Numerical solutions of systems with (p, δ)-structure using local discontinuous Galerkin finite element methods

Dietmar Kröner[1], Michael Růžička[1], and Ioannis Toulopoulos[2]

[1] Abteilung für Angewandte Mathematik Universität Freiburg,
[2] Johann Radon Institute for Computational and Applied Mathematics, Austrian Academy of Sciences
dietmar@mathematik.uni-freiburg.de
rose@mathematik.uni-freiburg.de
ioannis.toulopoulos@oeaw.ac.at,

Abstract. *In this paper we present local discontinuous Galerkin methods for systems with (p, δ)-structure. The unknown gradient and the nonlinear diffusivity function are introduced as auxiliary variables and the original (p, δ) system is decomposed into a first order system. Every equation of the produced first order system is discretized in the discontinuous Galerkin framework, where two different nonlinear viscous numerical fluxes are implemented. An a priori bound for a simplified problem is derived. The ODE system resulting from the LDG discretization is solved by Diagonal Implicit Runge-Kutta methods. The non linear system of algebraic equations with unknowns the intermediate solutions of the Runge-Kutta cycle, is solved using Newton and Picard iterative methodology. The performance of the two non linear solvers is compared on simple test problems. Numerical tests concerning problems with exact solutions are performed in order to validate the theoretical spatial accuracy of the proposed method. Further, more realistic numerical examples are solved in domains with non-smooth boundary to test the efficiency of the method.*

Key words: Local discontinuous Galerkin methods, (p, δ)-structure system of equations, (p, δ)-structure penalty jump terms, Newton-Picard iterative methods, numerical solutions in domains with non-smooth boundary.

1 Introduction

In this paper we present some computational issues of approximating solutions of systems with (p, δ)-structure. This type of equations appear as a mathematical model describing several physical problems such as non-Newtonian flows, plasticity and glaciology [30], [39]. The most common p-type problem of the present general (p, δ) model is the *p-Laplace* problem, for which $\delta = 0$. There are several contributions analyzing the discretization of p-Laplace equations, e.g. [38] using finite differences, [27], [33] using mixed finite element-volume methods. The first error analysis for the approximation of the p-Laplace solutions

by finite element methods has been presented in [29], and Ciarlet in [15], chp. 5, presented error estimates for the case of $p \geq 2$, by treating the $p-$ Laplace operator in a general class of monotone operators. Barrett and Liu in several papers, e.g. [3], [4], [5], introduce a so-called *quasi norm* for the error between the exact and the approximate solution and proved inproved error estimates. In [21], interpolation operators in Orlicz-Sobolev spaces were studied and were utilized in finite element methods for approximating solutions of (p, δ)-structure problems. Linear convergence rate of the method has been shown in case of using linear base polynomials.

During the last two decades, there is an intense effort towards devising local DG methods (LDG) for linear and non-linear elliptic problems. The main feature of LDG methods is the introduction of the gradient (flux) of the solution, say $\mathbf{L} = \nabla \mathbf{u}$, and the nonlinear diffusion term $\mathbf{A} = \tilde{A}(\mathbf{L})\mathbf{L}$, as new variables and to rewrite the original problem as a first order system, (a technique similar to the mixed finite element methods, however LDG methods are different in the discretization procedure). The resulting system is then solved in DG framework, where the variables $\mathbf{L}$, $\mathbf{A}$, $\mathbf{u}$ are approximated using the same order local spaces.

LDG methods were firstly proposed in [16] for convection - diffusion systems, based on the numerical approach applied in [6] for the discretization of the viscous fluxes of compressible Navier-Stokes equations. We can say, that this was the starting point of a systematic study of LDG methods for linear elliptic problems. Indicatively we refer to the following papers, [14] where the first a priori analysis was presented, [13] where the performance of several LDG methods for a model problem is shown, and for a detailed review [2], where LDG methods are presented as a particular choice of DG methods. Recently, LDG methods have been proposed and analyzed for applications to nonlinear elliptic problems. In [10], [9], [8] a theoretical study of LDG methods is presented for problems with mixed boundary conditions, under the assumption that the diffusion operator is monotone and has $p = 2$-structure. Moreover, Santillana and Dawson studied in [42] the applicability of LDG methods for nonlinear diffusion shallow water equations.

The objective of this study, is to present s to present a complete framework for the discretization of (p, δ)-structure systems using LDG methods. We generalize the nonlinear penalty jump terms of the numerical flux proposed in [7] for the case of $\delta = 0$ to $\delta \geq 0$ (see (11b)), so that to be compatible with the (p, δ)-structure of the problem. Furthermore, we apply new penalty jump terms (see (13)) which, instead of the jump of $\mathbf{u}_h$, use the trace of $\mathbf{L}_h$ in order to compute the "diffusivity" on the inter-element boundaries. Comparisons of the numerical results computed by (11b) and the new numerical flux (13) are shown in the last section of the paper. The problem is discretized in time using Diagonal Implicit Runge-Kutta methods. Initially, the nonlinear algebraic system with unknowns the intermediate solutions of one Runge-Kutta cycle is solved using a Newton method combined with matrix-free GMRES linear iterative solver. Despite the fact that this nonlinear iterative solution approach achieved fast convergence in many test problems, the performance speed of the method appeared to be de-

pendent on the values of the problem parameters. In particular, when we solve the problem on fine meshes, the CPU time of $p < 2$ test cases is significantly increased in comparison to the CPU time of $p \geq 2$ test cases. This has motivated us to develop and apply a Picard type method which is computationally less expensive and non parameter dependent, in the sense that the convergence speed performance is similar for all p test cases. In order to increase the converge speed of the Picard method, a new idea of applying nested Jacobi type iterations is utilized. Comparisons concerning the performance of the two different nonlinear iterative processes are shown in the numerical examples. We point out that, many real fluids are covered by the range of values for p considered in the numerical experiments, [35]. Moreover, the (p,δ)-structure ensures that many popular models in the engineering literature are contained in our investigation.

The outline of the paper is as follows. We begin by presenting the model problem in Section 2. We continue in the third Section, with the definition of LDG spaces, the numerical fluxes and finally the semi-discrete analogue of the problem. In the last paragraph a stability bound is given for a scalar model problem. In Section 4, we present the time marching scheme based on Diagonal implicit Runge-Kutta methods. A detailed description of the implementation of the Newton and the Picard non linear iterative processes is also included. Last, in Section 5, we present the experimental error convergence rates for different test problems. For first order polynomial space we confirm the theoretical results of the error analysis presented in [20], [13], [10], [9]. Also, we give comparisons for the two nonlinear iterative solvers. In the last part of the Section, we present numerical tests in domains with corner singular boundary points aimed to investigate the efficiency of the LDG method on realistic diffusive problems.

2 (p,δ)-Structure system: The model problem

Let Ω be a bounded domain in $\mathbb{R}^d$, $d = 2, 3$ with polygonal (polyhedral) Lipschitz boundary $\partial\Omega$ which is decomposed into $\partial\Omega = \Gamma_N \bigcup \Gamma_D$ with $\Gamma_N \bigcap \Gamma_D = \emptyset$, $|\Gamma_D| \neq \emptyset$ and let $(0,T]$ be a time interval. We consider the following system

$$\mathbf{u}_t - \mathrm{div}\mathbf{A}(\nabla\mathbf{u}) = \mathbf{f}, \qquad \text{in } \Omega \times (0,T] \tag{1a}$$

$$\mathbf{u}(.,0) = \mathbf{u}_0, \qquad \text{in } \Omega, \tag{1b}$$

$$\mathbf{u} = \mathbf{u}_D \qquad \text{on } \Gamma_D \times (0,T], \tag{1c}$$

$$\mathbf{A}(\nabla\mathbf{u}) \cdot \mathbf{n} = \mathbf{a}_N \qquad \text{on } \Gamma_N \times (0,T], \tag{1d}$$

where

$$\mathbf{A}(\nabla\mathbf{u}) = \tilde{A}(\nabla\mathbf{u})\nabla\mathbf{u} = (\delta + |\nabla\mathbf{u}|)^{p-2}\nabla\mathbf{u}. \tag{2}$$

Here $\mathbf{f} : \Omega \times (0,T] \to \mathbb{R}, \mathbf{u}_D : \Gamma_D \times (0,T] \to \mathbb{R}, \mathbf{a}_N : \Gamma_N \times (0,T] \to \mathbb{R}, \mathbf{u}_0 : \Omega \to \mathbb{R}$ are given appropriately smooth functions, $\mathbf{u} = (u_1, ..., u_{d_u})^\top$ is the unknown vector where through this paper we consider the case of $d = d_u$, $\nabla\mathbf{u}$ is the tensor gradient of $\mathbf{u}$, $\delta \geq 0$ is a parameter and $\tilde{A} : \mathbb{R}^{d\times d} \to \mathbb{R}^+$ is the *diffusivity function* given by $\tilde{A}(\nabla\mathbf{u}) = (\delta + |\nabla\mathbf{u}|)^{p-2}$. Our motivation for developing LDG methods

for the model problem (1), arises from our interest to develop efficient methods for non-Newtonian incompressible fluids modeled by $p-$power low systems, [37], [35]. For $\delta = 0$ and $p < 2$, in case of $|\nabla \mathbf{u}| \to 0$, model (1) leads to a singular system. For non-Newtonian fluids this possesses an infinite zero shear limit, while for $\delta > 0$ the zero shear limit is finite. Thus choosing a relative small value for the parameter δ in (2), we approximate the solution of the singular model problem. Results concerning existence and regularity of the solution of (1) and the interpretation of the time derivative can be found in [36], [17], [11]. A detailed discussion about these results is out of the context of the present work. The main goal here is to show how the problem (1) can be discretized by the LDG method.

In order to obtain the LDG formulation, we introduce auxiliary variables $\mathbf{L} = \nabla \mathbf{u}$ and $\mathbf{A} = \tilde{A}(\mathbf{L})\mathbf{L}$ and we rewrite the system (1) as a first order system

$$\mathbf{L} = \nabla \mathbf{u}, \qquad \text{in } \Omega \times (0,T], \tag{3a}$$

$$\mathbf{A} = \tilde{A}(\mathbf{L})\mathbf{L}, \quad \text{in } \Omega \times (0,T], \tag{3b}$$

$$\mathbf{u}_t - \operatorname{div}(\mathbf{A}) = \mathbf{f}, \qquad \text{in } \Omega \times (0,T]. \tag{3c}$$

In that way, problem (1) is reformulated as (3) and consequently we have to approximate $\mathbf{L}, \mathbf{A}, \mathbf{u}$ in an appropriate way.

Remark 1. We note that the operator $\mathbf{A}$ falls in the general class of (p,δ)-structure operators, [22], [19], this means that there exist $p \in (1,\infty), \delta \in [0,\infty)$ and constants C_0, C_1 such that

$$\sum_{j,l=1}^{d} \sum_{i,k=1}^{d} \partial_{kl} A_{ij}(\mathbf{P}) Q_{ij} Q_{kl} \geq C_0 (\delta + |\mathbf{P}|)^{p-2} |\mathbf{Q}|^2, \tag{4a}$$

$$|\partial_{kl} A_{ij}(\mathbf{P})| \leq C_1 (\delta + |\mathbf{P}|)^{p-2}, \tag{4b}$$

are satisfied for all $\mathbf{P},\ \mathbf{Q} \in \mathbf{R}^{d\times d}$ with $\mathbf{P} \neq \mathbf{0}$ and all $i,k,j,l = 1,...,d$.

3 The LDG method

3.1 Preliminaries - DG notation

For $\mathbf{u}, \mathbf{v} \in \mathbb{R}^d$ and $\mathbf{A} := (A_{ij}), \mathbf{T} := (T_{ij}) \in \mathbb{R}^{d\times d}$ we use standard notation, $\mathbf{u} \otimes \mathbf{v} = u_i v_j, \ \in \mathbb{R}^{d\times d}$, $\mathbf{A} : \mathbf{T} = \sum_{i,j=1}^{d} A_{ij} T_{ij}$, $\mathbf{u} \cdot (\mathbf{A}\mathbf{v}) = \sum_{i,j=1}^{d} u_i A_{ij} v_j = \mathbf{A} : (\mathbf{u} \otimes \mathbf{v})$.

Let $\mathcal{T}_h = \{E_i\}_{i=1}^{N_E}$ be a regular subdivision of Ω in triangular (or tetrahedral) elements with diameter h_{E_i} and $h = max_{E_i \in \mathcal{T}_h} h_{E_i}$. We denote the collection of the edges (or faces) of the elements by $\mathcal{E}_h = \{e\}_{i=1}^{N_e}$. We group the edges into three sets. The set $\mathcal{E}_I$ of the interior edges, $\mathcal{E}_I := \{e : e \in \mathcal{E}_h - \partial\Omega\}$. For $e \in \mathcal{E}_I$ there are two adjacent elements E_1, E_2 that share e. The set of the Dirichlet boundary edges $\mathcal{E}_D := \{e \in \mathcal{E}_h : e \cap \Gamma_D \neq \emptyset\}$ and the set of the Neumann boundary edges $\mathcal{E}_N := \{e \in \mathcal{E}_h : e \cap \Gamma_N \neq \emptyset\}$.

We denote by $\mathbb{P}^k, k \in \mathbf{N}_0$, the space of scalar olynomials of degree less than or equal to k and by p' the conjugate exponent $\frac{1}{p}+\frac{1}{p'}=1$. On $\mathcal{T}_h$, we define the discontinuous finite element spaces

$$(V_h^k)^d := \{\mathbf{v}_h \in (L^p(\Omega))^d : \mathbf{v}_h|_E \in (\mathbb{P}^k(E))^d, \forall E \in \mathcal{T}_h\}, \tag{5}$$

$$(Y_h^k)^{d\times d} := \{\mathbf{L}_h \in (L^p(\Omega))^{d\times d} : \mathbf{L}_h|_E \in (\mathbb{P}^k(E))^{d\times d}, \forall E \in \mathcal{T}_h\}, \tag{6}$$

$$(X_h^k)^{d\times d} := \{\mathbf{A}_h \in (L^{p'}(\Omega))^{d\times d} : \mathbf{A}_h|_E \in (\mathbb{P}^k(E))^{d\times d}, \forall E \in \mathcal{T}_h\}. \tag{7}$$

Interface jumps and averages: For a function $\mathbf{v} \in (V_h^k)^d$ we do not impose any continuity requirements on the interfaces of the elements. Thus, for two elements $E_1,\ E_2$ which share a common edge $e = \partial E_1 \bigcap \partial E_2$, we assume the outward normal vector $\mathbf{n}_{e,12}$ to be oriented from E_1 towards the interior of E_2 and conversely $\mathbf{n}_{e,21}$ from E_2 towards the interior of E_1. We define the average and the jump respectively of $\mathbf{v}$ on e by

$$\{\mathbf{v}\} := \frac{1}{2}(\mathbf{v}|_{E_1} + \mathbf{v}|_{E_2}),\ [\![\mathbf{v}\otimes\mathbf{n}]\!] := \mathbf{v}|_{E_1}\otimes\mathbf{n}_{e,12} + \mathbf{v}|_{E_2}\otimes\mathbf{n}_{e,21}, \tag{8}$$

$$\{\mathbf{A}\} := \frac{1}{2}(\mathbf{A}|_{E_1} + \mathbf{A}|_{E_2}),\ [\![\mathbf{A}\mathbf{n}]\!] := \mathbf{A}|_{E_1}\mathbf{n}_{e,12} + \mathbf{A}|_{E_2}\mathbf{n}_{e,21}. \tag{9}$$

3.2 The LDG discretization

We multiply each equation in (3) by a test functions $(\mathbf{X}_h, \mathbf{Y}_h, \mathbf{z}_h) \in ((X_h^k)^{d\times d} \times (Y_h^k)^{d\times d} \times (V_h^k)^d)$ respectively, integrate over one element $E \in \mathcal{T}_h$ and use partial integration to obtain the discrete formulation: we look for the LDG approximations $\mathbf{L}_h \in (Y_h^k)^{d\times d},\ \mathbf{A}_h \in (X_h^k)^{d\times d},\ \mathbf{u}_h \in (V_h^k)^d$ of $(\mathbf{L}, \mathbf{A}, \mathbf{u})$ of (3) such that for $E \in \mathcal{T}_h$ and $t \in (0,T]$ the following equations are satisfied

$$\int_E \mathbf{L}_h : \mathbf{X}_h\, dx = -\int_E \mathbf{u}_h \cdot \mathrm{div}\mathbf{X}_h\, dx + \int_{\partial E} \widehat{\mathbf{u}}_h \cdot (\mathbf{X}_h\mathbf{n})\, ds, \tag{10a}$$

$$\int_E \mathbf{A}_h : \mathbf{Y}_h\, dx = \int_E \tilde{A}(\mathbf{L}_h)\mathbf{L}_h : \mathbf{Y}_h\, dx, \tag{10b}$$

$$\int_E \mathbf{u}_{ht}\cdot\mathbf{z}_h\, dx = -\int_E \mathbf{A}_h : \nabla\mathbf{z}_h\, dx + \int_{\partial E} \widehat{\mathbf{A}}_h : (\mathbf{z}_h\otimes\mathbf{n})\, ds + \int_E \mathbf{f}\cdot\mathbf{z}_h\, dx, \tag{10c}$$

where the fluxes $\hat{(.)}$ on ∂E in (10) are the *numerical fluxes*, which must be defined suitably in order to ensure stability and convergence of the method (e.g. see [25], and [34] for comprehensive analysis of the numerical fluxes for a linear elliptic model). As in [20], we choose

$$\widehat{\mathbf{u}_h}^e := \begin{cases} \mathbf{u}_h, & \text{on } e \in \mathcal{E}_N \\ \mathbf{u}_D, & \text{on } e \in \mathcal{E}_D \\ \{\mathbf{u}_h\}, & \text{on } e \in \mathcal{E}_I \end{cases} \tag{11a}$$

$$\widehat{\mathbf{A}_h}^e := \begin{cases} \mathbf{a}_N, & \text{on } e \in \mathcal{E}_N \\ \mathbf{A}_h - \gamma\tilde{A}(\frac{[\![\mathbf{u}_h\otimes\mathbf{n}]\!]}{h})\frac{[\![\mathbf{u}_h\otimes\mathbf{n}]\!]}{h}, & \text{on } e \in \mathcal{E}_D \\ \{\mathbf{A}_h\} - \gamma\tilde{A}(\frac{[\![\mathbf{u}_h\otimes\mathbf{n}]\!]}{h})\frac{[\![\mathbf{u}_h\otimes\mathbf{n}]\!]}{h}, & \text{on } e \in \mathcal{E}_I, \end{cases} \tag{11b}$$

where $\gamma > 0$ is a constant (which will be specified in the numerical examples) and for $e \in \mathcal{E}_D$ the jump is defined as $[\![\mathbf{u}_h \otimes \mathbf{n}]\!] = (\mathbf{u}_h - \mathbf{u}_D) \otimes \mathbf{n}$. The term

$$\gamma \tilde{A}(\frac{[\![\mathbf{u}_h \otimes \mathbf{n}]\!]}{h}) \frac{[\![\mathbf{u}_h \otimes \mathbf{n}]\!]}{h}, \tag{12}$$

is called *penalty jump term*, and complies with the (p, δ)-structure of $\mathbf{A}$, see (2). Here, with the numerical flux (11b), we have generalized the numerical flux from [7], which was proposed for the case $\delta = 0$ to the general case of $\delta \geq 0$.

Another type of numerical flux: In addition, to the numerical flux defined in (11b) we propose a new numerical flux

$$\widehat{\mathbf{A}}^e_{h,\mathbf{L}} := \begin{cases} \mathbf{a}_N, & \text{on } e \in \mathcal{E}_N \\ \mathbf{A}_h - \gamma \tilde{A}(\mathbf{L}_h) \frac{[\![\mathbf{u}_h \otimes \mathbf{n}]\!]}{h}, & \text{on } e \in \mathcal{E}_D \\ \{\mathbf{A}_h\} - \gamma \{\tilde{A}(\mathbf{L}_h)\} \frac{[\![\mathbf{u}_h \otimes \mathbf{n}]\!]}{h}, & \text{on } e \in \mathcal{E}_I, \end{cases} \tag{13}$$

where $\gamma > 0$ is a constant and the diffusivity function $\tilde{A}$ on the edges depends on $\mathbf{L}_h$. The flux (13) increases the performance of the computational procedure. This will be explained in the numerical tests, where we perform comparisons between the results computed by the numerical fluxes (11b) and (13). Here is the first time where the numerical fluxes (11b), (13) are applied in LDG methods for solving problems with (p, δ)-structure. Notice that for the linear case $p = 2$ the two different penalty jump terms are the same.

3.3 Stability bounds for the scalar case

In the scalar case, where $u : \Omega \times (0, T) \to \mathbb{R}$, the model problem (1) with $\delta = 0$ (for $\delta \geq 0$ we refer to [20] for steady problems) has the form

$$u_t - \operatorname{div}\big(\tilde{a}(\nabla u)\nabla u\big) = f \quad \text{in } \Omega \times (0, T], \tag{14a}$$

$$u_0(x) = u(x, 0) \quad \text{in } \Omega, \tag{14b}$$

$$\big(\tilde{a}(\nabla u)\nabla u\big) \cdot \mathbf{n} = a_N \quad \text{on } \Gamma_N \times (0, T], \tag{14c}$$

$$u = u_D, \quad \text{on } \Gamma_D \times (0, T], \tag{14d}$$

where $\tilde{a}(\mathbf{v}) = |\mathbf{v}|^{p-2}$, $\mathbf{v} \in \mathbb{R}^d$. Following the same procedure as in the previous section we can easily obtain the coresponding LDG formulation (10) for the problem (14). Next, using the expression (11) for the numerical fluxes and summing up over all $E \in \mathcal{T}_h$, we obtain after some simple manipulations, see details in [2], the discrete variational formulation: find the LDG approximations $(\mathbf{l}_h, \mathbf{a}_h, u_h) \in ((Y_h^k)^{d\times d} \times (X_h^k)^{d\times d} \times (V_h^k)^d)$ such that the following discrete equa-

tions are satisfied

$$\int_\Omega \mathbf{l}_h \cdot \mathbf{x}_h \, dx = \int_\Omega \nabla u_h \cdot \mathbf{x}_h \, dx - \int_{\mathcal{E}_I} [\![u_h \mathbf{n}]\!] \cdot \{\mathbf{x}_h\} \, ds - \int_{\mathcal{E}_D} u_h \mathbf{x}_h \cdot \mathbf{n} \, ds + \int_{\mathcal{E}_D} u_D \mathbf{x}_h \cdot \mathbf{n} \, ds, \tag{15a}$$

$$\int_\Omega \mathbf{a}_h \cdot \mathbf{y}_h \, dx = \int_\Omega \tilde{a}(\mathbf{l}_h)\mathbf{l}_h \cdot \mathbf{y}_h \, dx, \tag{15b}$$

$$\begin{aligned} \int_\Omega u_{ht} z_h \, dx + \int_\Omega \mathbf{a}_h \cdot \nabla z_h \, dx &= \int_{\mathcal{E}_I} \{\mathbf{a}_h\} \cdot [\![z_h \mathbf{n}]\!] \, ds - \int_{\mathcal{E}_I} \gamma \tilde{a}\Big(\frac{[\![u_h \mathbf{n}]\!]}{h}\Big) \frac{[\![u_h \mathbf{n}]\!]}{h} \cdot [\![z_h \mathbf{n}]\!] \, ds \\ &+ \int_{\mathcal{E}_D} \mathbf{a}_h \cdot \mathbf{n} z_h \, ds - \int_{\mathcal{E}_D} \gamma \tilde{a}\Big(\frac{[\![u_h \mathbf{n}]\!]}{h}\Big) \frac{[\![u_h \mathbf{n}]\!]}{h} \cdot z_h \mathbf{n} \, ds \\ &+ \int_{\mathcal{E}_N} a_N z_h \, ds + \int_\Omega f z_h \, dx, \end{aligned} \tag{15c}$$

for $\mathbf{x}_h \in (X_h^k)^d, \mathbf{y}_h \in (Y_h^k)^d, z_h \in V_h^k$. Next, we derive an a priori bound for the LDG scheme (15) in a special case, namely we assume $\Gamma_N = \emptyset$ and $u_D = 0$ in problem (14). Choosing in (15) $z_h = u_h$, $\mathbf{x}_h = \mathbf{a}_h$, $\mathbf{y}_h = \mathbf{l}_h$ and adding all equations, we obtain after some manipulations

$$\begin{aligned} \int_\Omega u_{ht} u_h \, dx + \int_\Omega \tilde{a}(\mathbf{l}_h)|\mathbf{l}_h|^2 \, dx + \gamma \int_{\mathcal{E}_I} \tilde{a}(\frac{[\![u_h \mathbf{n}]\!]}{h}) \frac{[\![u_h \mathbf{n}]\!]^2}{h} \, ds \\ + \gamma \int_{\mathcal{E}_D} \tilde{a}(\frac{[\![u_h \mathbf{n}]\!]}{h}) \frac{[\![u_h \mathbf{n}]\!]^2}{h} \, ds = \int_\Omega f u_h \, dx. \end{aligned} \tag{16}$$

By virtue of the form of $\tilde{a}$ and using that $\tilde{a}(\frac{[\![u_h \mathbf{n}]\!]}{h}) \frac{[\![u_h \mathbf{n}]\!]^2}{h} = \frac{1}{h^{p-1}} |[\![u_h \mathbf{n}]\!]|^p$ on $e \in \mathcal{E}_I \cup \mathcal{E}_D$, we have that

$$\int_\Omega u_{ht} u_h \, dx + \frac{\gamma}{h^{p-1}} \int_{\mathcal{E}_I \cup \mathcal{E}_D} |[\![u_h \mathbf{n}]\!]|^p \, ds + \int_\Omega |\mathbf{l}_h|^p \, dx = \int_\Omega f u_h \, dx. \tag{17}$$

Next, applying Cauchy-Schwartz inequality on the right-hand side of (17) and using that $u_{ht} u_h = \frac{1}{2} \frac{\partial}{\partial t} (u_h(t))^2$ we have that

$$\frac{1}{2} \frac{d}{dt} \|u_h(t)\|_{L^2(\Omega)}^2 + \|\mathbf{l}_h\|_{L^p(\Omega)}^p + \frac{\gamma}{h^{p-1}} \|[\![u_h \mathbf{n}]\!]\|_{L^p(\mathcal{E}_I \cup \mathcal{E}_D)}^p \leq \frac{1}{2} \|f\|_{L^2(\Omega)}^2 + \frac{1}{2} \|u_h\|_{L^2(\Omega)}^2. \tag{18}$$

Integrating from 0 to t yields

$$\|u_h(t)\|_{L^2(\Omega)}^2 + 2 \int_0^t \|\mathbf{l}_h(\tau)\|_{L^p(\Omega)}^p + \frac{\gamma}{h^{p-1}} \|[\![u_h(\tau) \mathbf{n}]\!]\|_{L^p(\mathcal{E}_I \cup \mathcal{E}_D)}^p \, d\tau \leq \int_0^t \|f(\tau)\|_{L^2(\Omega)}^2 + \|u_h(\tau)\|_{L^2(\Omega)}^2 \, d\tau + \|u_h(0)\|_{L^2(\Omega)}^2. \tag{19}$$

Applying Gronwall's inequality in (19) we can obtain the following a priori bound for the LDG solutions of (15),

$$\|u_h(t)\|^2_{L^2(\Omega)} + 2\int_0^t \|\mathbf{l}_h(\tau)\|^p_{L^p(\Omega)} + \frac{\gamma}{h^{p-1}}\|[\![u_h(\tau)\mathbf{n}]\!]\|^p_{L^p(\mathcal{E}_I\cup\mathcal{E}_D)}\,d\tau \\ \leq e^t\Big(\int_0^t \|f(\tau)\|^2_{L^2(\Omega)}\,d\tau + \|u_h(0)\|^2_{L^2(\Omega)}\Big). \quad (20)$$

4 Time marching scheme

We denote by $\mathbf{S}(t) = [\mathbf{L}(t), \mathbf{A}(t), \mathbf{U}(t)]^\top$ the vector with the global degrees of freedom of the expressions of $(\mathbf{L}_h, \mathbf{A}_h, \mathbf{u}_h)$ in $((X_h^k)^{d\times d} \times (Y_h^k)^{d\times d} \times (V_h^k)^d)$. By the discrete formulation (10), we obtain the following nonlinear ODE system,

$$\begin{pmatrix} 0 \\ 0 \\ M\frac{d\mathbf{U}(t)}{dt} \end{pmatrix} = \begin{pmatrix} \mathbf{R}_1(t,\mathbf{S}(t)) \\ \mathbf{R}_2(t,\mathbf{S}(t)) \\ \mathbf{R}_3(t,\mathbf{S}(t)) \end{pmatrix}, \quad (21)$$

where M is the mass matrix of $\mathbf{u}_h$, and the components of $\mathbf{R}_1, \mathbf{R}_2, \mathbf{R}_3$ are derived from the variational formulation (10). We discretize the system (21) by applying s-stage Diagonal Implicit Runge-Kutta methods (sDIRK) [40], [12]. In sDIRK methods the coefficient matrix $[a_{ij}] \in \mathbb{R}^{s\times s}$ of Butcher's table is lower triangular with equal diagonal elements, offering a significant computational advantage, since for every time step s decoupled ODEs systems must be solved.

If $\tau_1, ..., \tau_s$ are the quadrature points and $b_1, ..., b_s$ are their weights, supposing that we have computed the solution $\mathbf{S}^n$ at time step t^n, then the solution $\mathbf{S}^{n+1}$ at time step $t^{n+1} = t^n + \Delta t$, is computed by the following formula

$$0 = \mathbf{R}_1(t^{n+1}, \mathbf{S}^{n+1}), \quad (22a)$$
$$0 = \mathbf{R}_2(t^{n+1}, \mathbf{S}^{n+1}), \quad (22b)$$
$$\mathbf{U}^{n+1} = \mathbf{U}^n + \Delta t\sum_{i=1}^s b_i M^{-1}\mathbf{R}_3(t^{n,i}, \mathbf{S}^{n,i}), \quad (22c)$$

where $\mathbf{S}^{n,i}$ are intermediate solutions at $t^{n,i} = t^n + \tau_i\Delta t$, given by

$$0 = \mathbf{R}_1(t^{n,i}, \mathbf{S}^{n,i}), \quad (23a)$$
$$0 = \mathbf{R}_2(t^{n,i}, \mathbf{S}^{n,i}), \quad (23b)$$
$$\mathbf{U}^{n,i} = \mathbf{U}^n + \Delta t\sum_{j=1}^i a_{ij} M^{-1}\mathbf{R}_3(t^{n,j}, \mathbf{S}^{n,i}). \quad (23c)$$

The computation of the intermediate solutions, requires the solution of the nonlinear system (in the general case of $p \neq 2$), which is achieved by an *iterative process*. Next, we develop and apply two well known iterative processes, *the Newton method* (the most widespread second order iterative method) and the *Picard method* (first order convergence rate), see details in [26].

Newton iterative process. The Newton method for the system (23) takes the form

$$\begin{pmatrix} \frac{\partial \mathbf{R}_1}{\partial \mathbf{S}} \\ \frac{\partial \mathbf{R}_2}{\partial \mathbf{S}} \\ M - \Delta t a_{ii} \frac{\partial \mathbf{R}_3}{\partial \mathbf{S}} \end{pmatrix} \Delta \mathbf{S}^{n,i,k+1} = \begin{pmatrix} \mathbf{R}_{1,S} \\ \mathbf{R}_{2,S} \\ \Delta t \mathbf{R}_{3,S} \end{pmatrix}, \ k = 0, 1, 2, ... \tag{24a}$$

$$\Delta \mathbf{S}^{n,i,k+1} = \mathbf{S}^{n,i,k+1} - \mathbf{S}^{n,i,k}, \tag{24b}$$

where $\frac{\partial \mathbf{R}_j}{\partial \mathbf{S}}$, $\mathbf{R}_{j,S}, j = 1, 2, 3$ are the associated Jacobian matrices and the resulting residual vectors, which are calculated at $\mathbf{S}^{n,i,k}$. In the materialization of (24), the Jacobian matrices are not computed explicitly but are replaced by numerical approximations (matrix-free implementation). Consequently a Krylov subspace projection method for non-symmetric systems (GMRES) is applied, see [41], in order to approximate the solution of the derived linear system. Further details, for implementing sDIRK methods for nonlinear equations are presented in [18]. In the numerical computations, $(k+1)$ order sDIRK method is applied.
We mention, that several multigrid or preconditioned techniques have been proposed in the literature for solving nonlinear systems similar to (24) produced by the discretization of p-type problems as (1), [32]. In our numerical examples we do not use any particular of these techniques.

The computational effort of the whole method depends strongly on the number of iterations of the Newton process, due to the fact that in every iteration k the Jacobian matrices in (24) must be computed. Also, it is important for the efficiency of the iterative process, the solution of the linear system (23) to be obtained in few GMRES iterations. It is known that, in many nonlinear problems, the condition number of the resulting Jacobian matrices is very large and as a result the convergence speed of the iterative solver is very slow (or even the iterative solver can fail), [26]. During the performance of the numerical examples, we saw that the performance speed of the iterative process (Newton and GMRES) is affected by the value of γ and p. Specifically, we saw that choosing $\gamma > 2$ for the $p < 2$ test cases, the CPU time is increased at a higher rate compared to the $p \geq 2$ test cases. For this reason, we develop and apply a Picard type method, which is less costly (but still first order), and more stable in the sense that the performance speed is not strongly affected by the parameter γ and is similar for all p test cases.

Picard method. The Picard iteration scheme for (23) is expressed as following

$$A(\mathbf{S}^{n,i,k})\mathbf{S}^{n,i,k+1} = \mathbf{F}, \ k = 0, 1, 2, ... \tag{25a}$$

The coefficient matrix A in block-form can be written as

$$\begin{bmatrix} \tilde{M} & 0 & B \\ \tilde{A}(\mathbf{L}^{n,i,k}) & \tilde{M} & 0 \\ 0 & \Delta t a_{ii} B^\top & C(\mathbf{L}^{n,i,k}) \end{bmatrix} \mathbf{S}^{n,i,k+1} = \begin{pmatrix} \mathbf{F}_1 \\ 0 \\ \mathbf{F}_3 \end{pmatrix}, \tag{25b}$$

where $\tilde{M}$ is the mass matrix of $\mathbf{L}_h$, the matrices $B, \tilde{A}(\mathbf{L}^{n,i-1})$ are defined by the discrete variational formulation (10), the components $\mathbf{F}_1, \mathbf{F}_2$ are derived by the formula (23), and $C(\mathbf{L}^{n,i,k}) = M - \Delta t a_{ii} \tilde{A}(\{\mathbf{L}^{n,i,k}\})$.
The convergence of the iteration scheme (25) can further speed up when a *block-Jacobi approach* is utilized. In this approach the whole structure of the system (25) is retained, but the iterative matrix $A(\mathbf{S}^{n,i,k})$ is split into a block diagonal matrix $D(\mathbf{S}^{n,i,k})$, which is related to the volume integrals of (10), and into off-block diagonal $O(\mathbf{S}^{n,i,k})$, which arises by the computation of the numerical fluxes on the interfaces. The matrix $O(\mathbf{S}^{n,i,k})$ is transferred to the right hand side of (25a) and a *Jacobi interior iterative process* with D as iterative matrix is applied to obtain an intermediate solution $\mathbf{S}^{n,i,k+\frac{1}{2}}$. In the next Picard iterative step we use the solution $\mathbf{S}^{n,i,k+\frac{1}{2}}$ for updating the matrix A in (25a). The iterative process (25a) including the inner Jacobi iterations can be written as

$$A(\mathbf{S}^{n,i,k+\frac{1}{2}})\mathbf{S}^{n,i,k+1} = \mathbf{F}, \ k = 0, 1, 2, ..., \tag{26a}$$

where the solution $\mathbf{S}^{n,i,k+\frac{1}{2}}$ is obtained by

$$D(\mathbf{S}_j^{n,i,k})\mathbf{S}_{j+1}^{n,i,k} = \mathbf{F} - O(\mathbf{S}_j^{n,i,k})\mathbf{S}_j^{n,i,k}, \ j = 0, 1, 2, ..., J_M. \tag{26b}$$

$$\text{We set} \ \ \mathbf{S}^{n,i,k+\frac{1}{2}} := \mathbf{S}_{j=J_M}^{n,i,k}. \tag{26c}$$

The advantage of this approach is the following: the matrix D is block diagonal where every block, say D_E, is associated with an element $E \in \mathcal{T}_h$. The first and last row of D_E are formed by linear terms. The second row of D_E includes nonlinear entries which are formed by the integral terms of (10b). Having found the solution $\mathbf{S}_j^{n,i,k}$ (by the previous Jacobi iteration step), the nonlinear entries of D_E are updated easily by applying simple integration rules. Thus, the linear system (26b) with unknowns $\mathbf{S}_{j+1}^{n,i,k}$, can be solved block by block using a LU (stored) factorization procedure. Therefore, after the last Jacobi iteration, we can update the entries of the coefficient matrix (26a) by the solution $\mathbf{S}^{n,i,k+\frac{1}{2}}$ which is "closer" to $\mathbf{S}^{n,i,k+1}$. In fact $\mathbf{S}^{n,i,k+\frac{1}{2}}$ is "closer to " $\mathbf{S}^{n,i,k+1}$ than the solution that we obtain in the case where we apply explicit time stepping from t^n to t^{n+1}. More precisely, in this case the expression (26b) takes the form of an explicit treatment of (21) (or (25b)), that is

$$M_{\mathbf{L},\mathbf{A},\mathbf{U}}\mathbf{S}_{j+1}^{n,i,k} = \mathbf{F} - \tilde{O}(\mathbf{S}_j^{n,i,k}), \ j = 0, 1, 2, ..., J_M, \tag{27}$$

where $M_{\mathbf{L},\mathbf{A},\mathbf{U}}$ is the block-diagonal mass matrix of $\mathbf{L}, \mathbf{A}, \mathbf{U}$ and $\tilde{O}(\mathbf{S}_j^{n,i,k})$ includes all the rest flux terms appear in (21). The interior method (27) inherits the small size of Δt that we have to apply for solving the original problem (21) by an explicit method. Thus the intermediate solution $\mathbf{S}^{n,i,k+\frac{1}{2}}$ obtained by (27), does not differ much by the initial starting solution $\mathbf{S}^{n,i,k}$, (specially for small number J_M), and as a result we do not have a remarkable improvement in the

performance of the original Picard method (25). In order to have more clear picture of this, we make comparisons in the numerical examples, see also comments in the next paragraph *Implementation remarks*.

We point out that, the nested approach (26b) is expected to have similar convergence rate per Runge-Kutta cycle as the original scheme (25a) but improved performance in terms of CPU time. This is shown in the numerical examples. Henceforth, we will call the iterative scheme (26) as Jacobi-Picard and denote as *Jc-Picard*, the iterative scheme with the explicit nested method (27) Explicit Jacobi-Picard and denote *ExplJc-Picard*.

4.1 Impementation remarks

In practice, the iteration processes are stopped when $\|\Delta \mathbf{S}^{n,i,k+1}\|_{l^2} \leq \epsilon$ for a prescribed tolerance ϵ, and we set $\mathbf{S}^{n,i} := \mathbf{S}^{n,i,k+1}$ at the last iteration. We set $\epsilon = 1.E-06$.

In both Picard's iteration schemes we increased the efficiency of the iterative linear solver by applying a preconditioned GMRES iterative solver. The use of a preconditioned GMRES solver was not possible in the previous implementation of the matrix-free Newton method.

The codes materialized for all iterative processes have the ability of adjusting (increasing or decreasing) the size of Δt at the next time step, according to a criterion which is derived by using the number of the nonlinear iterations and the number of the iterations of the linear solver (GMRES) of the current time step. The solution of ODE problem (21) starts by using an initial $\Delta t_0 = (\frac{h_i}{16})^{k+1}$ and an initial guess given either by the initial conditions of (1) or by previous time steps solutions.

The code which materializes (27) keeps fixed time step $\Delta t_{Explicit} = \frac{\Delta t}{200}$. Both Jc-Picard and ExplJc-Picard iterative schemes utilize $J_M = 2$ inner Jacobi iterations.

5 Numerical tests

In this section we present several numerical results in order to illustrate the performance of the proposed LDG method to problem (1) with (p, δ)-structure. In the first paragraph, we consider the problem (1) with known exact solutions for verifying experimentally the spatial convergence rate of the method. Then, we study the efficiency (convergence characteristics) of the two different nonlinear iterative processes. In the last paragraph, we focus on the applicability of the method for solving realistic flow problems on domains with non-convex corners.

5.1 Convergence studies

All the numerical examples presented here have been performed using the Newton nonlinear iterative method. The domain is $\Omega := [-2, 2] \times [-2, 2]$. We set

$\Gamma_D = \partial\Omega$ and the data $\mathbf{f}, \mathbf{u}_D$ of (1) are specified by the given exact solution. Every test problem has been solved up to final time $T = 0.25$, the error is computed at the final time step and is given by the expression, [20],

$$\|.\|_{LDG} := \Big(\|\mathbf{F}(\mathbf{L}) - \mathbf{F}(\mathbf{L}_h)\|^2_{L^2(\Omega)} + \|\mathbf{F}^*(\mathbf{A}) - \mathbf{F}^*(\mathbf{A}_h)\|^2_{L^2(\Omega)} + \gamma h \|\mathbf{F}(h^{-1}[\![(\mathbf{u}_h - \mathbf{u}) \otimes \mathbf{n}]\!])\|^2_{L^2(\mathcal{E}_I \bigcup \mathcal{E}_D)} \Big)^{\frac{1}{2}}, \tag{28}$$

where $\mathbf{F}(\mathbf{A}) = (\delta + |\mathbf{A}|)^{\frac{p-2}{2}}\mathbf{A}$ and $\mathbf{F}^*(\mathbf{A}) = (\delta^{p-1} + |\mathbf{A}|)^{\frac{p'-2}{2}}\mathbf{A}$ with $\frac{1}{p} + \frac{1}{p'} = 1$. We use $k = 2$ and $k = 1$ for the local polynomial spaces. For every test-problem the values of the parameters are given in Table 1.

Smooth problem. In this test the exact solution is

$$\mathbf{u}(\mathbf{x}, t) = \begin{cases} u(\mathbf{x}, t) = B(t)\sin(x), \\ v(\mathbf{x}, t) = B(t)\cos(y), \end{cases} \tag{29}$$

where $B(t) = 1 + exp(-100t)$, and $\mathbf{x} = (x, y)$. The initial unstructured mesh $\mathcal{T}_{h_0}$ is generated by a triangular mesh generator with $h_0 = 1$ and the next finer meshes $\mathcal{T}_{h_i}$ are obtained by subdividing the triangles to four equal triangles, $h_{i+1} = \frac{h_i}{2}$. The numerical convergence rates r are computed by the ratio $r = \frac{\ln(e_{h_i}/e_{h_{i+1}})}{\ln(2)}$, where e_{h_i} is the error (28) computed on the mesh $\mathcal{T}_{h_i}$. In Fig. 1 we plot the exact solutions (left column) side by side with the corresponding LDG solutions (right column) computed using $k = 2$, $p = 2.5$ and the numerical flux (11b). The rates r are presented in Fig. 1(e) and in Fig. 1(f). We observe that the rates are the expected according to the regularity of $\mathbf{u}$ and are similar to the corresponding convergence rates that have been presented in [13], [14], [10] for $p = 2$-structure linear-nonlinear elliptic problems.

Natural regularity problem. In the following numerical experiments (test1, test2, test3) we asses the convergence rate of the proposed method in case where the solution of (1) has poor regularity. We consider a solution $\mathbf{u}$ of (1) with a point singularity at the origin given by

$$\mathbf{u}(\mathbf{x}, t) = \begin{cases} u(\mathbf{x}, t) = B(t)|\mathbf{x}|^{a(p)}y, \\ v(\mathbf{x}, t) = B(t)|\mathbf{x}|^{a(p)}(-x), \end{cases} \tag{30}$$

where the exponent $a(p)$ will be specified in the test cases. An easy computation shows that $|\nabla\mathbf{u}|$ behaves like $|\mathbf{x}|^{a(p)}$ and $\mathbf{F}(\nabla\mathbf{u})$ behaves like $|\mathbf{x}|^{\frac{a(p)p}{2}}$. The numerical flux (11b) has been used.

Numerical test 1. In this example, we set $a(p) = 0.01$ and the regularity of $\mathbf{F}(\nabla\mathbf{u})$ is close to $\mathbb{W}^{1,2}(\Omega)$. Due to the reduced regularity of the exact solution, we expect the convergence rates r for the $k = 2$ solution to be

reduced in comparison with the results of Fig. 1(e). Indeed, in Fig. 2(a) we plot the error versus grid size in the case $k = 2$ and we can see that all error rates r are less than two even in the linear case $p = 2$. In the case $k = 1$, Fig. 2(b), all error rates are optimal $r = 1$ with respect to the regularity of the exact solution.

Numerical test 2. In this example, we increased the regularity of the exact solution (30), by setting $a(p) = \frac{2}{p} + 0.01$, which means $\mathbf{F}(\nabla \mathbf{u}) \in \mathbb{W}^{2,2}(\Omega)$. The rates are presented in Fig. 2(c) and for $p \neq 2$ are suboptimal (less than 2) and only for the linear case $p = 2$, the rate r tends to 2, since we have $\nabla \mathbf{u} \in \mathbb{W}^{2,2}(\Omega)$.

Numerical test 3. In order to see the influence of the parameter δ (see (2)) to the accuracy of the LDG method, we performed the numerical test 2, setting $\delta = 0$ and $\delta = 1$. The corresponding convergence rates are plotted in Fig. 2(d) and in Fig. 2(e). Although the rates in Fig. 2(d) and Fig. 2(e) are slightly increased in comparison with the rates in Fig. 2(c), the overall behavior is the same. Thus, the numerical convergence rate of the LDG method, that we found in this example, are compatible with the regularity properties of the solution and are not strongly affected by the change of the perturbation parameter δ.

Numerical test 4. In Section 3, we proved that the proposed LDG method is stable for any $\gamma > 0$. In order to see numerically the influence of γ, we performed again the last test setting $\gamma = 1$, see Table 1. The rates are presented in Fig. 2(f) and appear to be the same with the rates presented in Fig. 2(e), as it was expected. A comparison of the CPU time for the two different choices of γ (but the same $\delta = 1$) is given in Table 2. In general, for the $p \geq 2$ tests cases, the CPU times are very closed, but for $p = 1.5$ the CPU time increases when the value of γ is increased. This explains our initial choice (see first lines Table 1) to perform the $p = 1.5$ tests using small value for the parameter γ.

Numerical flux comparisons. Here we investigate the effect of the penalty jump terms of the numerical fluxes (11b), (13) to the convergence rate of the error

$$\|\mathbf{F}(\mathbf{L}) - \mathbf{F}(\mathbf{L}_h)\|_{L^2(\Omega)}. \tag{31}$$

The parameter values are given in Table 1.

test (i) We solved the problem (1) with the exact solution (29) applying $k = 2$ polynomial space. Fig. 3(a) and Fig. 3(b) present the error (31) versus the grid size. As we can observe the error of both fluxes (11b) and (13) converges with (the expected) rate $r = 2$.

test (ii) We solved the problem (1) with exact solution (30) in the case of $a(p) = 0.01$ ($\mathbf{F}(\nabla \mathbf{u}) \in \mathbb{W}^{1,2}(\Omega)$) using $k = 1$ for the polynomial space. Fig. 3(c) and Fig. 3(d) show the convergence rates of the error (31). In both graphs, we can see that the rates r are similar and remain close to the optimal

Numerical example	Parameter values						Figure-Table
Smooth problem	p	1.5	2	2.5	δ=0.001		Fig. 1
	γ	0.2	2	2			
Natural regularity, Numerical test 1 Numerical test 2	p	1.5	2	2.5	δ=0.001		Fig. 2(a)(b)(c)
	γ	0.2	2	2			
Natural regularity, Numerical test 3	p	1.5	2	2.5	δ=0,δ=1		Fig. 2(d)(e)
	γ	0.2	2	2			
Natural regularity, Numerical test 4	p	1.5	2	2.5	δ=1		Fig. 2(f), Table 2
	γ	1	1	1			
Numerical fluxes, test(i),test(ii), test (iii)		Flux (11b)			Flux (13)		Fig. 3, Table 3
	-	$\delta = 0.001$			$\delta = 0.001$		
	p	1.5	2.5		1.5	2.5	
	γ	0.2	2		0.5	2	
Numerical fluxes, test(iv)		Flux (13), $\delta = 1$					Table 3
	p	1.5	2.5				
	γ	1	1				

Table 1. Numerical examples with the parameter values the graphs and Tables.

	p =1.5		p =2		p =2.5	
$\mathcal{T}_{h_i}$	$\gamma = 0.2$	$\gamma = 1$	$\gamma = 2$	$\gamma = 1$	$\gamma = 2$	$\gamma = 1$
-	CPU time, $\delta = 1$					
$i = 0$	1.76	0.92	0.81	0.71	1.93	1.96
$i = 1$	8.79	15.35	12.47	12.30	16.	15.25
$i = 2$	46.20	72.08	43.75	45.38	71.11	89.71
$i = 3$	334.24	528.9	360.12	347.88	559.08	702.89
$i = 4$	4531.8	5928.25	3693.2	3605.34	6192.34	6751.8

Table 2. CPUs for the two different choices of γ

rate $r = 1$, as the mesh is progressively subdivided. The error magnitude of flux (11b) is greater than the error magnitude of (13).

test (iii) For the problem (1) with exact solution (29), we present the CPU times for every mesh in the first four columns of Table 3. As we can see, for the $p = 2.5$ test cases the CPU time is almost the same for both numerical fluxes. For the test case with $p = 1.5$, the CPU time of the flux (13) is less than the CPU time of flux (11b), specially for the finer meshes. This shows that for the $p = 1.5$ test cases, the flux (13) produces a faster iterative method than the flux (11b).

test (iv) The last two columns of Table 3 show the CPU times of the solution of the problem (30) utilizing the flux (13) and using $\gamma = 1$ and $\delta = 1$. A comparison with the coresponding results in Table 2 shows that the CPU time of $p = 1.5$ test is lees for the flux (13), but the CPU times for the test

306 $p = 2.5$ are very closed. Thus, the CPU times of the numerical fluxes have similar behavior as in the test (iii).

	Problem (29)				Problem (30)	
	p =1.5		p =2.5		p =1.5	p =2.5
$\mathcal{T}_{h_i}$	Flux (11b)	Flux (13)	Flux (11b)	Flux (13)	Flux (13) $\delta = 1,\ \gamma = 1$	
-	CPU time					
$i = 0$	2.18	1.9	2.2	2.186	2.01	2.06
$i = 1$	14.78	14.63	16.48	16.82	9.0	11.11
$i = 2$	150.026	120.446	162.6	141.24	68.07	99.67
$i = 3$	1504	1098.4	604.2	576.588	548.58	790.7
$i = 4$	16040.4	11091.1	4995.44	4854.34	4885.01	6540.3

Table 3. CPUs for the two different numerical fluxes

307

308 5.2 Picard Performance

309 In this paragraph, we present results related to the performance of the Picard
310 non-linear iterative process and make comparisons with the results of the New-
311 ton iterative method. Both Picard and Newton methods are utilized for the LDG
312 method with numerical flux (13). We do not present convergence rate graphs,
313 since the rates have been found to be similar with the rates presented in the
314 previous paragraph.
315 We solved the problem (1) with exact solution (29), using $k = 1$ for the local
316 space and $\gamma = 2$ for all p test cases, see first line in Table 1. Table 4 shows
317 the CPU time of the Picard iterative methods for every $\mathcal{T}_{h_i}$. The CPU times
318 of ExplJc-Picard and Picard methods are quite close. It seems that, the imple-
319 mentation of the interior method (27) for updating the nonlinear entries of the
320 Picard matrix does not remarkably improve the performance of the whole itera-
321 tive method. Conversely, the CPU time values decrease quite when the Jc-Picard
322 method is applied. The decrease is higher for the fine meshes. In general, the
323 CPU time values of the same Picard method for the two different p test cases
324 are close. We can not see the same relation of the Newton CPU time values,
325 which are presented in Table 3 and Table 2. Further, we observe that the in-
326 creasing rate of the Picard CPU time for both p cases is almost 9.80 and the
327 corresponding rate of the Jc-Picard is around 9.55 (even for the fine meshes),
328 this shows a stable behavior for the method. On the other hand the results of
329 the Newton method in Table 3 show that the CPU time of $p = 1.5$ test case is
330 increasing with rate about 11.12, but the increasing rate for the $p = 2.5$ case is
331 about 9.4, hence the performance of the Newton iterative method varies with
332 the choice of p. A comparison of the convergence behavior of the methods with
333 respect to the total number of the iterations is presented in the first lines of
334 Table 5. The Newton methods appear to have better computational efficiency,

due to the lower number of total iterations needed for the full solution of the problem. The iterations of the Jc-Picard are reduced enough in comparison to the iterations of Picard method, (specially for the fine meshes), showing the effectiveness of this method over the last method. The last line of Table 5 shows the minimum-maximum number of the GMRES iterations during the solution of the problem on the last three meshes. The numbers confirm, the comments in Section 5 (paragraph *Newton iterative process*), that for $p = 1.5$ case, the resulting Jacobian matrix of the Newton method has higher condition number than the Jacobian matrix of $p = 2.5$ case. On the other hand, the results indicate that the condition numbers of the produced Picard matrices are less sensitive to the choice of the p, since the GMRES solver has similar behavior for the two p test cases.

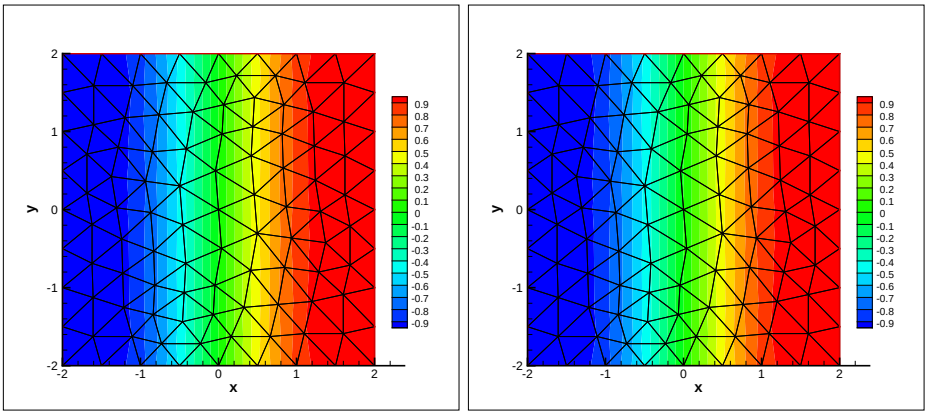
For the numerical examples that we present here, we point out that for the Newton method, we used a matrix-free implementation code (the Jacobian matrix is not explicitly stored), [18], [23], achieving better overall performance than the applied Picard methods, where the matrices are stored. This fact and the quadratic (expected) convergence rate of Newton approach, help this iterative method to be appeared more efficient than the two Picard methods, even though, as we observed in the last line of Table 5, the Newton method applies more GMRES iterations.

	Picard		ExplJc-Picard		Jc-Picard	
$\mathcal{T}_{h_i}$	p =1.5	p =2.5	p =1.5	p =2.5	p=1.5	p=2.5
-	CPU time					
$i = 0$	6.48	6.17	6.93	6.24	4.78	4.26
$i = 1$	58	55.36	58.05	55.2	38.36	36.4
$i = 2$	534.3	527.2	518.85	545.8	420.5	412.5
$i = 3$	5207.3	5087.6	5058.3	4756.07	4065.6	3966
$i = 4$	51361.4	49780	48628.9	44936	38462	37880

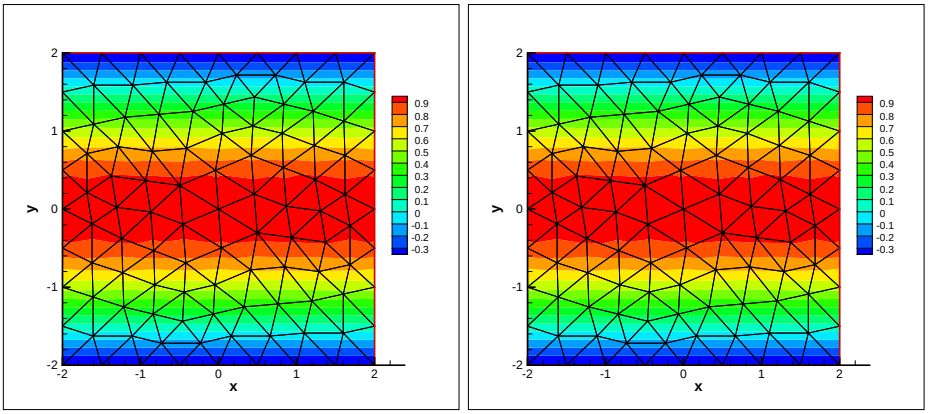
Table 4. CPUs for the Picard nonlinear iterative processes.

5.3 Realistic problems in non-smooth domains

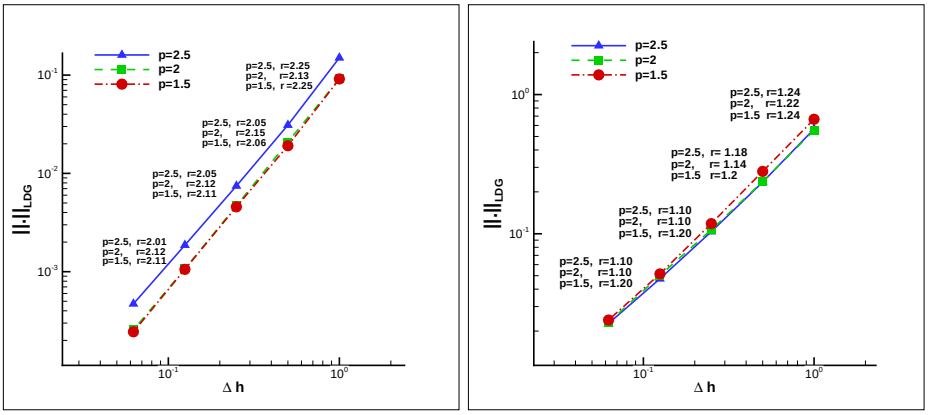
In this paragraph, we test the proposed LDG method on two more realistic flow problems where the boundary of the computational domain has non-convex corners. We know by the theory of linear elliptic problems,[31], that the solution has less regularity ($u \in W^{l=1+\varepsilon<2,2}$) around the corner points and the numerical method may lose its optimal accuracy. A numerical treatment in order to recover the accuracy of the method is the use of *locally graded mesh refinement technique*. This technique has been extensively studied for finite element methods for linear



(a) u (b) u_h

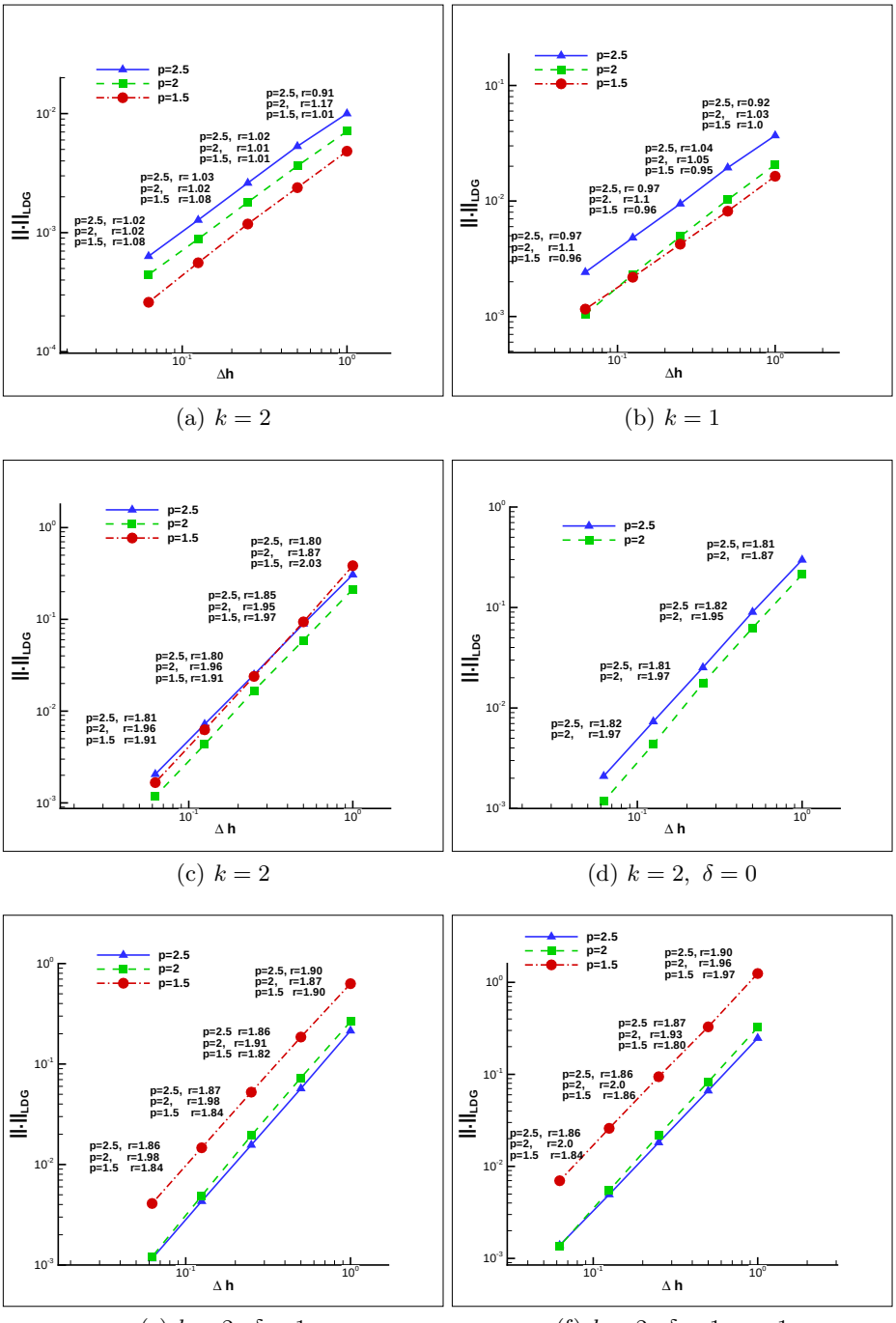


(c) v (d) v_h



(e) $k = 2$ (f) $k = 1$

Fig. 1. Convergence studies, smooth problem. First and second line: the contours of the exact solutions (29) side by side with the contours of the LDG solutions in case of $\mathbb{P}^2(E)$ and $p = 2.5$. Last line: (e) rates r for $\mathbb{P}^2(E)$, (b) rates r for $\mathbb{P}^1(E)$.

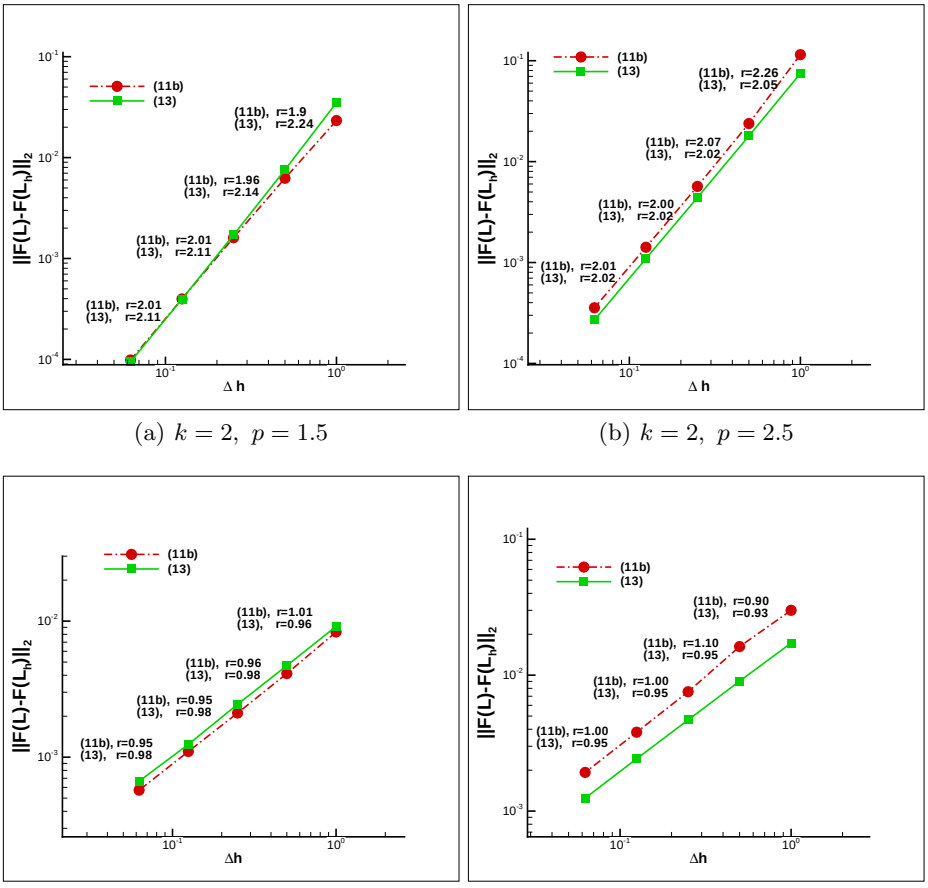


(a) $k = 2$ (b) $k = 1$

(c) $k = 2$ (d) $k = 2,\ \delta = 0$

(e) $k = 2,\ \delta = 1$ (f) $k = 2,\ \delta = 1, \gamma = 1$

Fig. 2. Convergence studies, natural regularity problem, (a) test 1: rates r for $\mathbb{P}^2(E)$ and $\mathbf{F}(\nabla \mathbf{u}) \in \mathbb{W}^{1,2}(\Omega)$, (b) test 1: rates r for $\mathbb{P}^1(E)$ and $\mathbf{F}(\nabla \mathbf{u}) \in \mathbb{W}^{1,2}(\Omega)$, (c) test 2: rates r using $\mathbb{P}^2(E)$ and $\mathbf{F}(\nabla \mathbf{u}) \in \mathbb{W}^{2,2}(\Omega)$, (d) test 3: rates r for $\mathbb{P}^2(E)$, $\delta = 0$ and $\mathbf{F}(\nabla \mathbf{u}) \in \mathbb{W}^{2,2}(\Omega)$. (e) test 3: rates r for $\mathbb{P}^2(E)$, $\delta = 1$ and $\mathbf{F}(\nabla \mathbf{u}) \in \mathbb{W}^{2,2}(\Omega)$. (f) test 4: rates r for $\mathbb{P}^2(E)$, $\delta = 1,\ \gamma = 1$ and $\mathbf{F}(\nabla \mathbf{u}) \in \mathbb{W}^{2,2}(\Omega)$.



(a) $k = 2,\ p = 1.5$ (b) $k = 2,\ p = 2.5$

(c) $k = 1,\ p = 1.5$ (d) $k = 1,\ p = 2.5$

Fig. 3. Numerical flux comparison. (a) test(i) exact solution (29): rates r for $\mathbb{P}^2(E)$, (b) test(i) exact solution (29): rates r for $\mathbb{P}^2(E)$, (c) test(ii) exact solution (30): rates r for $\mathbb{P}^1(E)$, and $\mathbf{F}(\nabla \mathbf{u}) \in \mathbb{W}^{1,2}(\Omega)$, (d) test(ii) exact solution (30): rates r for $\mathbb{P}^1(E)$, and $\mathbf{F}(\nabla \mathbf{u}) \in \mathbb{W}^{1,2}(\Omega)$.

	Picard		Jc-Picard		Newton	
$\mathcal{T}_{h_i}$	p =1.5	p =2.5	p=1.5	p=2.5	p=1.5	p=2.5
-	total number of iterations					
$i=0$	135	117	125	90	100	80
$i=1$	330	262	295	245	263	201
$i=2$	1406	1120	1183	1012	654	537
$i=3$	10055	7675	6545	5165	2202	1806
$i=4$	50412	46020	35770	26697	6640	5500
-	GMRES iterations					
Min-Max	5-17	3-15	5-17	3-15	14-35	11-24

Table 5. First lines: Total number of iterations for the nonlinear iterative processes. Last line: Min-Max iterations of the GMRES solver.

elliptic problems, see for example [24], [1], [28]. The refinement is determined by studying the singular behavior of the solution around the corners; and the finite element method, applied on the resulting graded refined mesh, has optimal convergence rate properties.
Next, for the solution of the first problem, we just use a quite fine uniform mesh and $k=2$ for the local polynomial space. For the numerical solution of the second problem, we construct a graded mesh.

Problem 1 The computational domain $\Omega = \Omega_1 - \Omega_2$ where $\Omega_1 = [-2,2] \times [-2,2]$, $\Omega_2 = [-1,1] \times [-1,1]$, with its triangulation is presented in Fig. 4(a). The corners of the interior square $\Omega_2 = [-1,1] \times [-1,1]$ constitute the severe singularities of the computational domain and make this problem challenging. Our particular interest here is, to study if there is any effect of the singular boundary points to the symmetric structure of the solution. This means that the flow field produced by the LDG method in the *upper* channel must be the same as this of the *lower* channel.

On the boundary of the interior square as on boundary parts $\{-2 \leq x \leq 2, y = \pm 2\}$, $\{x = -2, -2 \leq y \leq -1\}$, $\{x = -2, 1 \leq y \leq 2\}$ (denoted in Fig. 4(a) by Γ_0), Dirichlet boundary conditions $(u(x,y,t), v(x,y,t)) = (0,0)$ are imposed. On the part $\Gamma_{ud} = \{x = -2, -1 \leq y \leq 1\}$ we set $(u(x,y,t), v(x,y,t)) = (1,0)$, and on the *Neumann part* $\Gamma_N = \{x = 2, -2 \leq y \leq 2\}$ we set $\mathbf{a}_N = 0$, see Eq. (1d). The problem has been solved up to final time $T = 50$ using the numerical flux (11b). The values of the parameters are as in first line in Table 1. In Figs. 4(b)(c)(d) we plot the contour lines of the u_h LDG solution for $p = 2.5, p = 2, p = 1.5$ respectively. The contour lines show the symmetric flow field as we expected. Next we examine the profiles of u_h on the upper and lower channel. In Figs. 4(e)(f)(g) we plot the profiles of u_h computed on the points of the *upper* line, $L_{up} = \{x = 0, 1 \leq y_i^{up} \leq 2$, where $y_i^{up} = y_{i-1}^{up} + h,\ i = 1, ...\}$ and on the points of the *lower* line $L_{lw} = \{x = 0, -2 \leq y_i^{lw} \leq -1$, where $y_i^{lw} = y_{i-1}^{lw} + h,\ i = 1, ...\}$. Note that in this graph, for plotting reasons, we have set $y_i^{lw} := -y_i^{lw}$. The two profiles coincide for the three p test cases, as was expected for a symmetric flow

field. In Fig. 4(h), we plot the u_h profiles computed on points of the L_{up} line. All profiles are parabolic, depend on p, with the maximum value on the center point $M = (0, 1.5)$ of the axial direction. In comparison with the linear case $p = 2$, the profile becomes flatter with increasing the diffusivity ($p = 1.5$) and conversely becomes sharper reducing the diffusivity ($p = 2.5$), similar with the results that have been found for p-Navier Stokes systems in [35].

Problem 2 We consider the problem (1) in the computational domain Ω of Fig. 5(a) with $\mathbf{f} = 0$, where the boundary conditions are as follows: on $\Gamma_{D,1}$ we set periodic Dirichlet conditions $(u, v) = (\cos(2\pi t) + 2, 0)$, on $\Gamma_{D,0}$ Dirichlet conditions, $(u, v) = (0, 0)$ and on Γ_N Newman conditions $\mathbf{a}_N = 0$, (see (1d)).
The domain has three singular corners $C_s, s = 1, 2, 3$. The initial mesh $\mathcal{T}_h$ is graded in the following way. For every C_s, we consider a region $R_{C,s} = \{x \in \Omega : |x - C_s| \leq 0.5\}$. In every $R_{C,s}$ we construct N ring-type layers $L_i = \{x \in \Omega : r_{i-1} < |x - C_s| < r_i,\ i = 1, ..., N\}$ with $r_i = 0.5(\frac{i}{N})^{\frac{1}{\mu}}$, where the parameter $\mu \in (0, 1]$ is controlling the grading and we use $\frac{1}{N}$ to be of order h. The layers L_i are further partitioned to (approximately) equal side triangles of $h_{E \in L_i} \approx r_i - r_{i-1}$, see Fig. 5(a) for an illustration of the layers L_i and the resulting corner mesh for $N = 3, \mu = 0.4$. Such refinements (with the same grading parameters) were applied in [43] for the numerical solution of steady linear elliptic problems ($p = 2$).
The problem has been solved up to final time $T = 10$ using the numerical flux (11b) and $k = 1$ for the local polynomial space. For making comparisons, the problem has been solved numerically on a sequence of $\mathcal{T}_{h_i}, i = 0, ..., 3$ non graded meshes, too. Since there is no known exact solution, we check the numerical results using a *reference solution, U_r*, which has been obtained on a fine graded mesh and $k = 3$ for local polynomial space. The u_h fields computed on the graded $\mathcal{T}_{h_{i=1}}$ mesh at final time step are presented in Figs. 5(b)(c)(d). All the fields are symmetric with slow diffusion phenomena for $p = 1.5$ and more intense for $p = 2.5$. The diffusivity function for this test case is of the form $\tilde{\mathbf{A}}(\nabla \mathbf{u}(x, t)) = (\delta + |\nabla \mathbf{u}(x, t)|)^{p-2}$, one can expect that the $u_h(., t)$ will strongly vary with respect to t. For every time step t^n we compute the values of $u_h(P, t^n)$, where the mesh point $P = (0.368, 0)$ is located in the vicinity of the middle corner. In Figs. 5(e)(f)(g) the values of u_h and U_r for every p test case are plotted versus the time. It can be seen good agreement between the two solutions, both values u_h and U_r have the same periodic evolution. The amplitude of the obtained $p = 2.5$ solution is higher compared to the amplitude of $p = 1.5$ solution.
In Fig. 5(h) we display the convergence rates of the error for the graded ($\mu = 0.4$) meshes and in Fig. 5(i) the rates for uniform meshes ($\mu = 1$). The experimental results show that for the graded mesh the numerical solutions can approximate quite well the singular behavior of the solution, since the rates r of every p case approach the optimal convergence rate $r = 1$, (see the numerical examples in paragraph 6.1). On the other hand the convergence rates measured on non graded meshes are modulated by the poor regularity behavior of the solution.

Here, we have to mention that the convergence rates of $p = 2$ test presented in Fig. 5(i) are slightly higher than the rates which have been found in [43].

6 Conclusions

We have presented a LDG scheme for discretizing (p, δ)-structure systems. The proposed scheme utilizes two different nonlinear jump terms in the viscous numerical flux, which exhibit the (p, δ)-structure of the diffusion term. We have proven an a-priori bound for a simplified scalar problem. For the time integration Diagonal Implicit Runge-Kutta methods have been utilized. We have considered Newton and Picard iterative processes for solving the resulting nonlinear algebraic systems. The developing of further local-interior Jacobi type iterations, accelerated the overall performance of the Picard method. The main advantage of the last method is that it uses a better solution to update the nonlinear entries of the main iterative matrix. The performance of the iterative processes was compared on several test cases. The Picard methods are less affected by the change of the parameter values than the Newton method. We have discussed in detail through the numerical examples the convergence rates of the proposed LDG method and the rates were found to be optimal with respect to the regularity of the exact solution. More realistic problems have been considered in domains with non-smooth boundary in order to investigate the performance of the method. The problems were solved on graded meshes and the experimental convergence rates for all p cases, found to be similar with the rates of the finite element methods applied on similar linear problems.

7 Acknowledgments

The two first authors have been supported by the German Research Foundation under the project SFB TR 71 Geometric Partial Differential Equations, C2 Fluid structure interaction. The third author was partially supported by the German Research Foundation under the project SFB TR 71 Geometric Partial Differential Equations, C2 Fluid structure interaction and by the Austrian Academic Foundation under the project FWF-NFN S11703-N23.

References

1. T. Apel and B. Heinrich. Mesh Refinement and Windowing Near Edges for Some Elliptic Problem. *SIAM J. Numer. Anal.*, 31(3):695–708, 1994.
2. D. N. Arnold, F. Brezzi, B. Cockburn, and D. L. Marini. Unified analysis of discontinuous Galerkin methods for elliptic problem. *SIAM J. Numer. Anal.*, 39(5):1749–1779, 2002.
3. J. W. Barrett and W. B. Liu. Finite element approximation of the p-Laplacian. *Math. Comp.*, 61(204):523–537, 1993.
4. J. W. Barrett and W. B. Liu. Finite element approximation of the parabolic p-Laplacian. *SIAM J. Numer. Anal*, 31(2):413–428, 1994.

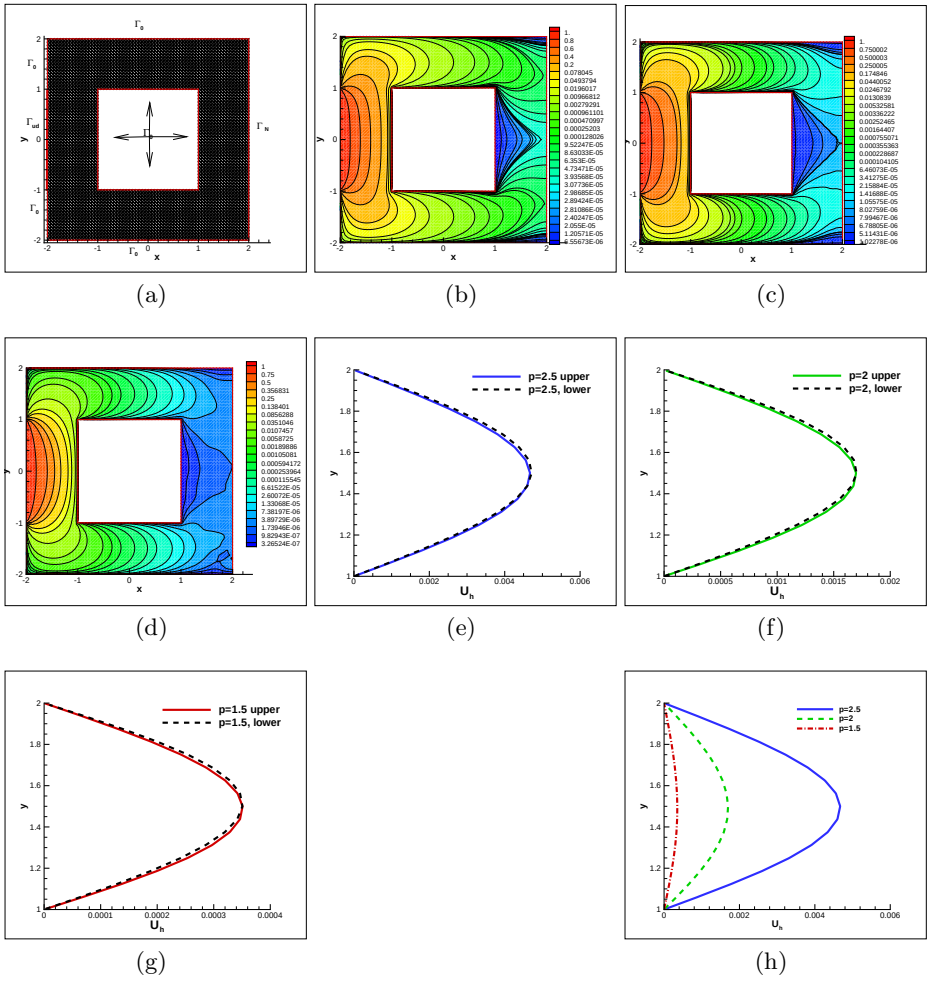


Fig. 4. Corner singularity problem. (a) the computational domain with the uniform triangulation, (b) u_h field for $p = 2.5$, (c) u_h field for $p = 2$, (d) u_h field for $p = 1.5$. (e) point value comparison of u_h on L_{up}, L_{lw} lines, for $p = 2.5$, (f) point value comparison of u_h on L_{up}, L_{lw} lines, for $p = 2$, (g) point value comparison of u_h on L_{up}, L_{lw} lines, for $p = 1.5$ (h) comparison of u_h profiles computed on L_{up} for the three values of p.

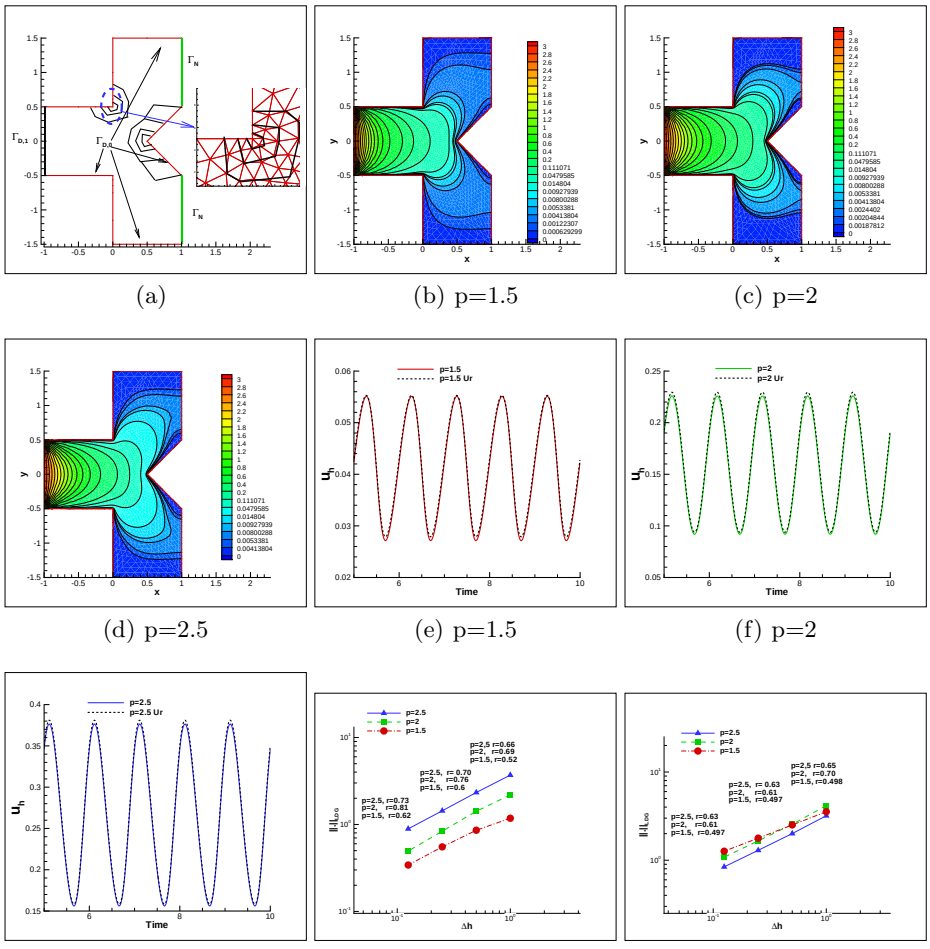


Fig. 5. Time dependent corner singularity problem: (a) The computational domain with the graded mesh regions near the corner singularities, (b) u_h flow field for $p = 1.5$, (c) u_h flow field for $p = 2$, (d) u_h flow field for $p = 2.5$, (e) comparisons between $u_h(P, t^n)$ and $U_r(P, t^n)$ point values for $p = 1.5$, (f) comparisons between $u_h(P, t^n)$ and $U_r(P, t^n)$ point values for $p = 2$, (g) comparisons between $u_h(P, t^n)$ and $U_r(P, t^n)$ point values for $p = 2.5$, (h) experimental convergence rates using graded meshes, (i) experimental convergence rates using uniform meshes.

5. J. W. Barrett and W. B. Liu. Finite element approximation of some degenerate monotone quasilinear elliptic systems. *SIAM J. Numer. Anal.*, 33(1):88–106, 1996.
6. F. Bassi and S. Rebay. A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier-Stokes equations. *J. Comput. Phys.*, 131(2):267–279, 1997.
7. E. Burman and A. Ern. Discontinuous Galerkin approximation with discrete variational principle for the nonlinear Laplacian. *C. R. Acad. Sci. Paris*, I 346:1013–1016, 2008.
8. R. Bustinza. A unified analysis of the local discontinuous Galerkin method for a class of nonlinear problems. *Appl. Numer. Math.*, 56(2):1293–1306, 2006.
9. R. Bustinza, G. Gatica, and B. Cockburn. An A Posteriori Error Estimate for the Local Discontinuous Galerkin Method Applied to Linear and Nonlinear Diffusion Problems. *J. Sci. Comput.*, 22 and 23:147–185, 2005.
10. R. Bustinza and G. N. Gatica. A Local Discontinuous Galerkin Method for Nonlinear diffusion problems with mixed boundary conditions. *SIAM, J. of Sci. Comput.*, 26(1):152–177, 2004.
11. Kreuzer C. Reliable and efficient a posteriori error estimates for finite element approximations for the parabolic p-laplacian. *Calcolo*, 110:50:79, 2013.
12. J. R. Cash. Diagonally implicit Runge-Kutta formulae with error estimates. *J. Inst. Maths Applies*, 24:293–301, 1979.
13. P. Castillo. Performance of discontinuous Galerkin methods for elliptic PDEs. *SIAM J. Sci. Comput.*, 24:524–547, 2002.
14. P. Castillo, B. Cockburn, I. Perugia, and D. Schötzau. An a priori error analysis of the local discontinuous Galerkin method for elliptic problems. *SIAM, J. Numer. Anal.*, 38:1676–1706, 2000.
15. P. G. Ciarlet. *The Finite Element Method for Elliptic Problems.* Studies in Mathematics and its Applications. North Holland Publishing Company, 1978.
16. B. Cockburn and C. W. Shu. The local discontinuous Galerkin method for time dependent convectiondiffusion systems. *SIAM J. Numer. Anal*, 6(35):24402463, 1998.
17. E. DiBenedetto. *Degenerate Parabolic Equations.* Springer Verlag, 1993.
18. D. Diehl. *High-Order Schemes for Simulation of Compressible Liquid-Vapor Flows with Phase Change.* PhD thesis, Mathematisches Institut, Abteilung fur Angewandte Mathematik, 2007.
19. L. Diening and F. Ettwein. Fractional Estimates for Non-differentiable Elliptic Systems with General Growth. *Forum Mathematicum*, 20(3):523–556, 2008.
20. L. Diening, D. Kröner, M. Růžička, and I. Toulopoulos. A local discontinuous Galerkin approximation for systems with p-structure. *IMA J. Numer. Anal.*, doi: 10.1093/imanum/drt040, 2013.
21. L. Diening and M. Růžička. Interpolation operators in Orliz-Sobolev spaces. *Numer. Math.*, 107:107–129, 2007.
22. L. Diening and M. Růžička. Non-Newtonian Fluids and Function Spaces. In *Nonlinear Analysis, Function Spaces and Applications, Proceedings of NAFSA 2006 Prague*, volume 8, pages 95–144, 2007.
23. DUNE. *, the Distributed and Unified Numerics Environment, http://www.dune-project.org/pdelab/.*
24. E. Stephan E. and J. R. Whiteman. Singularities of the Laplacian at corners and adges of three-dimensional domains and their treatment with finite element methods. *Math. Meth. Appl. Sci.*, 21(6):519–549, 1998.

25. Y. Epshteyn and B. Riviere. Estimation of penalty parameters for symmetric interior penalty Galerkin methods. *J. Comput. Appl. Math.*, 206:843–872, 2007.
26. I. Farago and J. Karatson. *Numerical Solution of Nonlinear Elliptic Problems Via Preconditioning Operators, Theory and Applications*, volume 11 of *Advances in Computation: Theory and Practice*. Nova Science Publisher, New York, 2002.
27. M. Farhloul. A mixed finite element method for a nonlinear Dirichlet problem. *IMA J. Numer. Anal.*, 18:121–132, 1998.
28. M. Feistauer and A.-M. Sändig. Graded mesh refinement and error estimates of higher order for DGFE solutions of elliptic boundary value problems in polygons. *Numer. Methods Partial Diff. Equations*, 28(4):1124–1151, 2012.
29. R. Glowinski and A. Marrocco. Sur l'approximation per elements finite d'ordre un, et la resolution, par penalization-dualite, d'une classe de Dirichlet non-lineaires. *RAIRO R-2*, pages 41–76, 1975.
30. R. Glowinski and J. Rappaz. Approximation of a nonlinear elliptic problem arising in a non-Newtonian fluid flow model in glaciology. *Math. Model. Numer. Anal.*, 37:175–186, 2003.
31. P. Grisvard. *Elliptic Problems in Nonsmooth Domains (Classics in Applied Mathematics)*. Number 69 in Classics in Applied Mathematics. SIAM, 2011.
32. Y. Q. Huang, L. Ruo, and L. Wenbin. Preconditioned descent algorithms for p-Laplacian. *J. Sci. Comput.*, 32(2):343–371, 2007.
33. Y. K. Kim. Error estimates for a mixed finite volume method for the p-Laplacian problem. *Mumer. Math*, 101:121–142, 2005.
34. R. M. Kirby and Em. G. Karniadakis. Selecting the numerical flux in discontinuous Galerkin methods for diffusion problems. *J. Sci. Comput.*, 22 and 23:385–411, 2005.
35. D. Kröner, M. Růžička, and I. Toulopoulos. Local discontinuous Galerkin numerical solutions of non-Newtonian incompressible flows modeled by p -Navier-Stokes equations. *J. Comp. Phys.*, 270:182–201, 2014.
36. J. L. Lions. *Quelques méthodes de résolution des problémes aux limites non linéarities*. Paris: Dunod, 1969.
37. J. Málek, K. R. Rajagopal, and M. Růžička. Existence and regularity of solutions and the stability of the rest state for fluids with shear dependent viscosity. *Math. Models Methods Appl. Sci.*, 5(5):789–812, 1995.
38. A. D. Oberman. A convergent difference scheme for the infinity Laplacian: construction of absolutely minimizing Lipschitz extensions. *Math. Comp.*, 74(5):1217–1230, 2005.
39. M. Picasso, J. Rappaz, A. Reist, M. Funk, and H. Blatter. Numerical simulation of the motion of a two dimensional glacier. *Int. J. Numer. Methods Eng.*, 60:995–1009, 2004.
40. A. Roger. Diagonally implicit Runge-Kutta methods for stiff O.D.E.'S. *SIAM J. Numer. Anal.*, 14(6):1006–1021, 1977.
41. Y. Saad. *Iterative Methods for Sparse Linear Systems, second edition*. Society of Industrial and Applied Mathematics, Philadelphia, 2003.
42. M. Santillana and C. Dawson. A local discontinuous Galerkin method for a doubly nonlinear diffusion equation arising in shallow water modeling. *Comput. Methods Appl. Mech. Engrg.*, 199:1424–1436, 2010.
43. A. Thomas, A.-M. Sändig, and J. R. Whiteman. Graded Mesh Refinement and Error Estimates for Finite Element Solutions of Elliptic Boundary Value Problems in Non-Smooth Domains. *Math. Meth. Appl. Sci.*, 19:63–85, 1996.