)$, the quantity

$$\sum_{E \in \mathcal{E}_h(K) \cap \mathcal{E}_h^{\mathrm{I}}} \int_E \frac{[\![u_h]\!]_E^2}{h_K^\alpha} \, \mathrm{d}s \tag{11}$$

can serve as an indicator [14]. Also note that this indicator works both on triangles and quadrilaterals. In [13, 14], it is proposed to use basically $\alpha = 5/2$. To be precise, all cells are marked for which

$$\sum_{E \in \mathcal{E}_h(K) \cap \mathcal{E}_h^{\mathrm{I}}} \int_E \frac{[\![u_h]\!]_E^2}{h_K |K|^{3/4}} \, \mathrm{d}s \geq 1. \tag{12}$$

On the marked cells, the discrete solution is replaced by the integral mean value $\overline{u}_{h,K}$, i.e., always a constant approximation is applied. Besides α, a second user-chosen constant is the 1 on the right-hand side of (12).

The original method from [13, 14] is included in the numerical studies, where it will be denoted by *ConstJump*. Note that the asymptotic behavior (9) for smooth solutions can be deduced also from error bounds, since a sum of jumps across facets is on the left-hand side of these bounds. In addition, the power in the denominator increases with increasing polynomial degree of the finite element function. However, for the post-processing approach, (9) is not really of interest. because one likes to detect the non-smooth subregions. These are indicated by (10), which holds independently of the polynomial degree. Consequently, the approach from [13, 14] can be applied also for DG methods with $r > 1$.

Note that the method *ConstJump* should be utilized only for small mesh cells, for which a discussion about an asymptotic behavior is meaningful. If $h_K |K|^{3/4} > 1$, or even $h_K |K|^{3/4} \gg 1$, then the denominator in (12) is large. In this case, (12) might not be satisfied even for large jumps and the cell is not marked.

The choice $\alpha = 2.5$ seems to be based on the experience of the authors of [13, 14]. Other choices are also possible, e.g., increasing α would increase the number of mesh cells to be marked. We did not perform numerical studies with respect to choosing α, but investigated a different modification of this method, which is inspired from (11) and will be called *ConstJumpMod*. First, instead of using the sum over all facets of K, each facet is considered individually. In this way, it does not play any role if the facets are of much different size. In addition, a non-smooth behavior of the discrete solution across

just one facet can be detected better, which might occur if this facet is aligned with a layer. Then, the ansatz for the smoothness indicator is

$$\int_E [\![u_h]\!]_E^2 \ \mathrm{d}s = C_0 h_E^{\alpha_E} \quad \Longrightarrow \quad \alpha_E = \frac{\ln\left(\frac{1}{C_0}\int_E [\![u_h]\!]_E^2 \ \mathrm{d}s\right)}{\ln(h_E)}, \quad \text{for } h_E < 1, \tag{13}$$

such that α_E can be computed for each facet. Next, α_K is set to be the smallest value of α_E for the facets of K. Finally, K is marked for a constant approximation if $\alpha_K \leq \alpha_{\mathrm{ref}}$ for some user-chosen constant α_{ref}. The second user-chosen constant in this approach was set to be $C_0 = 1$ for all numerical simulations.

Similarly to the original method *ConstJump*, there is an issue if $h_E \geq 1$. We think that also in this situation, the jumps of the numerical solution across facets provide information whether or not the facet is in a vicinity of layers. Because, on the one hand, $h_E \geq 1$ did not occur in our numerical studies and, on the other hand, we think that the ideas we have so far for a scaling invariant modification of this approach need still to be improved, we like to postpone this issue to future research.

4 Numerical Studies

The goal of the numerical studies consists in investigating to which extent the methods presented in Section 3 reduce spurious oscillations which are introduced by the DG method (3) – (6). To this end, two standard benchmark problems for convection-diffusion equations in two dimensions are considered.

All simulations were performed with the code PARMOON, cf. [16, 31]. The implementation of the DG method was validated by first considering a smooth solution and comparing the orders of convergence for the pure diffusion problem with the orders proposed by numerical analysis, e.g., see [12, Chapter 2.7]. Then, the same approach was performed for convection-diffusion-reaction equations with respect to the analytic convergence results from [25, 11]. For the sake of brevity, we will present below only results for the SIP discretization diffusive term, i.e., $\kappa = 1$ in (4) and (6). As already mentioned, we used always the upwind scheme, i.e., $\eta = 1$ in (5). The choice of the last parameter, $\sigma = 5r^2\varepsilon$, was guided by [28, Chapter 2.7.1] and we did not encounter any instabilities in the numerical simulations with this selection.

All linear systems of equations were solved with the sparse direct solver UMFPACK [10].

A first measure for the size of the spurious oscillations is just to take the smallest and largest value of the discrete solution u_h into account and compare them with the minimal value $u_{\min}$ and the maximal value $u_{\max}$ of the solution of the continuous problem. This approach gives the measure

$$\mathrm{osc}_{\max}(u_h) = \max_{(x,y)\in\Omega} u_h(x,y) - u_{\max} + u_{\min} - \min_{(x,y)\in\Omega} u_h(x,y). \tag{14}$$

In this measure, just two values of u_h determine the quality of the numerical solution. It will not be distinguished between numerical solutions with many large spurious oscillations, close to the maximal ones, and solutions with few or only one large spurious oscillations. For this reason, we decided to use

also a measure that takes the size of all spurious oscillations into account. This measure is

$$\begin{aligned}\mathrm{osc}_{\mathrm{mean}}(u_h) =& \frac{1}{|\mathcal{T}_h|}\Bigg[\sum_{K\in\mathcal{T}_h} \max\{0, \max_{(x,y)\in K} u_h(x,y) - u_{\max}\} \\ &+ \max\{0, u_{\min} - \min_{(x,y)\in K} u_h(x,y)\}\Bigg],\end{aligned} \tag{15}$$

where $|\mathcal{T}_h|$ is the number of mesh cells of $\mathcal{T}_h$.

The following approaches were studied in the numerical simulations:

- *Galerkin*. DG method (3)–(6) without post-processing for reducing spurious oscillations,
- *LinTriaReco*. triangular grids, post-processing with locally linear approximation, based on a reconstruction across facets, see Section 3.1,
- *ConstTriaReco*. modification of *LinTriaReco* as described in Section 3.1,
- *LinQuadReco*. extension of *LinTriaReco* to quadrilaterals, see Section 3.1,
- *ConstQuadReco*. extension of *ConstTriaReco* to quadrilaterals, see Section 3.1,
- *LinQuadDeriv*. quadrilateral grids, post-processing with locally linear functions, post-processing based on a mean derivative, $M_{\mathrm{lim}} = 0$, $\gamma = 1$, see Section 3.2,
- *ConstQuadDeriv*. like *LinQuadDeriv*, but with locally constant approximation, $M_{\mathrm{lim}} = 0$, $\gamma = 1$, see Section 3.2,
- *ConstJump*. all types of grids, locally constant approximation, post-processing based on evaluating jumps across facets, $\alpha = 2.5$, see Section 3.3,
- *ConstJumpMod*. modification of *ConstJump*, $\alpha_{\mathrm{ref}} = 4$, $C_0 = 1$, see Section 3.3.

For the sake of brevity, we do not report results of parameter studies, but show only results for the parameters given above. For *LinQuadDeriv* and *ConstJump*, these parameters are proposed in the literature.

Simulations were performed with polynomials of degree $r \in \{1, 2, 3, 4\}$.

Example 1 (*Convection skew to the mesh)*. This example is a slight variation of a classical benchmark problem proposed in [21]. It is given in $\Omega = (0,1)^2$ with $\boldsymbol{b} = (\cos(-\pi/3), \sin(-\pi/3))^T$, $c = f = 0$ and the following Dirichlet boundary condition

$$u = \begin{cases} 1 & (y = 1 \wedge x > 0) \text{ or } (x = 0 \wedge y > 0.75), \\ 0 & \text{else}, \end{cases}$$

see Figure 3 for a sketch of the solution. The solution has an interior layer in the direction of the convection that starts at the jump of the boundary condition and two boundary layers at the outflow boundary. It takes values in $[0,1] = [u_{\min}, u_{\max}]$.

The modification of the present configuration compared with [21] consists in placing the jump of the boundary condition at $(0, 0.75)$ instead of $(0, 0.7)$. With this modified position, the jump of the

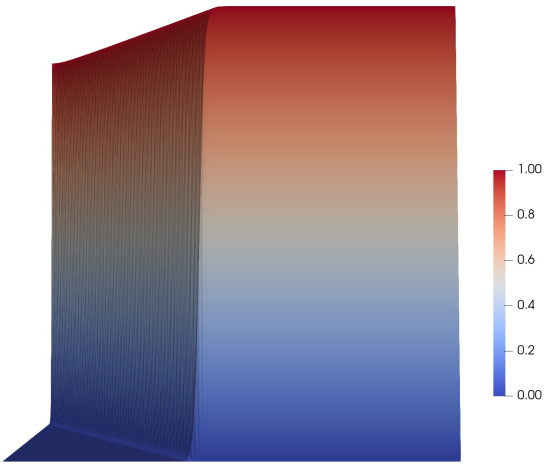

Figure 3: Example 1, sketch of the solution for $\varepsilon = 10^{-8}$, computed with a nonlinear algebraic flux-corrected (AFC) finite element method with Kuzmin limiter, see [3].

boundary condition is situated at a vertex of the triangulation after few refinement steps starting with a standard coarse grid. First, it is usually advisable to adjust the grid to known singularities of the solution. And second, particularly for DG methods, the piecewise smoothness of the solution, with respect to the given triangulation, plays a role. If the singularity is not located at a vertex, then this smoothness is very low in the affected mesh cell.

The initial quadrilateral grid consists of just one mesh cell and the initial triangular grid of two cells, which are obtained by dividing Ω with the diagonal from $(0, 1)$ to $(1, 0)$. Results will be shown starting from appropriate refinements of these grids.

Results for two different parameters are presented: $\varepsilon = 10^{-4}$ for a moderately convection-dominated problem and $\varepsilon = 10^{-8}$ for a strongly convection-dominated problem.

The results for $\varepsilon = 10^{-4}$ can be found in Figures 4–7. Considering first triangular grids, a considerable reduction of $\mathrm{osc}_{\max}(u_h)$ can be observed for *ConstJump* and *ConstJumpMod* on many grids, where *ConstJumpMod* is often a little bit better. For *ConstTriaReco*, there is usually a notable reduction achieved whereas the maximal size of spurious oscillations for *LinTriaReco* is often the same as for *Galerkin*. But there is no method that removes the spurious oscillations completely. Concerning $\mathrm{osc}_{\mathrm{mean}}(u_h)$, Figure 5, one can see that all approaches reduce the spurious oscillations compared with *Galerkin*. Again, *ConstJumpMod* is usually among the best methods, followed by *ConstJump* and *ConstTriaReco*. The values of $\mathrm{osc}_{\mathrm{mean}}(u_h)$ for the last method increase sometimes on finer meshes. The decrease of $\mathrm{osc}_{\mathrm{mean}}(u_h)$ for successive mesh refinement is expected, since the total number of mesh cells scales quadratically and the number of cells in a vicinity of layers, which are anticipated to be marked, scales linearly.

On quadrilateral grids, Figures 6 and 7, good reductions of $\mathrm{osc}_{\max}(u_h)$ are usually obtained with *ConstQuadReco*, *ConstJumpMod* and, apart of Q_3 with *LinQuadDeriv* and *ConstQuadDeriv*. These methods show also the best results with respect to $\mathrm{osc}_{\mathrm{mean}}(u_h)$.

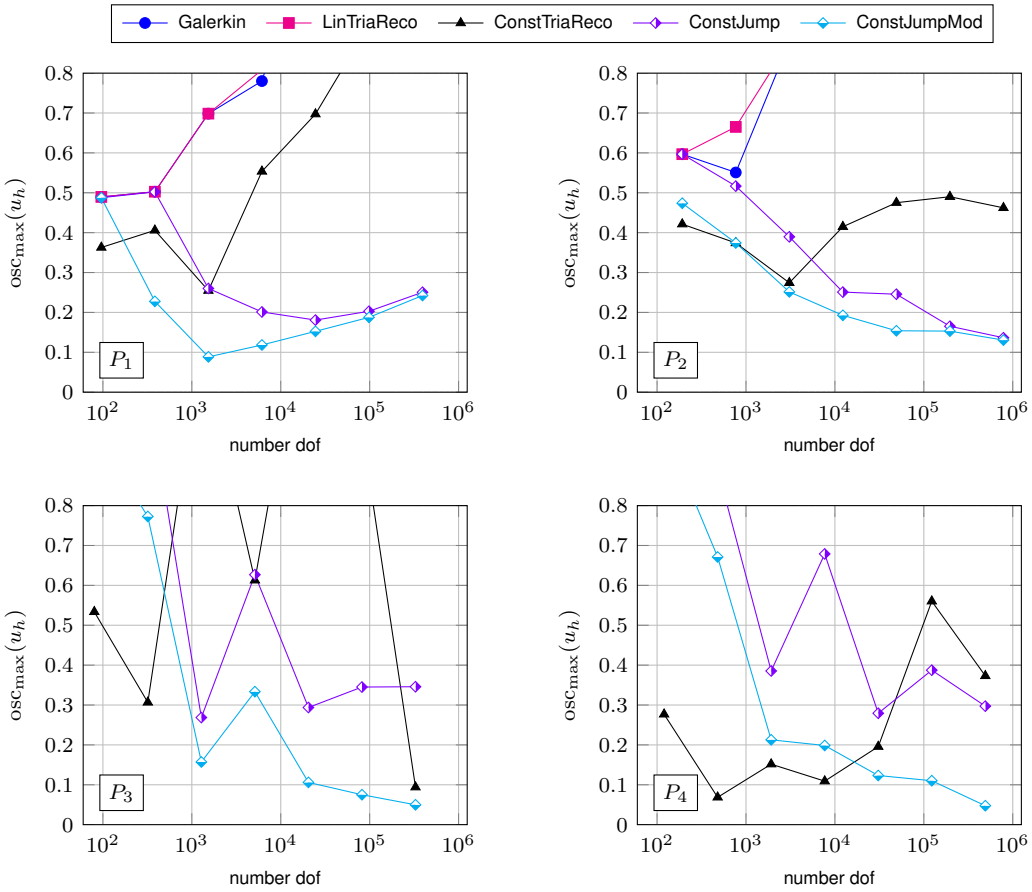

Figure 4: Example 1, triangular grids, $\varepsilon = 10^{-4}$, maximal value of oscillations defined in (14). The results of *LinTriaReco* lie often above the ones of *Galerkin*. Results not shown lie out of range of the plot.

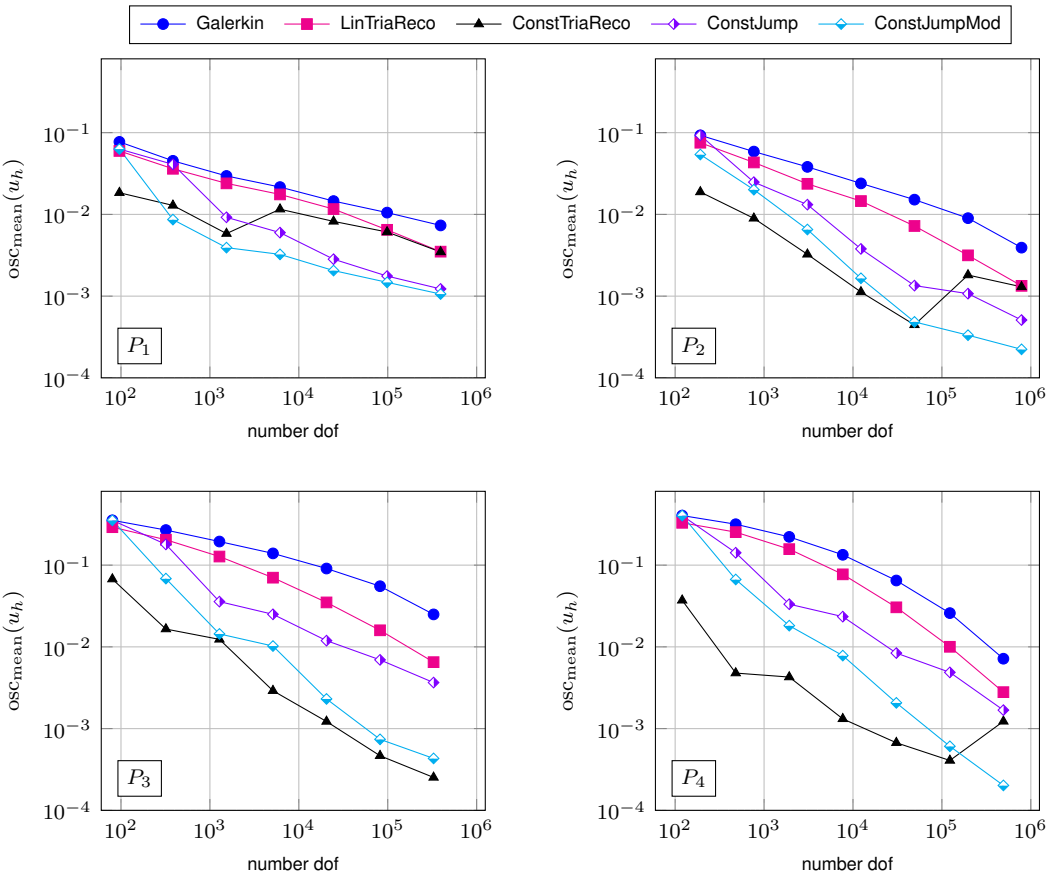

Figure 5: Example 1, triangular grids, $\varepsilon = 10^{-4}$, mean value of oscillations defined in (15).

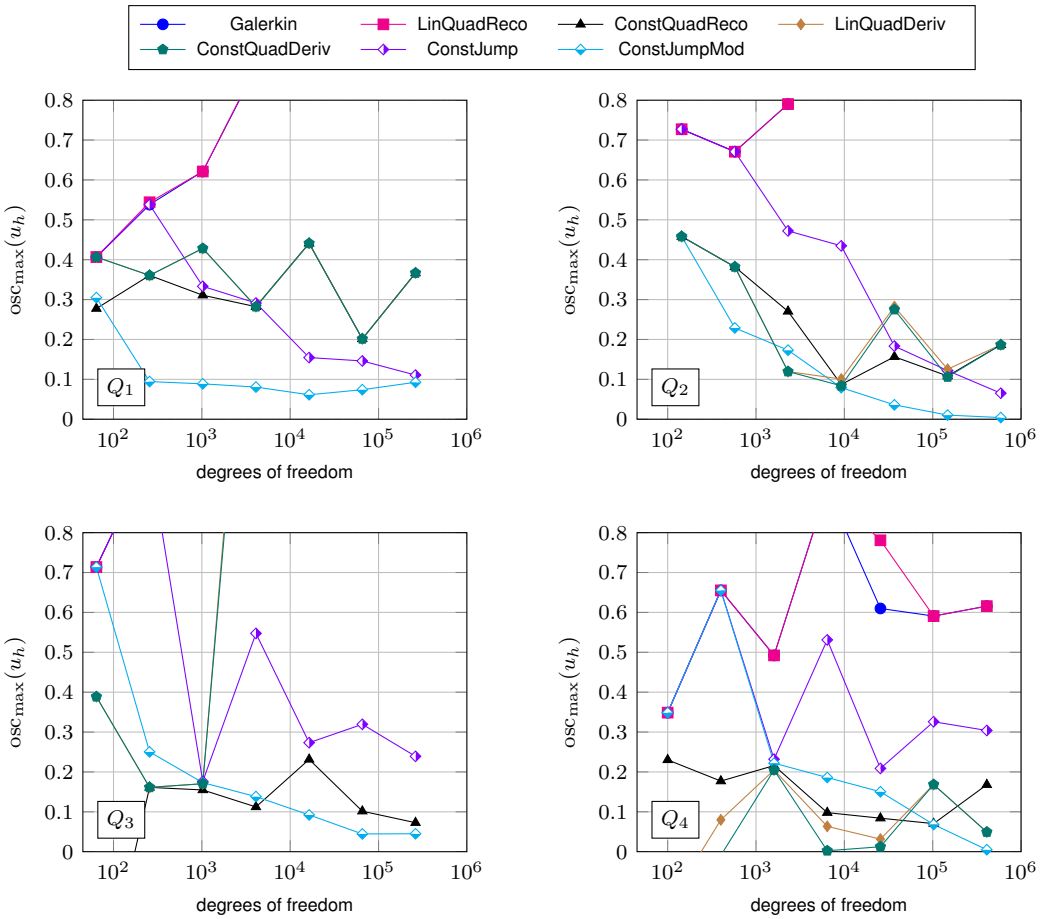

Figure 6: Example 1, quadrilateral grids, $\varepsilon = 10^{-4}$, maximal value of oscillations defined in (14). The results of *LinQuadReco* lie often above the ones of *Galerkin* as well as the values for $\mathrm{osc}_{\max}(u_h)$ of *ConstQuadDeriv* hide the results of *LinQuadDeriv*. Results not shown lie out of range of the plot.

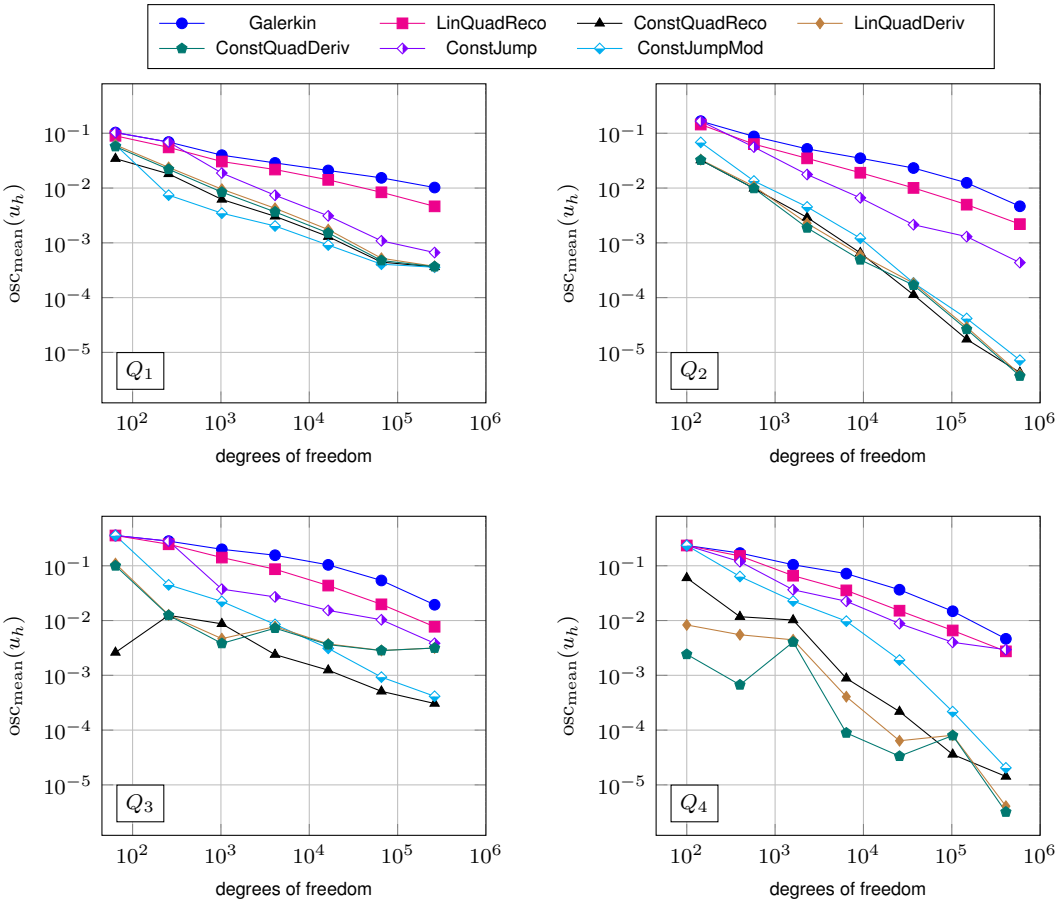

Figure 7: Example 1, quadrilateral grids, $\varepsilon = 10^{-4}$, mean value of oscillations defined in (15). The values for $\mathrm{osc}_{\mathrm{mean}}(u_h)$ of *ConstQuadDeriv* may hide the results of *LinQuadDeriv*.

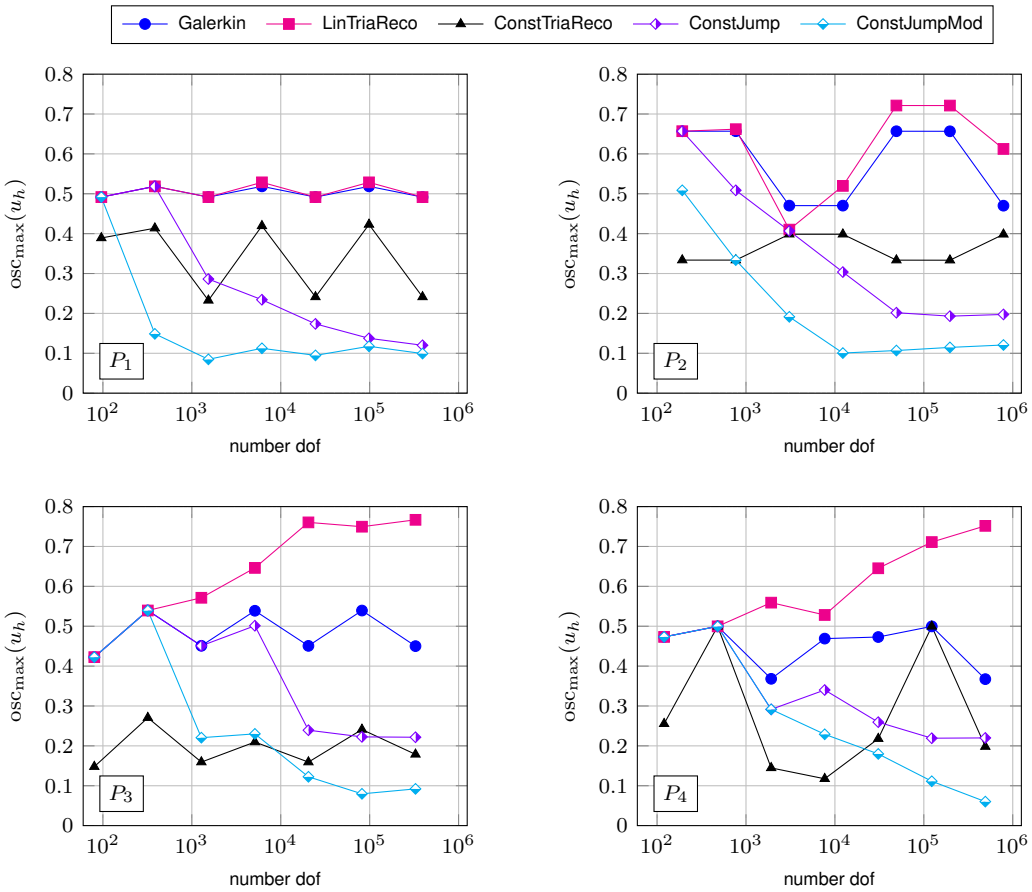

Figure 8: Example 1, triangular grids, $\varepsilon = 10^{-8}$, maximal value of oscillations defined in (14). The values for $\mathrm{osc}_{\max}(u_h)$ of *LinTriaReco* may hide the results of *Galerkin*.

Figures 8–11 depict the results for the strongly convection-dominated regime $\varepsilon = 10^{-8}$.

On triangular grids, good results concerning $\mathrm{osc}_{\max}(u_h)$ are usually obtained with *ConstJumpMod*. Often, also *ConstJump* reduces the maximal oscillations quite well, sometimes *ConstTriaReco*. With *LinTriaReco*, one can see in some cases even larger maximal oscillations than with *Galerkin*. However, Figure 9 demonstrates that all methods reduce the mean oscillations. In this respect, *ConstJumpMod* is again often the best approach. For *LinTriaReco*, one can conclude from the obtained results that there are much less oscillations than for *Galerkin*, but there are still very few large ones among them.

Evaluating the results on quadrilateral grids, one finds that *ConstJumpMod* belongs also in this case to the best method concerning $\mathrm{osc}_{\max}(u_h)$. *ConstJump* for Q_1 and *LinQuadDeriv* and *ConstQuadDeriv* for Q_4 show also good results. With respect to $\mathrm{osc}_{\mathrm{mean}}(u_h)$, *ConstJumpMod* was always a good approach, for Q_1 and Q_2 usually the best one. For higher order elements, the methods *ConstQuadReco*, *LinQuadDeriv*, and *ConstQuadDeriv* lead usually to similar or even better results than *ConstJumpMod*.

Two numerical solutions for approaches that lead to good solutions of both $\mathrm{osc}_{\max}(u_h)$ and $\mathrm{osc}_{\mathrm{mean}}(u_h)$ are presented in Figure 12. The exponential layers at the outflow boundaries are not present in these solutions due to the weak imposition of the homogeneous Dirichlet boundary condition. In both numerical solutions, one can observe that there are spurious oscillations caused by mesh cells on which the numerical solution is not constant, i.e., these mesh cells were not marked by the respective meth-

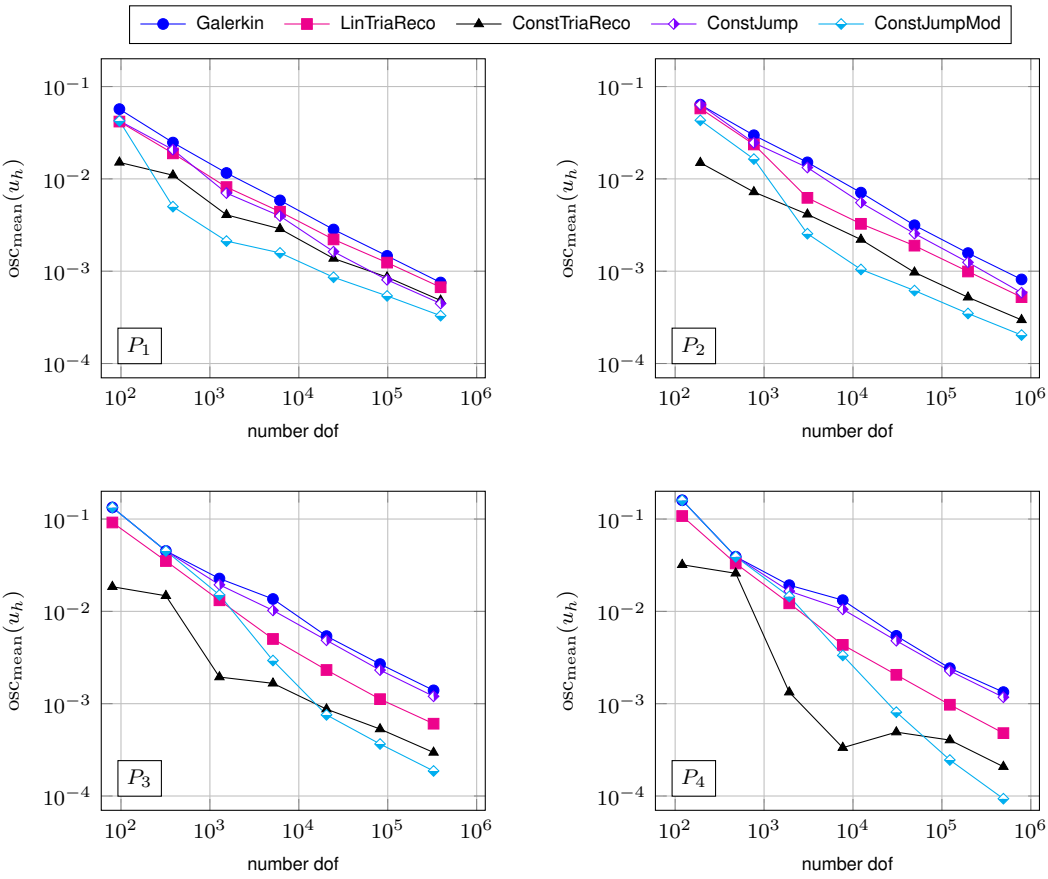

Figure 9: Example 1, triangular grids, $\varepsilon = 10^{-8}$, mean value of oscillations defined in (15).

ods.

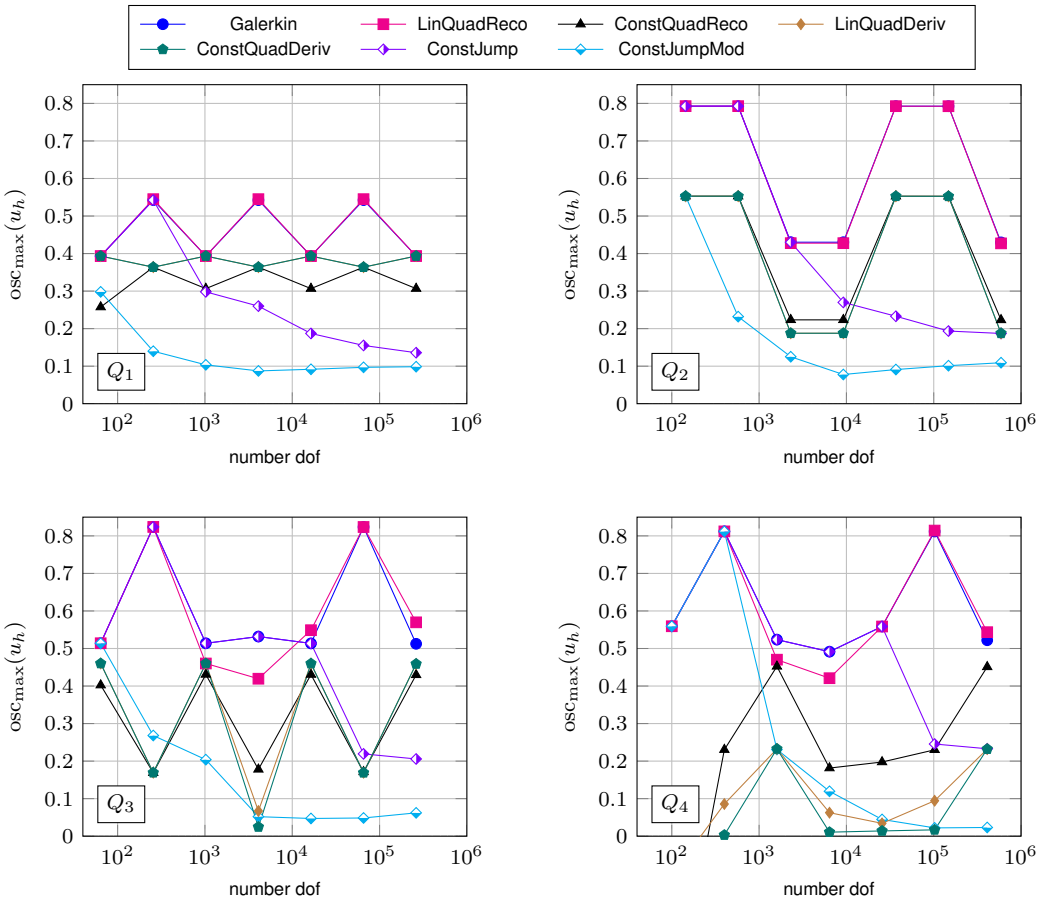

Figure 10: Example 1, quadrilateral grids, $\varepsilon = 10^{-8}$, maximal value of oscillations defined in (14). The results of *LinQuadReco* lie often above the ones of *Galerkin* as well as the values for $\mathrm{osc}_{\max}(u_h)$ of *ConstQuadDeriv* hide the results of *LinQuadDeriv*.

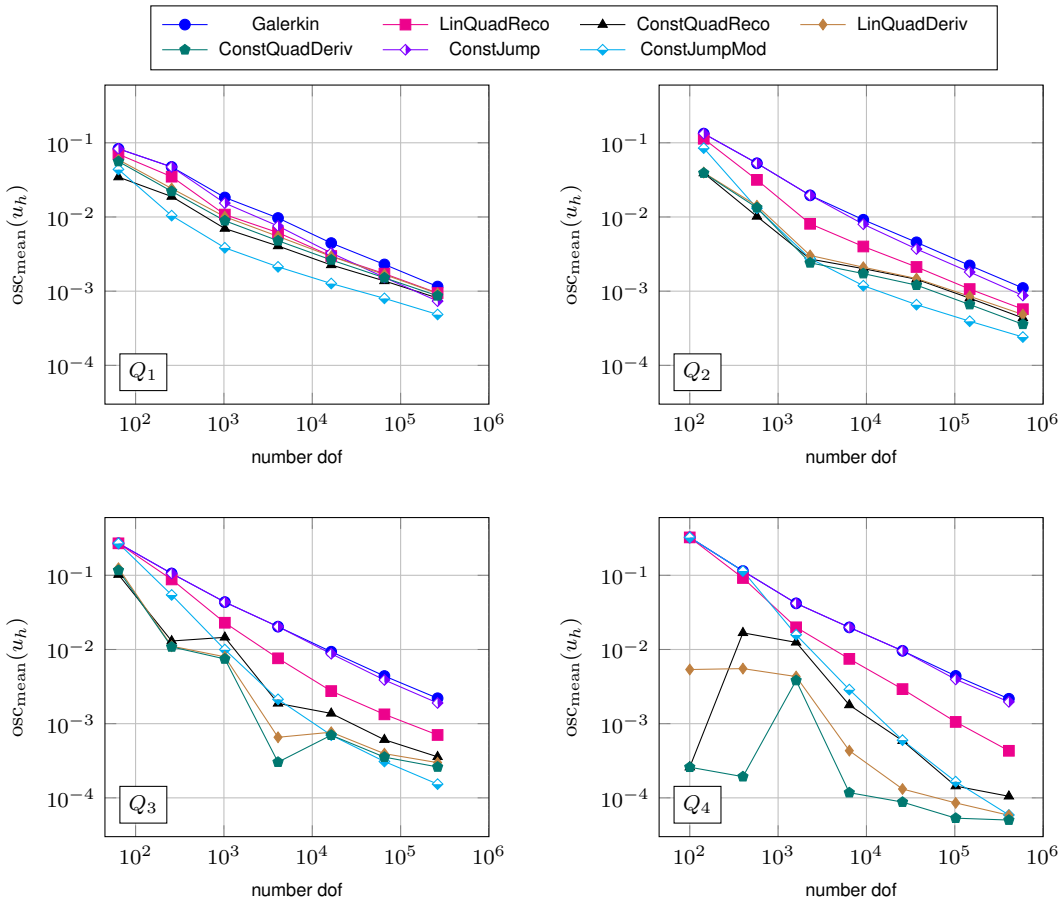

Figure 11: Example 1, quadrilateral grids, $\varepsilon = 10^{-8}$, mean value of oscillations defined in (15).

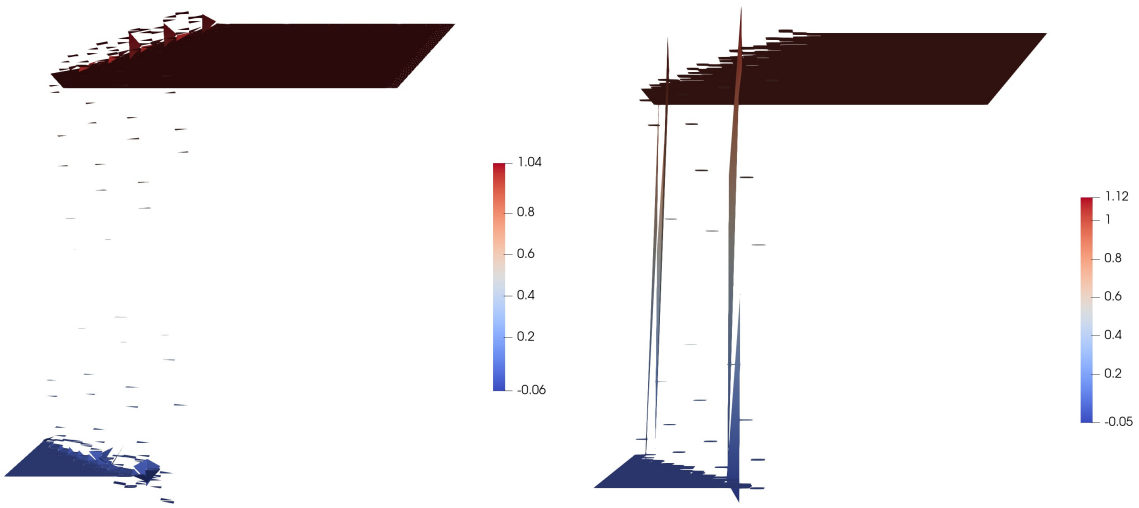

Figure 12: Example 1. Left: solution with P_2 and *ConstJumpMod* for $\varepsilon = 10^{-8}$. Right: solution with Q_4 and *ConstQuadReco* for $\varepsilon = 10^{-8}$. The solutions are projected to piecewise linear or bilinear functions by the visualization software.

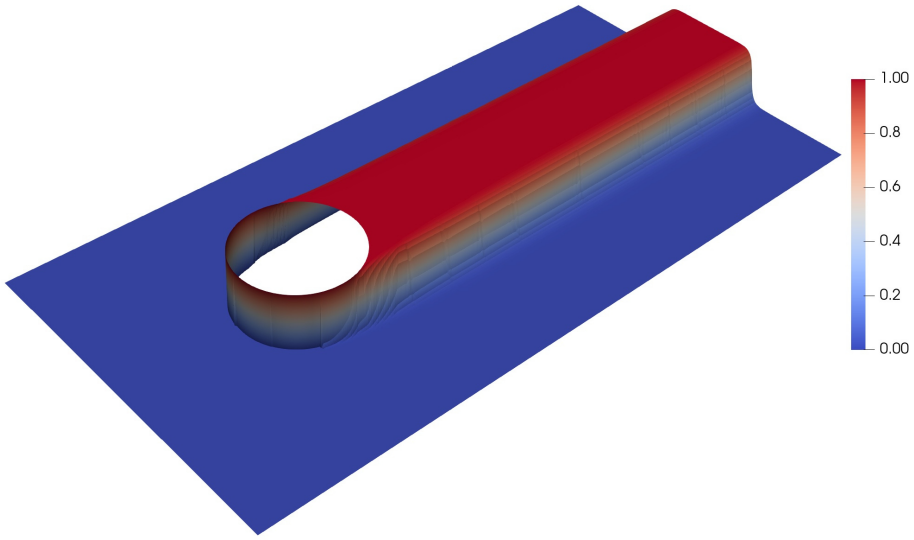

Figure 13: Example 2, sketch of the solution for $\varepsilon = 10^{-8}$, computed with a nonlinear algebraic flux-corrected (AFC) finite element method with Kuzmin limiter, see [3].

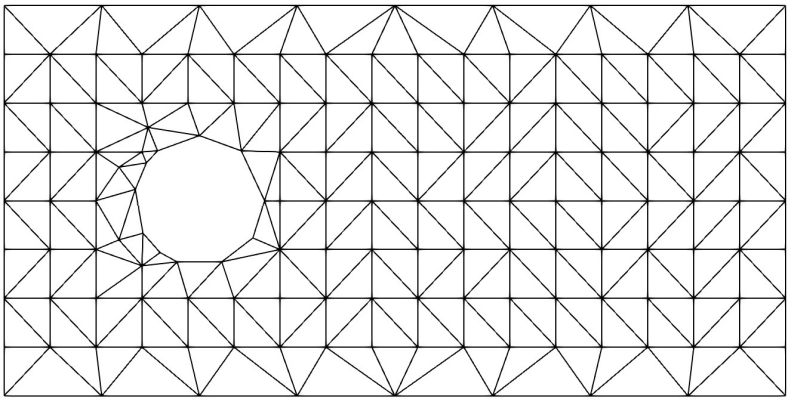

Figure 14: Example 2, initial grid.

Example 2 (*Hemker problem*)**.** The Hemker problem, proposed in [18], is probably the benchmark problem for convection-diffusion equations that possesses most features of problems which can be found in application. It models the transport of energy from a body through a channel. The domain for the Hemker problem is given by $\Omega = \{(-3,9) \times (-3,3)\} \setminus \{(x,y) \ : \ x^2 + y^2 \leq 1\}$, and the coefficients by $\boldsymbol{b} = (1,0)^T$, $c = f = 0$. Dirichlet boundary conditions are prescribed at $x = -3$, with $g = 0$, and at the circular boundary with $g = 1$. On all other boundaries, homogeneous Neumann conditions are used. A sketch of the solution is presented in Figure 13. The solution takes values in $[0,1]$. Boundary layers appear in front of the interior boundary and interior layers in the direction of the convection starting at the body.

For the sake of brevity, only results on triangular grids and for a strongly convection-dominated regime with $\varepsilon = 10^{-8}$ are presented. The initial grid for the simulations, consisting 259 of triangles, is shown in Figure 14.

Concerning $\mathrm{osc}_{\max}(u_h)$, see Figure 15, only *ConstJumpMod* was usually able to compute better solutions than *Galerkin* and it was never worse than *Galerkin*. Again, *LinTriaReco* often increased the maximal oscillations. With respect to the mean of the oscillations $\mathrm{osc}_{\mathrm{mean}}(u_h)$, Figure 16, *ConstJumpMod* was the best method for P_1. But for higher order elements, the reductions obtained with

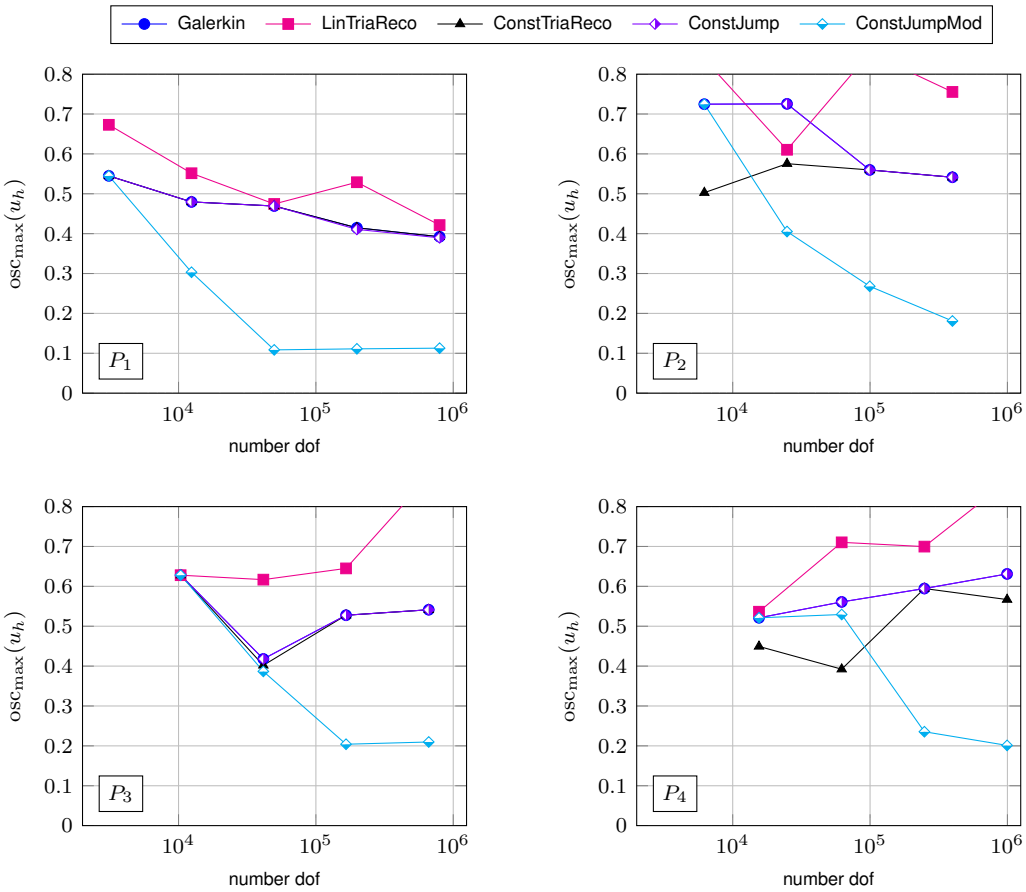

Figure 15: Example 2, triangular grids, $\varepsilon = 10^{-8}$, maximal value of oscillations defined in (14). The results of *ConstJump* may hide the ones of *Galerkin* and *ConstTriaReco*. Results not shown lie out of range of the plot.

ConstTriaReco were often larger. Almost no improvement, compared with *Galerkin*, can be observed for *ConstJump*.

Examples of numerical solutions are presented in Figure 17. Because of the weak imposition of the Dirichlet boundary condition, the exponential layer at the front of the body is not present in these solutions. Spurious oscillations occur above all at the starting points of the interior layers, which can be seen best for the solution computed with P_3 and *ConstTriaReco*.

5 Summary and Outlook

In this paper, a discretization of steady-state convection-diffusion-reaction equations by a DG finite element method was considered. Post-processing methods for reducing the size of the spurious oscillations in the obtained numerical solutions were studied: three methods from the literature and several new modifications and extensions. All these methods are computationally very efficient since they do not require to solve any linear or nonlinear system of equations.

The first step of a systematic numerical assessment was performed. It turned out that none of these methods could remove all spurious oscillations in the considered examples. However, there were always methods that could reduce the size of these oscillations, measured with the maximal value or

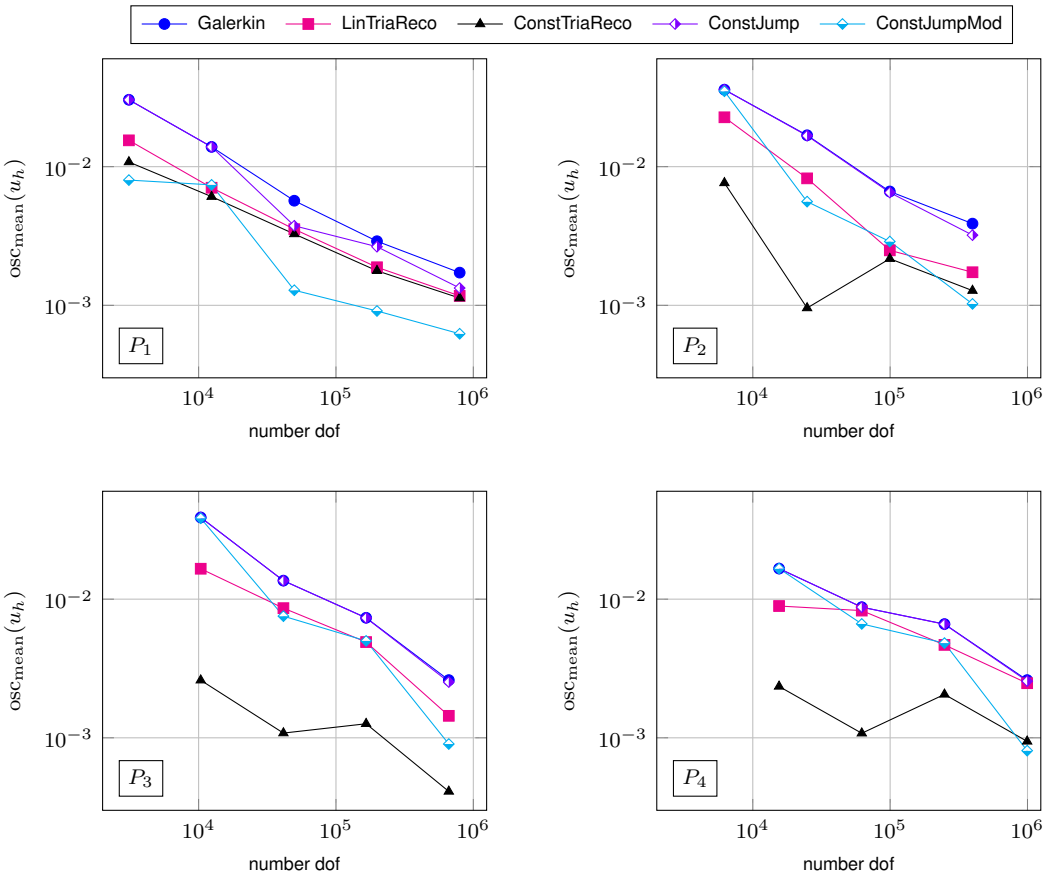

Figure 16: Example 2, triangular grids, $\varepsilon = 10^{-8}$, mean value of oscillations defined in (15). The results of *ConstJump* may hide the ones of *Galerkin*.

with a mean value, considerably. On triangular grids, the method *ConstJumpMod* is the most promising approach, but also *ConstTriaReco* led often to good results. On quadrilateral grids, there are even more methods that behaved similarly well: *ConstQuadReco*, *ConstQuadDeriv*, *LinQuadDeriv*, and *ConstJumpMod*.

Future work will include more studies on quadrilateral grids, which were deferred here because of the length of the paper, and parameter studies for the methods with parameters. Algorithmic changes of methods are possible, e.g., for *ConstTriaReco*. For this method, see Section 3.1, a different choice of the extension to a virtual mesh cell could be a linear extension of the solution at the edge midpoint with a slope given by the integral mean of the gradient of u_h along the edge. Open issues are the extension of *LinQuadDeriv* and *ConstQuadDeriv* for non-affine transforms, see Section 3.2, and the scaling invariance of *ConstJump* and *ConstJumpMod* mentioned at the end of Section 3.3. Because of the computational efficiency, it is also possible to combine methods, e.g., to mark mesh cells with two different methods, which may be examined in the future.

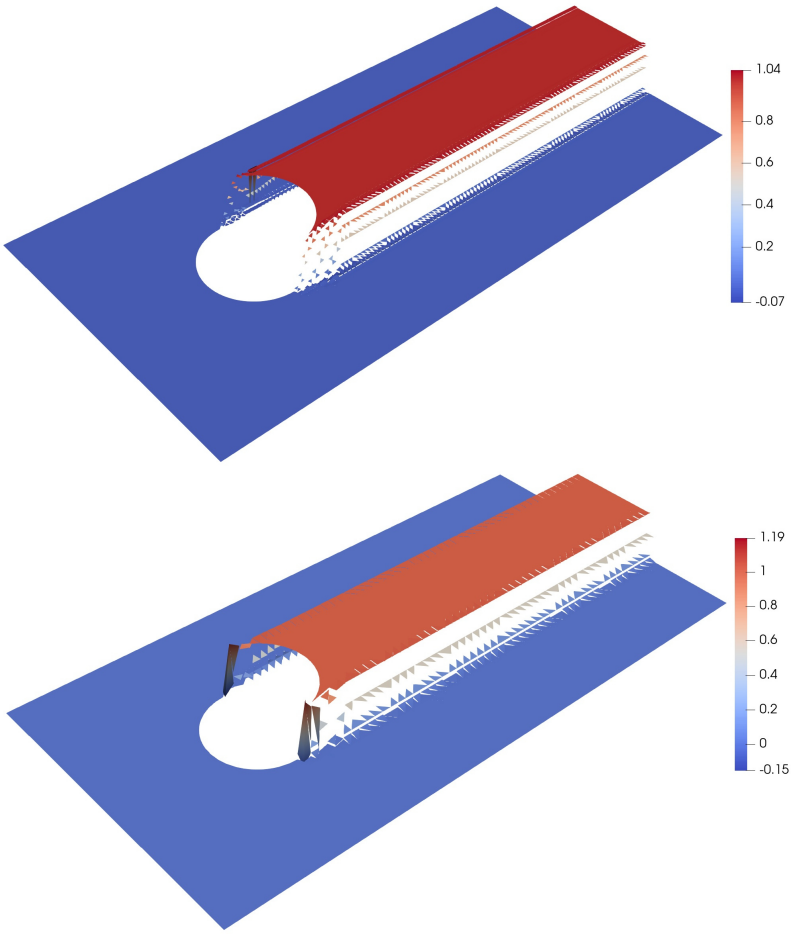

Figure 17: Example 2. Top: solution with P_1 and *ConstJumpMod*. Bottom: solution with P_3 and *ConstTriaReco*.

References

[1] M. Augustin, A. Caiazzo, A. Fiebach, J. Fuhrmann, V. John, A. Linke, and R. Umla. An assessment of discretizations for convection-dominated convection-diffusion equations. *Comput. Methods Appl. Mech. Engrg.*, 200(47-48):3395–3409, 2011.

[2] B. Ayuso and L. D. Marini. Discontinuous Galerkin methods for advection-diffusion-reaction problems. *SIAM J. Numer. Anal.*, 47(2):1391–1420, 2009.

[3] G. R. Barrenechea, V. John, P. Knobloch, and R. Rankin. A unified analysis of algebraic flux correction schemes for convection-diffusion equations. *SeMA J.*, 75(4):655–685, 2018.

[4] S. C. Brenner and L. R. Scott. *The mathematical theory of finite element methods*, volume 15 of *Texts in Applied Mathematics*. Springer, New York, third edition, 2008.

[5] A. N. Brooks and T. J. R. Hughes. Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Comput. Methods Appl. Mech. Engrg.*, 32(1-3):199–259, 1982. FENOMECH '81, Part I (Stuttgart, 1981).

[6] A. Buffa, T. J. R. Hughes, and G. Sangalli. Analysis of a multiscale discontinuous Galerkin method for convection-diffusion problems. *SIAM J. Numer. Anal.*, 44(4):1420–1440, 2006.

[7] A. Cangiani, Z. Dong, E. H. Georgoulis, and P. Houston. *hp-version discontinuous Galerkin methods on polygonal and polyhedral meshes*. SpringerBriefs in Mathematics. Springer, Cham, 2017.

[8] P. G. Ciarlet. *The finite element method for elliptic problems*. North-Holland Publishing Co., Amsterdam, 1978. Studies in Mathematics and its Applications, Vol. 4.

[9] B. Cockburn and C.-W. Shu. The Runge-Kutta discontinuous Galerkin method for conservation laws V: Multidimensional systems. *Journal of Computational Physics*, 141(2):199 – 224, 1998.

[10] T. A. Davis. Algorithm 832: UMFPACK V4.3—an unsymmetric-pattern multifrontal method. *ACM Trans. Math. Software*, 30(2):196–199, 2004.

[11] D. A. Di Pietro and A. Ern. *Mathematical aspects of discontinuous Galerkin methods*, volume 69 of *Mathématiques & Applications (Berlin) [Mathematics & Applications]*. Springer, Heidelberg, 2012.

[12] V. Dolejší and M. Feistauer. *Discontinuous Galerkin method*, volume 48 of *Springer Series in Computational Mathematics*. Springer, Cham, 2015. Analysis and applications to compressible flow.

[13] V. Dolejší, M. Feistauer, and C. Schwab. On discontinuous Galerkin methods for nonlinear convection-diffusion problems and compressible flow. *Mathematica Bohemica*, 127(2):163–179, 2002. Proceedings of EQUADIFF 10.

[14] V. Dolejší, M. Feistauer, and C. Schwab. On some aspects of the discontinuous Galerkin finite element method for conservation laws. *Mathematics and Computers in Simulation*, 61(3):333 – 346, 2003. MODELLING 2001 - Second IMACS Conference on Mathematical Modelling and Computational Methods in Mechanics, Physics, Biomechanics and Geodynamics.

[15] V. Dolejší and P. Solin. *hp*-discontinuous Galerkin method based on local higher order reconstruction. *Appl. Math. Comput.*, 279:219–235, 2016.

[16] S. Ganesan, V. John, G. Matthies, R. Meesala, S. Abdus, and U. Wilbrandt. An object oriented parallel finite element scheme for computing pdes: Design and implementation. In *IEEE 23rd International Conference on High Performance Computing Workshops (HiPCW) Hyderabad*, pages 106–115. IEEE, 2016.

[17] J. Gopalakrishnan and G. Kanschat. A multilevel discontinuous Galerkin method. *Numer. Math.*, 95(3):527–550, 2003.

[18] P. W. Hemker. A singularly perturbed model problem for numerical computation. *J. Comput. Appl. Math.*, 76(1-2):277–285, 1996.

[19] P. Houston, C. Schwab, and E. Süli. Discontinuous *hp*-finite element methods for advection-diffusion-reaction problems. *SIAM J. Numer. Anal.*, 39(6):2133–2163, 2002.

[20] T. J. R. Hughes and A. Brooks. A multidimensional upwind scheme with no crosswind diffusion. In *Finite element methods for convection dominated flows (Papers, Winter Ann. Meeting Amer. Soc. Mech. Engrs., New York, 1979)*, volume 34 of *AMD*, pages 19–35. Amer. Soc. Mech. Engrs. (ASME), New York, 1979.

[21] T. J. R. Hughes, M. Mallet, and A. Mizukami. A new finite element formulation for computational fluid dynamics. II. Beyond SUPG. *Comput. Methods Appl. Mech. Engrg.*, 54(3):341–355, 1986.

[22] V. John and P. Knobloch. On spurious oscillations at layers diminishing (SOLD) methods for convection-diffusion equations. I. A review. *Comput. Methods Appl. Mech. Engrg.*, 196(17-20):2197–2215, 2007.

[23] V. John, P. Knobloch, and J. Novo. Finite elements for scalar convection-dominated equations and incompressible flow problems: a never ending story? *Comput. Vis. Sci.*, 19(5-6):47–63, 2018.

[24] V. John, P. Knobloch, and S. B. Savescu. A posteriori optimization of parameters in stabilized methods for convection-diffusion problems—Part I. *Comput. Methods Appl. Mech. Engrg.*, 200(41-44):2916–2929, 2011.

[25] G. Kanschat. *Discontinuous Galerkin Methods for Viscous Incompressible Flow*. Teubner Research : Deutscher UniversitaÌĹts-Verlag, 2007.

[26] P. Knobloch, P. Lukáš, and P. Solin. On error indicators for optimizing parameters in stabilized methods. *Adv. Comput. Math.*, 45(4):1853–1862, 2019.

[27] W. Reed and T. Hill. Triangular mesh methods for the neutron transport equation. Technical Report LA-UR-73-479, Los Alamos Scientific Laboratory, Los Alamos, NM, 1973.

[28] B. Rivière. *Discontinuous Galerkin methods for solving elliptic and parabolic equations*, volume 35 of *Frontiers in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2008. Theory and implementation.

[29] H.-G. Roos, M. Stynes, and L. Tobiska. *Robust numerical methods for singularly perturbed differential equations*, volume 24 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, second edition, 2008. Convection-diffusion-reaction and flow problems.

[30] A. M. P. Valli, R. C. Almeida, I. P. Santos, L. Catabriga, S. M. C. Malta, and A. L. G. A. Coutinho. A parameter-free dynamic diffusion method for advection-diffusion-reaction problems. *Comput. Math. Appl.*, 75(1):307–321, 2018.

[31] U. Wilbrandt, C. Bartsch, N. Ahmed, N. Alia, F. Anker, L. Blank, A. Caiazzo, S. Ganesan, S. Giere, G. Matthies, R. Meesala, A. Shamim, J. Venkatesan, and V. John. ParMooN—A modernized program package based on mapped finite elements. *Comput. Math. Appl.*, 74(1):74–88, 2017.