THE UNIVERSITY OF
WARWICK

University of Warwick institutional repository: http://go.warwick.ac.uk/wrap

This paper is made available online in accordance with
publisher policies. Please scroll down to view the document
itself. Please refer to the repository record for this item and our
policy information available from the repository home page for
further information.

To see the final version of this paper please visit the publisher's website.
Access to the published version may require a subscription.

Author(s): S. Brdar, A. Dedner, and R. Klöfkorn
Article Title: Compact and Stable Discontinuous Galerkin Methods for
Convection-Diffusion Problems
Year of publication: 2012
Link to published article:
http://dx.doi.org/10.1137/100817528

Publisher statement: © 2012 SIAM

SIAM J. SCI. COMPUT.
Vol. 34, No. 1, pp. A263–A282
© 2012 Society for Industrial and Applied Mathematics

COMPACT AND STABLE DISCONTINUOUS GALERKIN METHODS FOR CONVECTION-DIFFUSION PROBLEMS*

S. BRDAR†, A. DEDNER‡, AND R. KLÖFKORN†

Abstract. We present a new scheme, the compact discontinuous Galerkin 2 (CDG2) method, for solving nonlinear convection-diffusion problems together with a detailed comparison to other well-accepted DG methods. The new CDG2 method is similar to the CDG method that was recently introduced in the work of Perraire and Persson for elliptic problems. One main feature of the CDG2 method is the compactness of the stencil which includes only neighboring elements, even for higher order approximation. Theoretical results showing coercivity and stability of CDG2 and CDG for the Poisson and the heat equation are given, providing computable bounds on any free parameters in the scheme. In numerical tests for an elliptic problem, a scalar convection-diffusion equation, and for the compressible Navier–Stokes equations, we demonstrate that the CDG2 method slightly outperforms similar methods in terms of L^2-accuracy and CPU time.

Key words. discontinuous Galerkin, higher order discretization, stability, convection-diffusion, compressible Navier–Stokes

AMS subject classifications. 35G05, 35G50, 65M60, 65M99, 74S05

DOI. 10.1137/100817528

1. Introduction.

In this paper we introduce the compact discontinuous Galerkin 2 (CDG2) method for solving nonlinear convection-diffusion problems. The CDG2 method belongs to the group of the discontinuous Galerkin (DG) methods which were proposed and analyzed for the first time in the 1970s for solving partial differential equations. In particular, in 1973 for solving neutron transport equations of hyperbolic type by Reed and Hill [37]. In the work of Cockburn, Shu, and their collaborators the DG method has undergone a major development which has resulted in a series of papers, for example, [11, 12] for nonlinear hyperbolic conservation laws. The advantages of these methods over some other higher order finite element (FE) methods like Lagrange methods or higher order finite volume (FV) methods like ENO or WENO schemes include, for example, the following:

- Robust design of higher order function spaces due to the fact that higher order is achieved by choosing polynomial degree locally on each grid element.
- Easy implementation on nonconforming unstructured meshes with hanging nodes. These meshes are well suited for local grid adaptation, a desirable feature for resolving multiscale character of different problems.
- Locality of the method as a result of discontinuous numerical solutions, where the discontinuity occurs only in the intersection of grid elements. For the first order partial differential equations a DG spatial operator of arbitrary higher

*Submitted to the journal's Methods and Algorithms for Scientific Computing section December 9, 2010; accepted for publication (in revised form) October 20, 2011; published electronically February 2, 2012.

http://www.siam.org/journals/sisc/34-1/81752.html

†Section of Applied Mathematics, University of Freiburg, Hermann-Herder-Strasse 10, D-79104 Freiburg, Germany (slavko@mathematik.uni-freiburg.de, robertk@mathematik.uni-freiburg.de). The first author's work was supported by the German Research Foundation (the DFG) under the project "DFG Schwerpunktprogramm (SPP) 1276." The work of the third author has been supported by the *Baden-Württemberg Stiftung* under the project HPC-11.

‡Mathematics Institute, University of Warwick, Coventry CV4 7AL, United Kingdom (A.S. Dedner@warwick.ac.uk).

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

order requires only information from direct neighbors. This is a key feature for efficient computation on today's *multicore* parallel architectures.

Various versions of the DG method to solve elliptic problems have emerged over the years, and it is interesting to mention the work [2] which unifies several of them in an abstract framework and provides analysis of their accuracy and stability for Poisson's equation.

Two of the methods mentioned in [2] which satisfy the properties stated above are the interior penalty (IP) (introduced in [18]) or Bassi-Rebay 2 (BR2) (introduced in [4, 5]). More recently the compact discontinuous Galerkin (CDG) method was introduced in [36].

The IP scheme with different stabilization terms is analyzed for the two-dimensional (2D) compressible nonlinear Navier–Stokes equations in the work of Hartmann and Houston [27]. Additional stabilization of the IP method is based on the penalization of jumps of the numerical solution across grid interfaces which has to take into account the order of the method. Estimates of the penalization parameters for one-dimensional parabolic problems are presented in [34], for 2D elliptic problems in [1, 19, 20], and for 2D compressible nonlinear Navier–Stokes equations in [26]. In [22, 38] the problem of estimating the penalty coefficient, in case of simplified diffusion term, is transformed into a problem of finding an estimate for a series of inequalities between different norms. The BR2 method is compared with IP in [27] (see also the references therein). The stabilization of the BR2 method, as well as for the CDG and the CDG2 methods, is based on special *lifting operators*. This approach may come at a considerable computational cost, since the lifting operators need to be computed on both grid elements which share an interface. The advantage of CDG and CDG2 over BR2 is exactly at this point, as they require the evaluation of one lifting operator on only one side of each interface. In the context of nonlinear problems this is even more important since one might consider a matrix free implementation of these methods. Calculations of these liftings add a nonnegligible part to the computational cost of the scheme especially on general quadrilateral and hexahedral grids.

The rest of this paper is organized as follows. In section 2, we describe the CDG2, CDG, BR2, and IP methods in a suitable form for the stability analysis carried out in section 3. Here, the analysis of the coercivity in the case of Poisson's equations and L^2-stability in the case of a linear heat equation is carried out for CDG and CDG2. In section 4, we highlight implementation details. Most notably we use the stability estimate for the CDG2 method to derive a special *switching function* which improved the performance of the method considerably. Practical results, including comparisons of the new CDG2 method with CDG, IP, and BR2, are presented in section 5. Conclusions are drawn in section 6.

2. DG formulation for convection-diffusion equations. In this section we will derive the primal DG formulation for general nonlinear convection-diffusion-reaction equations of the form

$$\begin{aligned} \partial_t u + \nabla \cdot \big(f(u) - A(u)\nabla u\big) &= s(u) && \text{in } \Omega \times (0, t_{\text{end}}), \\ u &= g_D && \text{on } \partial\Omega \times [0, t_{\text{end}}), \\ u(0,\cdot) &= u_0 && \text{in } \Omega, \end{aligned} \tag{2.1}$$

where $u : \Omega \times [0, t_{\text{end}}] \to \mathbb{R}$, $A : \mathbb{R} \to \mathbb{R}^{d\times d}$, $f : \mathbb{R} \to \mathbb{R}^d$, $s : \mathbb{R} \to \mathbb{R}$, and $\Omega \subset \mathbb{R}^d$ is a bounded subset with polygonal (for $d = 2$, or polyhedral for $d = 3$) boundary.

In this paper we focus, in particular, on the discretization of the diffusion term in (2.1). To this end we first consider the discretization of a linear elliptic problem

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

with variable coefficients which will serve as a building block for the discretization of (2.1) discussed in section 2.2. The discretization is described for a 2D problem, but it is straightforward to extend it to three space dimensions.

2.1. Elliptic problems. In order to derive the discretization of the diffusion term in (2.1) we consider the following elliptic problem in $\mathbb{R}^d$, $d = 2$, of the form

$$\begin{aligned} -\nabla \cdot (A(x)\nabla u(x)) &= s(x) \qquad x \in \Omega, \\ u &= g_D \qquad \text{on } \partial\Omega, \end{aligned} \tag{2.2}$$

where $\Omega \subset \mathbb{R}^d$ is a bounded polygonal area, $A \in L^\infty(\Omega, \mathbb{R}^{d\times d})$ a positive definite diffusion matrix, and $s \in L^2(\Omega)$.

We are interested in deriving a discrete *primal formulation* for (2.2) of the form

$$B(u_h, \varphi) = \int_\Omega s\varphi \qquad \forall \varphi \in V_h \ . \tag{2.3}$$

The discrete solution u_h is in the piecewise polynomial space $V_h = V_h^1$ with

$$V_h^l = \{\boldsymbol{v} \in L^2(\Omega, \mathbb{R}^l) : \boldsymbol{v}|_K \in [\mathcal{P}_k(K)]^l\} \quad \text{for some } l \in \mathbb{N}$$

defined for a given partition $\mathcal{T}_h = \{K\}$ of Ω into polygons K. The space V^l is contained in the function space

$$V^l = \{\boldsymbol{v} \in L^2(\Omega, \mathbb{R}^l) : \boldsymbol{v}|_K \in [H^2(K)]^l\} \ .$$

In addition to $V_h = V_h^1$ we also use the abbreviation $\Sigma_h = V_h^d$, $V = V^1$, and $\Sigma = V^d$ in the following.

To derive the bilinear form B we need to introduce some standard notation (see [2]). By Γ_i we denote the family of all interior intersections e of grid elements $K_e^+, K_e^- \in \mathcal{T}_h$, where $e = K_e^- \cap K_e^+$ and positive Hausdorff measure in $\mathbb{R}^{d-1}$. We restrict ourselves to conforming grids, so that an intersection e can be only a whole edge of an element $K_e^\pm$. Additionally, let Γ be the family of all intersections $e \subset \partial K$, where $K \in \mathcal{T}_h$. For each intersection e we define the local mesh width $h_e = \frac{|e|}{\max\{|K_e^-|, |K_e^+|\}}$. For $e \in \Gamma_i$, $\varphi \in V$, and $\boldsymbol{\tau} \in \Sigma$ we introduce operators $[\cdot]_e$, $\{\cdot\}_e$, $\{\!\{\cdot\}\!\}_e$, and $[\![\cdot]\!]_e$ as

$$\begin{aligned} [\![\varphi]\!]_e &= \varphi_{|K_e^-}\boldsymbol{n}_{K_e^-} + \varphi_{|K_e^+}\boldsymbol{n}_{K_e^+}, \qquad & \{\varphi\}_e &= \frac{1}{2}(\varphi_{|K_e^-} + \varphi_{|K_e^+}), \\ [\boldsymbol{\tau}]_e &= \boldsymbol{\tau}_{|K_e^-} \cdot \boldsymbol{n}_{K_e^-} + \boldsymbol{\tau}_{|K_e^+} \cdot \boldsymbol{n}_{K_e^+}, \qquad & \{\!\{\boldsymbol{\tau}\}\!\}_e &= \frac{1}{2}(\boldsymbol{\tau}_{|K_e^-} + \boldsymbol{\tau}_{|K_e^+}), \end{aligned}$$

and for a boundary intersection $e \subset \partial\Omega$ as

$$\begin{aligned} [\![\varphi]\!]_e &= (\varphi - g_D')\boldsymbol{n}, \qquad & \{\varphi\}_e &= \varphi, \\ [\boldsymbol{\tau}]_e &= \boldsymbol{\tau} \cdot \boldsymbol{n}, \qquad & \{\!\{\boldsymbol{\tau}\}\!\}_e &= \boldsymbol{\tau}, \end{aligned}$$

where $g_D' = g_D$ in case $[\![\cdot]\!]_e$ acts on u_h; otherwise $g_D' = 0$. Note that instead of the arithmetic averages $\{\cdot\}_e, \{\!\{\cdot\}\!\}_e$ one could also use, for example, weighted harmonic averages as suggested in [10, 17, 20] and other papers. This, however, is outside of the scope of this paper.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Following the derivation of the DG primal formulation found in [2] we obtain, for given numerical fluxes $\widehat{u}$ and $\widehat{A}$, both mapping u_h to $[L^2(\Gamma)]^d$, the flux based bilinear form

$$
\begin{aligned}
B(u_h,\varphi) := &\int_\Omega (A\nabla u_h)\cdot\nabla\varphi - \sum_{e\in\Gamma_i}\int_e [A^T\nabla\varphi]_e\{u_h-\widehat{u}\}_e \\
&- \sum_{e\in\Gamma}\left(\int_e \{\!\{A^T\nabla\varphi\}\!\}_e\cdot[\![u_h]\!]_e + \widehat{A}\cdot[\![\varphi]\!]_e\right).
\end{aligned}
\tag{2.4}
$$

The method is completely described once the physical parameter functions A, s, and g are known and appropriate numerical fluxes have been chosen. To define the numerical diffusion fluxes, let us define two kinds of *lifting operators* $\boldsymbol{r}_e : [L^2(e)]^d \to \Sigma_h$ and $\boldsymbol{l}_e : L^2(e) \to \Sigma_h$, for every $e \in \Gamma$, with

$$
\int_\Omega \boldsymbol{r}_e(\boldsymbol{\xi})\cdot\boldsymbol{\tau} = -\int_e \boldsymbol{\xi}\cdot\{\!\{\boldsymbol{\tau}\}\!\}_e, \quad \int_\Omega \boldsymbol{l}_e(\phi)\cdot\boldsymbol{\tau} = -\int_e \phi[\boldsymbol{\tau}]_e
\tag{2.5}
$$

for all $\boldsymbol{\tau}\in\Sigma_h$, $\boldsymbol{\xi}\in[L^2(e)]^d$, and $\phi\in L^2(e)$. For our convenience we define $\boldsymbol{L}_e(u) := \boldsymbol{r}_e([\![u]\!]_e)+\boldsymbol{l}_e(\boldsymbol{\beta}_e\cdot[\![u]\!]_e)$ on $e\in\Gamma$. The parameter $\boldsymbol{\beta}$ (in the literature frequently $\mathbf{C}_{12}$) is called the *switch function*. We assume in the following that for an interior intersection e with neighboring element K_e^+, K_e^- we have

$$
\boldsymbol{\beta}_e = \frac{1}{2}\boldsymbol{n}_{K_e^-} = -\frac{1}{2}\boldsymbol{n}_{K_e^+}\ ,
\tag{2.6}
$$

while on the boundary we set $\boldsymbol{\beta}_e = \boldsymbol{n}_{K_e}/2$ for CDG2 and $\boldsymbol{\beta}_e = 0$ for the other methods. Here, $\boldsymbol{n}_{K_e}$ is the outer unit normal of K_e on intersection e. Different choices for $\boldsymbol{\beta}_e$ have been suggested (e.g., in [2, 13, 36]). We will discuss two definitions for $\boldsymbol{\beta}_e$ in section 4, one of which is motivated by our coercivity estimate. Note that the meaning of K_e^+, K_e^- is fixed once a switching function has been chosen.

For a given switch function $\boldsymbol{\beta}_e$ we call e an *outflow* intersection of a grid element K if $e\subset\partial K$ and $\boldsymbol{n}_K\cdot\boldsymbol{\beta}_e > 0$. Notice that this definition tells us that $K = K_e^-$. The number of all outflow intersections of K is denoted by N_K^{out}, whereas the maximal number of outflow intersections for one grid element is denoted by $N_{\mathcal{T}_h}^{out}$, i.e.,

$$
N_K^{out} = \#\{e\in\Gamma : \boldsymbol{n}_e\cdot\boldsymbol{\beta}_e > 0 \text{ and } e\subset\partial K\}\ , \quad N_{\mathcal{T}_h}^{out} = \max_{K\in\mathcal{T}_h} N_K^{out}.
\tag{2.7}
$$

In addition we denote by N_K the total number of interfaces of K and we define

$$
N_{\mathcal{T}_h} := \max_{K\in\mathcal{T}_h} N_K.
\tag{2.8}
$$

In Table 1 we show the numerical diffusion fluxes $\hat{u}$ and $\widehat{A}$ considered in this paper. The numerical fluxes on the boundary are also prescribed in the table based on the definition of the operators $[\cdot]_e$, $[\![\cdot]\!]_e$, $\{\cdot\}_e$, and $\{\!\{\cdot\}\!\}_e$ on the boundary. The only exception to this is that $\hat{u} = g_D$ on a boundary intersection.

Remark 1. The term A in the penalty term $\boldsymbol{\delta}_e(u)$ plays an important role in case of a strongly nonlinear system of equations, like in the case of the Navier–Stokes system. The absence of this term leads to insufficient numerical diffusion resulting in suboptimal convergence rates. The stability of the IP method is especially prone to changes of the stability coefficient η (see [27]) whereas the CDG and the CDG2

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

TABLE 1

Numerical diffusion fluxes on an interface e for all methods considered in this paper. For simplicity we write u instead of u_h and use the abbreviations $\boldsymbol{d}(u) = A\nabla_h u$, $\boldsymbol{j}_e(u) = A\boldsymbol{L}_e(u)$, and $\boldsymbol{\delta}_e(u) = \frac{\eta}{h_e}\{\!\{A\}\!\}_e[\![u]\!]_e$ with $\eta \geq 0$ and $\chi \geq 0$.

Method	$\hat{u}$	$\widehat{A}$
IP	$\{u\}_e$	$\{\!\{\boldsymbol{d}(u)\}\!\}_e - \boldsymbol{\delta}_e(u)$
BR2	$\{u\}_e$	$\{\!\{\boldsymbol{d}(u)\}\!\}_e$
		$+\chi\{\!\{A\boldsymbol{r}_e([\![u]\!]_e)\}\!\}_e$
CDG	$\{u\}_e - \boldsymbol{\beta}_e \cdot [\![u]\!]_e$	$\{\!\{\boldsymbol{d}(u)\}\!\}_e - \boldsymbol{\delta}_e(u) + \boldsymbol{\beta}_e[\boldsymbol{d}(u)]_e$
		$+\chi\big(\{\!\{\boldsymbol{j}_e(u)\}\!\}_e + \boldsymbol{\beta}_e[\boldsymbol{j}_e(u)]_e\big)$
CDG2	$\{u\}_e$	$\{\!\{\boldsymbol{d}(u)\}\!\}_e - \boldsymbol{\delta}_e(u)$
		$+\chi\big(\{\!\{\boldsymbol{j}_e(u)\}\!\}_e + \boldsymbol{\beta}_e[\boldsymbol{j}_e(u)]_e\big)$

methods are shown to be completely immune to changes of η (see section 3). Note that η is denoted with C_{11} in other papers, e.g., in [13, 36]. Also note the slight difference in the definition of the flux for the CDG method compared to [36], where the method was introduced with $\chi = 1$. Similarly, BR2 was introduced in [4, 5] with $\chi = 3$ for triangular grids, but we leave the constant χ free, so that we can take different grid types (triangular, quadrilateral, tetrahedral etc.) into account.

2.2. Convection-diffusion-reaction equations. In this section we briefly discuss a discrete formulation for equations of the form (2.1) based on the diffusion discretization presented in the previous section:

$$\int_\Omega \varphi \partial_t u_h = B_{CDR}(u_h, \varphi) \quad \forall \varphi \in V_h \ . \tag{2.9}$$

The bilinear form is $B_{CDR}(u_h, \varphi) := -B(u, \varphi) + B_{CR}(u, \varphi)$ with B given by (2.4), where this time u_h in (2.4) is time-dependent and A depends on u_h, and

$$B_{CR}(u, \varphi) = \int_\Omega f(u) \cdot \nabla\varphi - \sum_{e\in\Gamma} \int_e \widehat{f}(u) \cdot [\![\varphi]\!]_e + \int_\Omega \varphi s(u) \quad \forall \varphi \in V_h. \tag{2.10}$$

The first two summands in (2.10) correspond to the discretization of the convection term and the last summand to the discretization of the reaction term. In the DG context the discretization of the convective terms is well studied, e.g., in [11, 12] and many other papers; possibly a stabilization is required in convection dominated cases as discussed in [15] and references therein. Other stabilization techniques such as entropy viscosity (cf. [25]) or artificial viscosity methods (e.g., [23]) require a diffusion discretization as presented in the previous section. The convective numerical flux $\widehat{f}(u)$ can be any appropriate numerical flux known for standard FV methods, e.g., presented in textbooks [32, 39] on the subject.

3. Theoretical results. In this section we derive coercivity results for the different methods for elliptic problems presented in the previous section. The coercivity results are expressed with respect to the grid-dependent norm:

$$|\!|\!| v |\!|\!|^2 = \sum_{K\in\mathcal{T}_h} |v|^2_{1,K} + \sum_{e\in\Gamma} \|\boldsymbol{r}_e([\![v]\!]_e)\|^2_\Omega$$

for $v \in V_h + H^1_0(\Omega) \cap H^2(\Omega)$. This is the same norm used in [8].

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

For the method presented in the previous section the following theorem can be proven. The results for the IP and BR2 methods can be found in the literature, e.g., [2]. The new results are the proofs for the CDG and CDG2 methods.

THEOREM 2 (coercivity estimate). *Let $\mathcal{T}_h$ be a conforming grid on Ω such that there is an affine mapping from a fixed reference element $\hat{K}$ to each $K \in \mathcal{T}_h$[1] and let $\boldsymbol{\beta}_e$ be as in* (2.6)*. Consider the problem* (2.2) *with $A = const$ and $g = 0$. For the bilinear form B given in* (2.4) *the inequality*

$$B(u,u) \geq C|||u|||^2 \quad \forall u \in V_h$$

holds for some $C > 0$ if one of the following conditions is fulfilled:

(a) *η is chosen sufficiently large and $\chi \geq 0$ (for IP, CDG, and CDG2),*
(b) *$\eta \geq 0$ and $\chi > \chi_0$, where*
   1. *$\chi_0 = N_{\mathcal{T}_h}$ for the BR2 method,*
   2. *$\chi_0 = N^{out}_{\mathcal{T}_h}$ for the CDG method, and*
   3. *for the CDG2 method*

$$\chi_0 = \frac{N_{\mathcal{T}_h}}{4}\big(1 + \nu(\boldsymbol{\beta})\big),$$

with $\nu(\boldsymbol{\beta}) = \max_{e\in\Gamma_i}\{|K_e^-|/|K_e^+|\}$ and K_e^-, K_e^+ determined by $\boldsymbol{\beta}$.

The mesh-dependent constants $N^{out}_{\mathcal{T}_h}$ and $N_{\mathcal{T}_h}$ used to define χ_0 are given in (2.7) *and* (2.8)*, respectively.*

The first part of this theorem is a simple extension of the proof found in [2] and was also mentioned for the original CDG method, i.e., with $\chi = 1$ in [36]. The generalized estimate for the coercivity coefficient of the BR2 method is an easy consequence of the discussion in [2, 8]. Therefore, we focus on the results for the CDG and the CDG2 methods. The proof is given in section 3.2.

Before we proceed with the proof of Theorem 2 in section 3.2, we need to summarize some properties of the lifting operators that we require for the proof. In the following we will use the abbreviations $\boldsymbol{L}(u) = \sum_{e\in\Gamma} \boldsymbol{L}_e(u)$ and $\boldsymbol{r}(u) = \sum_{e\in\Gamma} \boldsymbol{r}_e([\![u]\!]_e)$.

3.1. Properties of the lifting operators. In the following we assume that the conditions stated in Theorem 2 are satisfied. The following lemmas summarize some simple observations.

LEMMA 3. *Let $\boldsymbol{\beta}_e$ be chosen as in* (2.6) *and $e \in \Gamma_i$. Then for $u \in V_h$ it holds that*

$$\operatorname{supp} \boldsymbol{r}_e([\![u]\!]_e) = K_e^+ \cup K_e^- \qquad \text{and} \qquad \operatorname{supp} \boldsymbol{L}_e(u) = K_e^- \,. \tag{3.1}$$

Furthermore, $\boldsymbol{L}_e(u) = 2\boldsymbol{r}_e([\![u]\!]_e)$ on K_e^-.

For $e \in \Gamma \setminus \Gamma_i$ we have $\boldsymbol{L}_e(u) = \boldsymbol{r}_e([\![u]\!]_e)$ for the CDG method and $\boldsymbol{L}_e(u) = 2\boldsymbol{r}_e([\![u]\!]_e)$ for the CDG2 method.

[1] This is, for example, the case for simplex or Cartesian grids.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Proof. The result for $\boldsymbol{r}_e$ follows from the definition. For the second part we denote with $\boldsymbol{\tau}_e^+$ the restriction of a function $\boldsymbol{\tau} \in \Sigma$ on K_e^+. Similarly $\boldsymbol{\tau}_e^- = \boldsymbol{\tau}|_{K_e^-}$. Then

$$\begin{aligned}\int_\Omega \boldsymbol{L}_e(\phi)|_{K_e^+} \cdot \boldsymbol{\tau} &= \int_\Omega \boldsymbol{L}_e(\phi) \cdot \boldsymbol{\tau}_e|_{K_e^+} = -\int_e [\![\phi]\!]_e \cdot \{\!\{\boldsymbol{\tau}_e^+\}\!\}_e - \int_e \boldsymbol{\beta}_e \cdot [\![\phi]\!]_e [\boldsymbol{\tau}_e^+]_e \\ &= -\int_e [\![\phi]\!]_e \cdot \left(\frac{1}{2}\boldsymbol{\tau}_e^+ + \boldsymbol{\beta}_e \cdot \boldsymbol{n}_{K_e^+}\boldsymbol{\tau}_e^+\right) \\ &= -\int_e [\![\phi]\!]_e \cdot \left(\frac{1}{2}\boldsymbol{\tau}_e^+ - \frac{1}{2}\boldsymbol{\tau}_e^+\right) = 0\,.\end{aligned} \tag{3.2}$$

Similarly one can show that $\int_\Omega \boldsymbol{L}_e^-(\phi)\cdot\boldsymbol{\tau} = 2\int_\Omega \boldsymbol{r}_e^-([\![\phi]\!]_e)\cdot\boldsymbol{\tau}$ for any $\boldsymbol{\tau} \in \Sigma$ from which the other equality in the lemma follows directly.

For e on the boundary we have for CDG2 that $\boldsymbol{\beta}_e$ is defined as in the interior, so that the result stated above also holds in this case. For the CDG method $\boldsymbol{\beta}_e = 0$ and this gives us $\boldsymbol{L}_e(u) = \boldsymbol{r}_e([\![u]\!]_e)$. ∎

The key ingredient in the proof of Theorem 2 is the following lemma, which relates $\boldsymbol{r}_e(\cdot)|_{K_e^-}$ and $\boldsymbol{r}_e(\cdot)|_{K_e^+}$.

LEMMA 4. *Let $\mathcal{T}_h$ be as in the Theorem 2, $\boldsymbol{\beta}_e$ as in (2.6), and $e \in \Gamma_i$. Then there exists a positive constant α_e such that*

$$\alpha_e \|\boldsymbol{r}_e([\![u]\!]_e)\|^2_{K_e^-} = \|\boldsymbol{r}_e([\![u]\!]_e)\|^2_{K_e^+} \quad \forall u \in V_h. \tag{3.3}$$

Moreover, we have $\alpha_e = |K_e^-|/|K_e^+|$.

If we define for all $e \in \Gamma \setminus \Gamma_i$ $\alpha_e = 3$ (CDG) and $\alpha_e = 0$ (CDG2), we have

$$\|\boldsymbol{L}_e(u)\|^2_\Omega = \frac{4}{1+\alpha_e}\|\boldsymbol{r}_e([\![u]\!]_e)\|^2_\Omega \tag{3.4}$$

for all $e \in \Gamma$.

Proof. In the following we use the abbreviation $\boldsymbol{r}_e = \boldsymbol{r}_e([\![u]\!]_e)$. Consider the affine mappings F_{e-}, F_{e+} of the reference element $\hat{K}$ to K_e^-, K_e^+, respectively, such that $F_{e+}^{-1}(e) = F_{e-}^{-1}(e)$ and with an orientation such that for all $i \in \overline{1,N}$ ($N = \dim \Sigma_h(K_e^-) = \dim \Sigma_h(K_e^+)$)

$$\hat{\boldsymbol{\tau}}_i \circ F_{e+}^{-1}(x) = \hat{\boldsymbol{\tau}}_i \circ F_{e-}^{-1}(x) \quad \forall x \in e,$$

for some orthonormal basis $\{\hat{\boldsymbol{\tau}}_i\}_{i=\overline{1,N}}$ of $\Sigma_h(\hat{K})$. Consequently, $\{\boldsymbol{\tau}_i^\pm = \hat{\boldsymbol{\tau}}_i \circ F_{e\pm}^{-1}\}_{i=\overline{1,N}}$ is an orthogonal basis of $\Sigma_h(K_e^\pm)$ such that $\boldsymbol{\tau}_i^- = \boldsymbol{\tau}_i^+ =: \boldsymbol{\tau}_{e,i}$ on e. By setting $\boldsymbol{\tau}_i^\pm$ to zero outside of $K^\pm$ we obtain functions in Σ_h. Let us introduce the notation

$$\boldsymbol{r}_e^\pm = \begin{cases} \boldsymbol{r}_e & \text{on } K_e^\pm, \\ 0 & \text{elsewhere.}\end{cases}$$

We can now represent $\boldsymbol{r}_e^-$, $\boldsymbol{r}_e^+$ on $K_e^\pm$ in the basis $\{\boldsymbol{\tau}_i^\pm\}$, i.e., $\boldsymbol{r}_e^\pm = \sum_{i=1}^N r_{e,i}^\pm \boldsymbol{\tau}_i^\pm$, and we introduce

$$\vec{\boldsymbol{b}}_e = -\frac{1}{2}\left[\int_e [\![u]\!]_e \cdot \boldsymbol{\tau}_{e,i}\right]_{i=\overline{1,N}}, \quad \vec{\boldsymbol{r}}_e^\pm = \left[r_{e,i}^\pm\right]_{i=\overline{1,N}}, \quad M_e^\pm = \left[\int_{K_e^\pm} \boldsymbol{\tau}_i^\pm \cdot \boldsymbol{\tau}_j^\pm\right]_{i,j=\overline{1,N}}.$$

By definition of the lifting operators (2.5) we compute

$$\int_{K_e^\pm} \boldsymbol{r}_e^\pm \cdot \boldsymbol{\tau}_i^\pm = \int_\Omega \boldsymbol{r}_e \cdot \boldsymbol{\tau}_i^\pm = -\int_e [\![u]\!]_e \cdot \{\!\{\boldsymbol{\tau}_i^\pm\}\!\}_e = -\frac{1}{2}\int_e [\![u]\!]_e \cdot \boldsymbol{\tau}_{e,i} \;= \vec{\boldsymbol{b}}_{e,i}\,. \tag{3.5}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Consequently, we have

$$\vec{\boldsymbol{r}}_e^{\pm} = (M_e^{\pm})^{-1}\vec{\boldsymbol{b}}_e \ .$$

Since $\int_{K^\pm}(\boldsymbol{r}_e^{\pm})^2 = \vec{\boldsymbol{r}}_e^{\pm}\cdot M_e^{\pm}\vec{\boldsymbol{r}}_e^{\pm}$ we conclude that

$$\|\boldsymbol{r}_e\|^2_{K_e^{\pm}} = (M_e^{\pm})^{-1}\vec{\boldsymbol{b}}_e\cdot\vec{\boldsymbol{b}}_e = \frac{1}{|K_e^{\pm}|}|\vec{\boldsymbol{b}}_e|^2 \ .$$

For the last equality we used the fact that we are working with an orthonormal set of basis functions so that the mass matrix satisfies $M_e^{\pm} = |K^{\pm}|I$, where I is the identity matrix in $\mathbb{R}^{N\times N}$. Thus we can choose $\alpha_e = \frac{|K_e^-|}{|K_e^+|}$ in (3.3). To show (3.4) we notice first that $\|\boldsymbol{r}_e\|^2_{K_e^-} = (1+\alpha_e)^{-1}\|\boldsymbol{r}_e\|^2_\Omega$, since $\|\boldsymbol{r}_e\|^2_\Omega = \|\boldsymbol{r}_e\|^2_{K_e^-} + \|\boldsymbol{r}_e\|^2_{K_e^+}$. Now, it is easy to see from Lemma 3 that (3.4) is fulfilled. ∎

Note that although we used an orthonormal basis function in the proof of Lemma 4, the coercivity result does not depend on this choice. Also note that the assumption of Lemma 4 requires that the grid is conforming and that each grid element is an affine mapping of a same reference element. These grids include, for example, simplicial grids, Cartesian grids, or a grid whose elements are parallelograms in two dimensions, or parallelepipeds in three dimensions.

Finally, our analysis requires the following estimate on the lifting operator $\boldsymbol{r}_e$, with a proof in [8].

LEMMA 5 (see Brezzi et al. [8]). *There is a positive constant C_2 independent of h_e and u such that*

$$\|\boldsymbol{r}_e(\llbracket u\rrbracket_e)\|^2_\Omega \le C_2 h_e^{-1}\|\llbracket u\rrbracket_e\|^2_e$$

for each $u\in V_h$ and for each $e\in\Gamma$.

3.2. Proof of Theorem 2 for the CDG and CDG2 methods. To carry out the proof of Theorem 2 we first rewrite the bilinear form using parameters β_1, $\widetilde{\boldsymbol{L}_e}$, and $\widetilde{\boldsymbol{\delta}_e}$ and assuming $A\equiv 1$:

$$\begin{aligned} B(u_h,\varphi) = &\int_\Omega \nabla u_h\cdot\nabla\varphi + \chi\sum_{e\in\Gamma}\int_\Omega \widetilde{\boldsymbol{L}_e}(\varphi)\cdot\widetilde{\boldsymbol{L}_e}(u_h)\\ &+\sum_{e\in\Gamma}\int_e (\{\!\{\nabla\varphi\}\!\}_e\cdot\llbracket u_h\rrbracket_e + \{\!\{\nabla u_h\}\!\}_e\cdot\llbracket\varphi\rrbracket_e)\\ &+\beta_1\sum_{e\in\Gamma_i}\int_e \boldsymbol{\beta}_e\cdot\Big([\nabla u_h]_e\llbracket\varphi\rrbracket_e + [\nabla\varphi]_e\llbracket u_h\rrbracket_e\Big)\\ &-\sum_{e\in\Gamma}\int_e \widetilde{\boldsymbol{\delta}}_e(u_h)\cdot\llbracket\varphi\rrbracket_e \quad \forall\varphi\in V_h \ . \end{aligned} \tag{3.6}$$

We have $\beta_1 = 1$ for CDG, and $\beta_1 = 0$ for all other methods, $\widetilde{\boldsymbol{L}_e}$ is zero for IP, equal to $\boldsymbol{L}_e$ for CDG, CDG2, and $\widetilde{\boldsymbol{L}_e}(\,\cdot\,) = \boldsymbol{r}_e(\llbracket\,\cdot\,\rrbracket_e)$ for BR2, and finally $\widetilde{\boldsymbol{\delta}_e} = \boldsymbol{\delta}_e$ for IP, CDG, CDG2 and zero for BR2.

First we prove the coercivity result for the CDG method.

Proof (*for the CDG method*). We note that the bilinear form of the CDG method can be written in the following form:

$$B(u,u) = \|\nabla u + \boldsymbol{L}(u)\|^2_\Omega - \|\boldsymbol{L}(u)\|^2_\Omega + \chi\sum_{e\in\Gamma}\|\boldsymbol{L}_e(u)\|^2_\Omega + \sum_{e\in\Gamma}\frac{\eta}{h_e}\|\llbracket u_h\rrbracket_e\|^2_e \, . \tag{3.7}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Since $\boldsymbol{L}_e = 0$ on all K where e is an inflow edge, we have

$$\|\boldsymbol{L}(u)\|_\Omega^2 \leq N_{\mathcal{T}_h}^{out} \sum_{e\in\Gamma} \|\boldsymbol{L}_e(u)\|_\Omega^2 \ . \tag{3.8}$$

For any $\varepsilon \in (0,1)$ we obtain

$$\|\nabla u + \boldsymbol{L}(u)\|_\Omega^2 \geq \sum_{K\in\mathcal{T}_h} (1-\varepsilon)|u|_{1,K}^2 + (1-\varepsilon^{-1})\|\boldsymbol{L}(u)\|_\Omega^2 \ ,$$

which follows from the Cauchy–Schwarz inequality. Furthermore, from Lemma 5, inequality (3.8), equality (3.4), and the last inequality we get

$$B(u,u) \geq \sum_{K\in\mathcal{T}_h} (1-\varepsilon)|u|_{1,K}^2 + \sum_{e\in\Gamma} \left(\frac{4}{1+\alpha_e}(\chi - \varepsilon^{-1} N_{\mathcal{T}_h}^{out}) + \eta C_2^{-1} \right) \|\boldsymbol{r}_e\|_\Omega^2.$$

If condition (a) of Theorem 2 is satisfied we can choose η large enough that the coefficient of the second sum becomes positive. If condition (b) is satisfied, we can choose ε so close to 1 that $\chi - \varepsilon^{-1} N_{\mathcal{T}_h}^{out}$ becomes positive. The positive number C from the formulation of the theorem is now the minimal value of the positive numbers $1-\varepsilon$ and $\frac{4}{1+\alpha_e}(\chi - \varepsilon^{-1} N_{\mathcal{T}_h}^{out}) + \eta C_2^{-1}$ ∎

We now continue with the proof for the CDG2 method.

Proof (for the CDG2 method). In the case of the CDG2 method the bilinear form $B(u,u)$ can be rewritten as

$$B(u,u) = \|\nabla u + \boldsymbol{r}(u)\|_\Omega^2 - \|\boldsymbol{r}(u)\|_\Omega^2 + \chi \sum_{e\in\Gamma} \|\boldsymbol{L}_e(u)\|_\Omega^2 + \sum_{e\in\Gamma} \frac{\eta}{h_e} \|[\![u]\!]_e\|_e^2 \ .$$

Note that $\boldsymbol{r}_e(\cdot) \equiv 0$ in grid elements not having e as one edge. From that fact we derive the inequality

$$\|\boldsymbol{r}(u)\|_\Omega^2 \leq N_{\mathcal{T}_h} \sum_{e\in\Gamma} \|\boldsymbol{r}_e([\![u]\!]_e)\|_\Omega^2 \ . \tag{3.9}$$

As in the proof for the CDG method we get

$$\|\nabla u + \boldsymbol{r}(u)\|_\Omega^2 \geq \sum_{K\in\mathcal{T}_h} (1-\varepsilon)|u|_{1,K}^2 + (1-\varepsilon^{-1})\|\boldsymbol{r}(u)\|_\Omega^2$$

for any $\varepsilon \in (0,1)$. Combining Lemma 5, inequality (3.9), and the last inequality we obtain

$$B(u,u) \geq \sum_{K\in\mathcal{T}_h} (1-\varepsilon)|u|_{1,K}^2 + \sum_{e\in\Gamma} \left(\frac{4\chi}{1+\alpha_e} - \varepsilon^{-1} N_{\mathcal{T}_h} + \eta C_2^{-1} \right) \|\boldsymbol{r}_e\|_\Omega^2 \ .$$

Note that $\alpha_e = 0$ for $e \in \Gamma \backslash \Gamma_i$. By similar arguments as in the CDG case, we conclude that we can choose η large enough if condition (a) of Theorem 2 is satisfied, or ε close to 1 if condition (b) is satisfied, so that both coefficients in front of $|u|_{1,K}$ and $\|\boldsymbol{r}_e\|_\Omega^2$ are positive. The positive number C from the formulation of the theorem is now the minimal value of the positive numbers $1-\varepsilon$ and $\frac{4\chi}{1+\alpha_e} - \varepsilon^{-1} N_{\mathcal{T}_h} + \eta C_2^{-1}$. ∎

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

3.3. Further remarks. Following the discussion from [2] one can prove for bounded, coercive, consistent, and adjoint consistent methods the following a priori error estimate for the discrete solution of a linear elliptic problem $u_h \in V_h$:

$$\|u - u_h\|_\Omega \leq C h^{k+1} |u|_{k+1,\Omega} \tag{3.10}$$

if the exact solution u is in $H^{k+1}(\Omega) \cap H_0^1(\Omega)$. Here C is a constant and k is the polynomial degree of the basis functions from V_h. All the methods studied in this paper fall into this category. It is straightforward to see that all the methods studied here are consistent and adjoint consistent; Theorem 2 gives conditions for their coercivity, and boundedness can be shown using the estimates found in [2].

From the coercivity results one can conclude the L^2-*stability* estimate $\frac{d}{dt}\int_\Omega u^2 \leq 0$ in the parabolic case for each method. The stability proof for the case that both convection and diffusion terms are present can be treated as described, for example, in [13] under suitable conditions on the numerical flux $\widehat{f}$. The arguments found there demonstrate that it suffices to concentrate on the diffusion part.

In the case of the CDG2 method the switch function can be chosen such that $\chi_0 = \frac{N_{\mathcal{T}_h}}{2}$, independent of the underlying grid as we will show in the following section. This observation turns out to be crucial, because choosing the smallest lifting and stability factor which guarantee stability leads to the most efficient method.

We conclude this section with the observation that on some special meshes, i.e., Cartesian meshes as well as triangular meshes created out of Cartesian meshes by dividing each quadrilateral into two triangles, the BR2 and CDG2 methods coincide when applied to linear problems.

COROLLARY 6 (BR2 and CDG2 on special meshes). *Consider the setting of Theorem 2. The methods BR2 and CDG2 coincide on grids $\mathcal{T}_h$ with $|K_1| = |K_2|$ and equal shape for all $K_1, K_2 \in \mathcal{T}_h$ if $\chi_{BR2} = 2\chi_{CDG2}$ and $\boldsymbol{\delta}_e(u) \equiv 0$.*

Proof. This result is a consequence of observing that in this situation $\int_\Omega \boldsymbol{L}_e(u) \cdot \boldsymbol{L}_e(\varphi) = 4\int_{K_e^-} \boldsymbol{r}_e([\![u]\!]_e)\cdot\boldsymbol{r}_e([\![\varphi]\!]_e)$ and $\int_\Omega \boldsymbol{r}_e([\![u]\!]_e)\cdot\boldsymbol{r}_e([\![\varphi]\!]_e) = 2\int_{K_e^-} \boldsymbol{r}_e([\![u]\!]_e)\cdot\boldsymbol{r}_e([\![\varphi]\!]_e)$ holds. The last equality follows from $\int_{K_e^+} \boldsymbol{r}_e([\![u]\!]_e)\cdot\boldsymbol{r}_e([\![\varphi]\!]_e) = \int_{K_e^-} \boldsymbol{r}_e([\![u]\!]_e)\cdot\boldsymbol{r}_e([\![\varphi]\!]_e)$ as can be shown using change of variables since there exists an affine mapping F of K_e^+ into K_e^- such that $r_e([\![u]\!]_e)(\boldsymbol{x}) = r_e([\![u]\!]_e)(F(\boldsymbol{x}))$, $r_e([\![\varphi]\!]_e)(\boldsymbol{x}) = r_e([\![\varphi]\!]_e)(F(\boldsymbol{x}))$, and $|\det(D(F(\boldsymbol{x}))/D\boldsymbol{x})| = 1$ for all $\boldsymbol{x} \in K_e^+$. Thus for the two methods the term $\int_\Omega \widetilde{\boldsymbol{L}_e}(\varphi)\cdot\widetilde{\boldsymbol{L}_e}(u_h)$ in (3.6) differs only by a factor of 2. ∎

4. Implementation. In this section we provide some details on our implementation of the compact DG methods presented in this paper, focusing once again on the CDG and CDG2 methods. Our implementation is part of the DUNE-FEM module [16], which is based on the free software environment DUNE [6, 7]. DUNE-FEM provides a range of different methods for solving general systems of nonlinear partial differential equations on parallel, locally adapted grids.

Employing the DG method for the spatial discretization of a convection-diffusion-reaction equation leads to

$$\int_\Omega \varphi \partial_t u_h = B_{CDR}(u_h, \varphi) \quad \forall \varphi \in V_h \tag{4.1}$$

for the semi-discrete function u_h with $u_h(t,\cdot) \in V_h$. The bilinear form B_{CDR} is described in section 2.2. Writing $u_h(t,x) = \sum_i u_i(t)\varphi_i(x)$, where $\{\varphi_i\}_i$ forms a basis of V_h, we arrive at a system of ODEs

$$U'(t) = M^{-1}b(t), \quad \text{where} \quad U = (u_i)_i\,, \quad b(t) = \big(B_{CDR}(u_h(t,\cdot),\varphi_i)\big)_i$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

and M is the mass matrix $\left(\int_\Omega \varphi_i\varphi_j\right)_{i,j}$. Thus solving the system of evolution equations requires computing the bilinear form $B_{CDR}(u_h(t,\cdot),\varphi_i)$, the inverse of the mass matrix, and solving the resulting system of ODEs.

4.1. Spatial discretization. In the DUNE-FEM framework either Lagrange type basis functions or orthonormal basis functions are available to build the discrete space V_h. For the work presented here we have used an orthonormal basis, so that the mass matrix is diagonal. All our numerical experiments have shown that taking $\eta = 0$ in the CDG and CDG2 methods leads to the best results if χ is taken according to Theorem 2 part (b). Therefore we will only present details of the implementation for the schemes with $\eta = 0$.

Assuming now that the test function $\varphi = \varphi_i$ has support on only one element K of the grid $\mathcal{T}_h$ and denoting with u_K the restriction of u_h to K, B given in (2.4) takes on a much simpler form. We use the definition of the jump and average operators and the unified formulation of the fluxes found at the beginning of section 3.2 to arrive at

$$\begin{aligned} B_{CDR}(u_h,\varphi) = &-\int_K (f(u_K)\cdot\nabla\varphi + s_E(u_K)\varphi) + \int_{e\subset\partial K} \widehat{f}(u)\cdot\boldsymbol{n}_e\varphi \\ &+ \int_K (A(u_K)\nabla u_K\cdot\nabla\varphi - s_I(u_K)\varphi) - \int_{e\subset\partial K} \widehat{A}(u)\cdot\boldsymbol{n}_e\varphi \\ &- \int_{e\subset\partial K\cap\Gamma_i} \left(\frac{1}{2} + \beta_1\boldsymbol{\beta}_e\cdot\boldsymbol{n}_e\right) A(u_K)[\![u]\!]_e\cdot\nabla\varphi \\ &- \int_{e\subset\partial K\cap\partial\Omega} (u_K - g_D)A(u_K)\boldsymbol{n}_e\cdot\nabla\varphi \\ &=: B_E(u_h,\varphi) + B_I(u_h,\varphi)\,, \end{aligned} \tag{4.2}$$

with $\beta_1 = 1$ for CDG and 0 for all other methods. In B_E we combine the first two integrals which discretize the convection forces and a part of the source term s_E. The second term B_I contains the diffusion and the remaining part of the source term denoted with $s_I = s - s_E$. This splitting is used to employ a semi-implicit (IMEX) ODE solver as described later in this section.

It remains to study the implementation of the convective flux $\widehat{f}$ and the flux $\widehat{A}$ for the diffusion in the primal formulation (2.4). For the convection we use the *Rusanov flux* described, for example, in [39]. Using (3.2) we obtain the following representation for the diffusion flux $\widehat{A}(u)$:

$$\widehat{A}(u) = \begin{cases} \{\!\{A(u)\nabla u + \chi\big(A(u)\boldsymbol{r}_e([\![u]\!]_e)\big)\}\!\}_e & \text{BR2,} \\ \big(A(u)\nabla u + 2\chi A(u)\boldsymbol{r}_e([\![u]\!]_e)\big)|_{K_e^-} & \text{CDG,} \\ \{\!\{A(u)\nabla u\}\!\}_e + 2\chi\big(A(u)\boldsymbol{r}_e([\![u]\!]_e)\big)|_{K_e^-} & \text{CDG2.} \end{cases} \tag{4.3}$$

Hence we see that for the BR2 method, as well as for the CDG, CDG2 methods, we need only compute the lifting $\boldsymbol{r}_e$. While for the BR2 method $\boldsymbol{r}_e$ must be computed on both elements K_e^-, K_e^+, for the CDG and CDG2 we have to compute the lifting only on K_e^- depending on the switch function $\boldsymbol{\beta}_e$. To compute $\boldsymbol{r}_e$ on K_e^- or on K_e^+ we can make use of the fact that we are using orthonormal basis functions. Thus the degrees of freedom $(r_i)_i$ defining $\boldsymbol{r}_e([\![u]\!]_e)$ on K are easily computed through $r_i = -\frac{1}{2}\int_e [\![u]\!]_e\cdot\boldsymbol{\tau}_i$, where $(\boldsymbol{\tau}_i)_i$ is an orthonormal basis of $\mathcal{P}_k(K)^d$. The lifting required for the CDG and CDG2 methods therefore involves only the computation of a single integral over each intersection $e\in\Gamma_i$ and can be computed while the flux $\widehat{A}$ over that intersection is evaluated.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

It remains to fix the switch $\boldsymbol{\beta}_e$ for each interior edge $e \in \Gamma_i$. One suggestion, which is widely used in the literature [13, 2, 36], is the *upwind switch*

$$\boldsymbol{\beta}_e = \frac{1}{2}\operatorname{sgn}(\boldsymbol{n}_{K_e} \cdot \boldsymbol{w})\boldsymbol{n}_{K_e} \ . \tag{4.4}$$

The *upwind* vector $\boldsymbol{w} \in \mathbb{R}^d$ is chosen a priori (i.e., before each time step), so that $\boldsymbol{w} \cdot \boldsymbol{n} \neq 0$ for the normal $\boldsymbol{n}$ to ∂K for all $K \in \mathcal{T}_h$. Thus K_e^- is chosen to be the element adjacent to e with $\boldsymbol{n}_{K_e^-} \cdot \boldsymbol{w} > 0$.

Our suggestion is motivated by the coercivity estimate for the CDG2 method. According to Theorem 2, the CDG2 method is stable provided $\chi > \frac{N_{\mathcal{T}_h}}{4}\big(1 + \nu(\boldsymbol{\beta})\big)$. The efficiency of the method is severely influenced by the magnitude of χ, larger values increasing the condition number of the system matrix. Thus it is advantageous to render $\nu(\boldsymbol{\beta}) = \max_{e\in\Gamma_i}\{|K_e^-|/|K_e^+|\}$ as small as possible. This can be achieved by using the *area switch*

$$\boldsymbol{\beta}_e = \frac{1}{2}\boldsymbol{n}_{K_e^*} \ , \tag{4.5}$$

where K_e^* is the element adjacent to e with the smaller area. Thus $|K_e^-| \leq |K_e^+|$ and consequently $\nu(\boldsymbol{\beta}) \leq 1$.

4.2. Temporal discretization. Using the splitting defined in (4.2) together with the method of lines described above leads to a system of ODEs

$$U'(t) = F_E(U(t), t) + F_I(U(t), t), \tag{4.6}$$

where $U(t) = (u_i(t))_i$ is the vector of degrees of freedom for the unknown function $u_h(t, x) = \sum_i u_i(t)\varphi_i(x)$ and the components of F_E and F_I are $M^{-1}B_E(u_h(t, \cdot), \varphi_i)$ and $M^{-1}B_I(u_h(t, \cdot), \varphi_i)$, respectively. To avoid the strong time step restriction imposed by the diffusion and stiff sources (combined into s_I), we want to use an implicit method for these terms, while allowing for time exact simulation of the convection forces, through an explicit treatment of these terms.

For the numerical examples we use IMEX Runge–Kutta methods of order $k + 1$, where k is the polynomial order of the basis functions used to construct V_h. For $k = 1$ we use the IMEX-SSP2(2,2,2); details on the corresponding Butcher array can be found in [35]. A detailed convergence analysis for the IMEX-SSP2(2,2,2) and a DG method for convection-diffusion equations is carried out in [9]. For $k = 2$ we use the method YZ(3,3) presented in [40], and for $k = 3$ we use the IERK(4,5) presented in [33].

Using these methods the time step Δt_E is merely restricted by the CFL condition. Note that fully implicit methods could also be used, in which case the bilinear form is not split into two parts.

Implicit methods naturally lead to large systems of nonlinear equations, increasing in size if higher order Runge–Kutta methods are used. In our implementation we reduce the size by using diagonally implicit methods, but nevertheless the resulting nonlinear algebraic system is large and difficult to solve. To avoid the computation of the Jacobian of the function F_I we use a matrix free Newton solver. Since our time step is restricted by the convection forces, the number of iterations required for the nonlinear and the subsequent linear solver is small. Hence, the data from the previous time step is a good starting point for the iteration and matrix free implementations are, in general, less expensive compared to setting up matrices.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

In each step of the Newton iteration a linear system of the form $DF_I^n \delta^n = -F_I^n$, where $F_I^n = F_I(U^n, t)$ required the evaluation of the bilinear form B_I. This linear system is solved using a CG type method (BiCG-stab or GMRES for nonsymmetric A). Instead of using the exact Jacobian we use a one sided approximation to DF_I^n. Since the iterative solvers used require only the implementation of the application of the linear operator DF_I^n to given vectors $v^1, \dots, v^S$, we can use the following approximation $DF_I^n v^s = \frac{1}{\varepsilon^n}\left(F_I(U^n + \varepsilon^n v^s) - F_I^n\right)$. Thus only one application of the bilinear form is required in each iteration step $s = 1, \dots, S$ of the linear solver. The step size ε^n is computed following the suggestions found in [31]: $\varepsilon = \varepsilon_{\mathrm{dbl}} \frac{1+|U^n|}{|v|^2}$.

4.3. Choice of parameters in the CDG and CDG2 methods. In contrast to the IP method, CDG is quite robust with respect to the choice of the penalty parameter η. As was already pointed out in [36], the original CDG method (taking $\chi = 1$) is very stable even with $\eta = 0$ at least on 2D triangular grids. In this case, we have not been able to construct examples where the system matrix has negative eigenvalues, at least in the case of the Laplace operator. Only when used to discretize the operator $-\nabla\cdot\big(A(x)\nabla u(x)\big)$, we found problems with stability in some cases, which could be solved by a slight increase of either η or χ. A second case, where we observed negative eigenvalues even for the Laplace operator when taking the CDG method in its original formulation, was on some quadrilateral grids.

The three-dimensional (3D) setting was not tested in [36] and here we did encounter problems even for the Laplace equation when using the CDG method with $\chi = 1$. For Test Case 1 of the *3D Benchmark on Discretization Schemes for Anisotropic Diffusion Problems on General Grids* (cf. [21]) using mesh `tet.0.msh`, we discovered that with $\chi = 1$ the minimal eigenvalue of the stiffness matrix was -12.167 so that the bilinear form B is not coercive in this case. This example demonstrates the necessity of using a constant $\chi > 1$. In numerical experiments we discovered that χ can be chosen smaller than the bounds given in Theorem 2. However, these values differ from method to method, from problem to problem, and from grid to grid. For those reasons we choose for the CDG, CDG2, and BR2 methods $\chi = \chi_0$ in all numerical experiments presented in section 5.

For our comparison with the much simpler IP method, we choose the parameter η on each $e \in \Gamma$ as suggested in [1]:

$$\eta_e = \frac{1}{2} \max_{\substack{K \in \mathcal{T}_h \\ e \in \partial K}} k(k+1)\, \lambda_{max}^A \sum_{e' \in \partial K} \frac{\Lambda_{e'} |e'|^2}{|e|^2} \quad \forall e \in \Gamma. \tag{4.7}$$

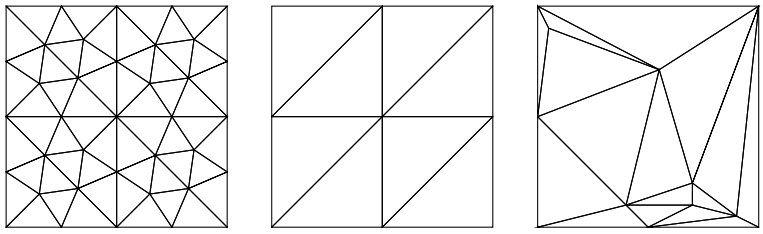
Here k is the polynomial order of the basis functions and λ_{max}^A denotes the largest eigenvalue of the diffusion matrix A. $\Lambda_e = 1/2$ if $e \in \Gamma_i$, $\Lambda_e = 1$ if e is part of the Dirichlet boundary, and $\Lambda_e = 0$ for Neumann boundaries. In [1] it is shown that choosing $\frac{\eta_e}{h_e}$ in the stability term with h_e defined in section 2.1 results in coercivity for the IP method on triangular grids in two space dimensions. Alternative choices of the parameter η_e are, for example, given in [19, 20].

5. Numerical results. In order to obtain some insight into the qualitative characteristics of the CDG2 method, we will compare CDG2 with the other methods mentioned in Table 1. In particular we will consider numerical examples for scalar equations—starting with Poisson's equation to measure the condition of the system matrices and continuing with the heat equation to compare runtime efficiency of the schemes. Finally we conclude this section with results for the nonlinear compressible

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Navier–Stokes system. Further models for which we have employed these methods include flow problems with detailed chemical reaction and free surface shallow water flow.

We focus on a comparison of the efficiency of the methods, i.e., we compare their error to runtime ratio. In all numerical examples we choose the parameters η and χ according to the theoretical bound given by Theorem 2. For the CDG, CDG2, and BR2 methods this means that $\eta = 0$ and stability is achieved by choosing χ equal to the bound χ_0 given by Theorem 2. For the IP method the penalization coefficient η is chosen according to (4.7).



(a) mesh1 (b) structured mesh (c) unstructured mesh

FIG. 1. *Two structured and one unstructured macrotriangulation.*

5.1. Elliptic problem. In the first example we consider a linear elliptic problem of the form

$$
\begin{aligned}
-\nabla \cdot (A \nabla u) &= 0 \quad \text{in } \Omega = [0,1]^2, \\
u &= g \quad \text{on } \partial\Omega, \qquad\qquad A = \begin{pmatrix} 1 & 0 \\ 0 & \varepsilon \end{pmatrix}.
\end{aligned} \tag{5.1}
$$

If the boundary conditions are defined by $g(x,y) = \sin(2\pi x) e^{-2\pi\sqrt{1/\varepsilon} y}$, then $u \equiv g$ is a solution to this problem. For the following numerical experiment we choose $\varepsilon = 10^3$. This problem corresponds to Test 2 of the *Benchmark on Anisotropic Diffusion Problems* presented at the FVCA 5 [28]. Note that for the CDG scheme results for all test cases of the benchmark can be found in [14]. For the current $3D$ *Benchmark on Anisotropic Diffusion Problems* results for the CDG2 method can be found in [30].

The calculations are performed on `mesh1` of the benchmark, presented in Figure 1(a). For the EOC calculation this mesh is refined by quartering all elements in each step. The tolerance for the linear solver has been chosen sufficiently small to obtain optimal convergence rates. The parameters for the DG methods are (according to Theorem 2) BR2 $\chi = 3$, CDG $\chi = 2$, and CDG2 $\chi = 1.5$. For the IP method the stability coefficient has been chosen as described in (4.7). For the CDG method we used the *upwind switch* and for the CDG2 method the *area switch*.

All methods considered show the expected convergence rate of $k+1$, k being the polynomial order of the basis functions used to build V_h. Differences in terms of L^2-accuracy between BR2, CDG, and CDG2 on a fixed grid are usually lower than 5%, but for the IP method we get up to 25% higher L^2-errors. In Figure 2 we compare L^2-error and computational time required for solving the linear system. We see that BR2 and CDG2 are the most efficient methods. This still holds when an ILU(0) preconditioner is applied. A comparison of the CPU time for the unconditioned linear

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

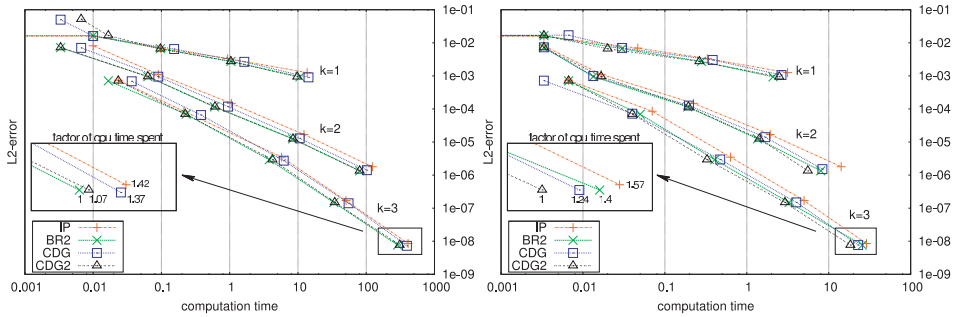


FIG. 2. *CPU time to L^2-error comparison for the elliptic problem* (5.1). *Left, the CPU time for the CG solver. Right, CPU time for the CG solver and $ILU(0)$ preconditioning is plotted. Solver and preconditioner from PETSc* [3] *have been used.*

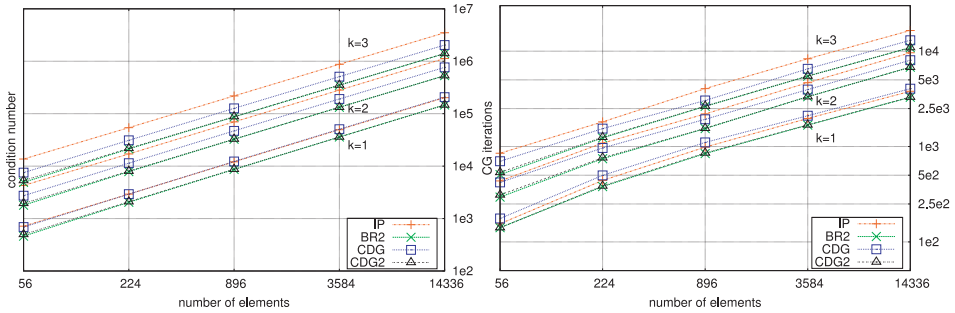


FIG. 3. *Condition number (left), calculated with the Krylov–Shur method implemented in SLEPc* [29], *and iterations of the CG solver (right) for the Poisson equation on a triangular grid for* $k = 1, 2, 3$.

solver makes sense in our setting since we want to apply the methods in a matrix free solver where the application of preconditioning is very difficult.

In Figure 3 we present the condition number which is $\lambda_{max}/\lambda_{min}$ where λ_{max} is the largest eigenvalue of the bilinear form B and λ_{min} the smallest eigenvalue. The eigenvalues have been calculated using the Krylov–Shur method implemented in the software package SLEPc [29]. As we can see, CDG2 and BR2 have the smallest condition number followed by CDG, while IP has the largest condition number. In this example a smaller condition number coincides with fewer iterations of the linear solver needed to achieve a certain reduction of the error, plotted in the right part of Figure 3. We can see that for this example the CDG2 and the BR2 methods seem to be the best choice since they are the most accurate and efficient. In a matrix free implementation, CDG2 could be expected to have a slight advantage over BR2 since the evaluation of the bilinear form is more expensive for the BR2 method, where the lifting operator $\boldsymbol{r}_e$ has to be evaluated twice as much as for the CDG2 method. The approximation quality and the efficiency of the IP method are inferior to those of BR2, CDG, and CDG2 in our example. Additionally, although for the IP method no lifting operator has to be evaluated, the calculation of the parameter η_e in (4.7) depends on the largest eigenvalue of the diffusion matrix A. These computations of eigenvalues can become very difficult for nonconstant A or for nonlinear systems of equations, e.g., the Navier–Stokes equations. On the other hand, just taking a global

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

upper bound for the eigenvalue leads to an over excessive penalization, increasing the condition number of the system matrix and thus the efficiency of the scheme.

5.2. Linear parabolic problem. In the second example we consider a 2D linear advection-diffusion equation of the form

$$\begin{aligned} \partial_t u + \nabla \cdot (u\boldsymbol{v}) - \varepsilon \Delta u = 0 \quad &\text{on } [0,1]^2 \times [0, t_{\text{end}}], \\ u(\cdot, 0) = u_0 \quad &\text{on } [0,1]^2, \end{aligned} \tag{5.2}$$

for which we can construct an exact analytical solution which we use to prescribe the Dirichlet boundary data. We take a constant velocity $\boldsymbol{v} = (0.1, 0.2)$ and $\varepsilon = 0.1$. The initial data is given by $u_0(x,y) = \sum_{n=1}^{N} u_0^n(x,y)$ with $u_0^n(x,y) = (\alpha_{x,n}\cos(\gamma_{x,n}\pi x) + \beta_{x,n}\sin(\gamma_{x,n}\pi x))(\alpha_{y,n}\cos(\gamma_{y,n}\pi y) + \beta_{y,n}\sin(\gamma_{y,n}\pi y))$ with $N = 2$ and $\alpha_x = (0.6, 0.9)$, $\alpha_y = (1.2, 0.3)$, $\beta_x = (0.8, 0.2)$, $\beta_y = (0.4, 0.1)$, and $\gamma_x = (2, 0.7)$, $\gamma_y = (1, 0.5)$. With this choice we have

$$u(t,x,y) = \sum_{n=1}^{N} e^{-\pi\varepsilon t(\gamma_{x,n}+\gamma_{y,n})} u_0^n(x,y) \ .$$

In Figure 4 we compare the efficiency of the different schemes by studying runtimes and discretization errors at $t_{\text{end}} = 0.1$ in the L^2 norm. For this problem we show only results for $k = 2$ but use two different grids. The first grid is a regular *criss-cross* grid (see Figure 1(b)) obtained refining a macrogrid consisting of a Cartesian grid where each cube element is split into two triangles. Thus each element of the grid has the same volume so that CDG2 and BR2 are identical in this case (see Corollary 6). This can be seen in the results since on a fixed refinement level, both CDG2 and BR2 lead to the same error. In this case the CDG2 method is more than 10% more efficient, due to the lower cost in evaluating the bilinear form B. IP is the least efficient scheme, followed by CDG. The second grid is a highly unstructured grid (see Figure 1(c)). Due to the larger value of $\nu(\boldsymbol{\beta})$, the CDG2 method with the upwind switch is considerably less efficient than BR2 and in this case even less efficient than CDG. Using the area switch leads to $\nu(\boldsymbol{\beta}) \leq 1$ and the method is again the most efficient one, outperforming BR2 again by more than 10%.

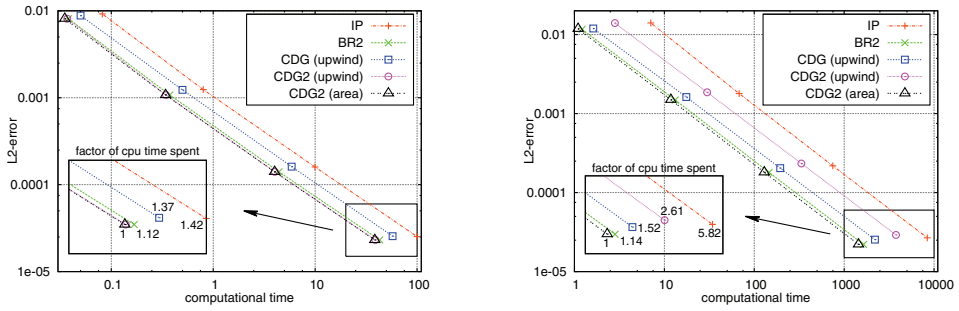


FIG. 4. *Results for the advection-diffusion problem. The L^2-error on a triangular grid for $k = 2$ is plotted with respect to the runtime. On the left, a criss-cross grid is used, leading to theoretical parameters of $\chi = 2$ (CDG), $\chi = 3$ (BR2), and $\chi = 1.5$ (CDG2). On the right, results on the unstructured grid are shown; the theoretical parameters for this setting are the same for all methods except for the CDG2 method with upwind switch where we have $\chi = 8.4375$. For the IP method the stability coefficient has been chosen as described in* (4.7).

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

5.3. Nonlinear parabolic problem. As a third example we consider the time-dependent compressible nonlinear Navier–Stokes equations

$$\partial_t \boldsymbol{u} + \nabla \cdot (F(\boldsymbol{u}) - A(\boldsymbol{u})\nabla \boldsymbol{u}) = \boldsymbol{s}(\boldsymbol{u}) \qquad \text{in } \Omega \times [0, t_{\text{end}}], \tag{5.3}$$

where $\boldsymbol{u} = (\rho, \rho \boldsymbol{v}, \rho e)$, ρ is density, $\rho \boldsymbol{v}$ momentum vector, and ρe total energy. In this section we will restrict ourselves to two dimensions for the sake of simpler notation. The convective and diffusive fluxes are given as

$$F(\boldsymbol{u}) = \begin{bmatrix} \rho u & \rho w \\ \rho u^2 + p & \rho uw \\ \rho uw & \rho w^2 + p \\ u(\rho e + p) & w(\rho e + p) \end{bmatrix}, \quad A(\boldsymbol{u})\nabla \boldsymbol{u} = \begin{bmatrix} 0 & 0 \\ \tau_{11} & \tau_{12} \\ \tau_{21} & \tau_{22} \\ E_1^{\text{diff}} + \kappa \partial_x T & E_2^{\text{diff}} + \kappa \partial_z T \end{bmatrix}$$

with $E_1^{\text{diff}} = u\tau_{11} + w\tau_{21}$ and $E_2^{\text{diff}} = u\tau_{12} + w\tau_{22}$. The viscous stress tensor $\boldsymbol{\tau}$ for Newtonian fluids is defined as

$$\boldsymbol{\tau} = \begin{bmatrix} (2\mu + \lambda)\partial_x u + \lambda \partial_z w & \mu(\partial_x w + \partial_z u) \\ \mu(\partial_x w + \partial_z u) & \lambda \partial_x u + (2\mu + \lambda)\partial_z w \end{bmatrix}; \tag{5.4}$$

κ is thermal conductivity coefficient. The term $\kappa \nabla T$ represents the heat flux according to the *Fourier's law*. In the convective and diffusive fluxes we have other unknowns p and T, which are related to $\boldsymbol{u}$ by

$$p = (\gamma - 1)\rho \left(e - \frac{1}{2}\boldsymbol{v}^2 \right), \qquad T = p/(\rho R_d), \tag{5.5}$$

where $R_d = c_p - c_v$ is the specific gas constant of the dry air and $\gamma = c_p/c_v$ is the ratio of specific heat capacity at constant pressure c_p and at constant volume c_v. Finally, we close the system of equations with $\lambda = -\frac{2}{3}\mu$ and $\kappa = c_p \mu \text{Pr}^{-1}$, where Pr is the Prandl number.

For this test case we choose parameters and exact solution similar to [24]:

$$\rho = e = \frac{1}{2}\sin(\pi(x + y) - t) + 2 \quad \text{and} \quad \boldsymbol{v} = (1, 1),$$

with $\mu = 0.1$, $\text{Pr} = 0.72$, $c_p = 1004$, and $c_v = 717$. The source term $\boldsymbol{s}$ in (5.3) is chosen so that we have an analytical solution. The computational domain is $\Omega = [0, 2]^2$.

In Figure 5 we compare CDG2 with area switch with CDG and BR2 at the end-time $t_{\text{end}} = 0.1$. We use the same time integration scheme for all methods of the same order. These time integrators have been described in section 4.2. On the left picture of Figure 5 we observe that the CDG2 method of order 2, 3, and 4 is at least 10% faster in terms of the CPU time than BR2 of the corresponding order, and more than 35% than CDG, whereas the difference in the L^2-error is less than 4% for all methods. We also observe that the GMRES solver, in case of the CDG method, needs 54% more iterations for the whole simulation than in the case of CDG2 and BR2, which require the same number of iterations.

6. Conclusions. We have presented a new method for spatial discretization of the diffusion fluxes based on the discontinuous Galerkin (DG) approach. The new method is derived from the compact discontinuous Galerkin (CDG) method in such a way that inherits all positive characteristics of the CDG method, most notably its compact stencil. For CDG2 and CDG we give new necessary conditions for coercivity

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

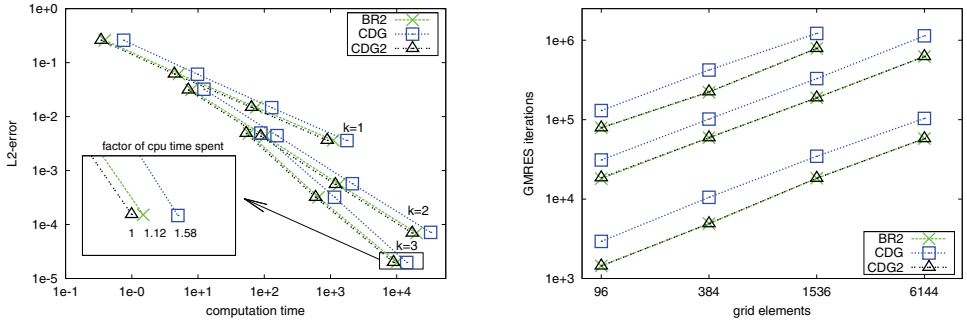
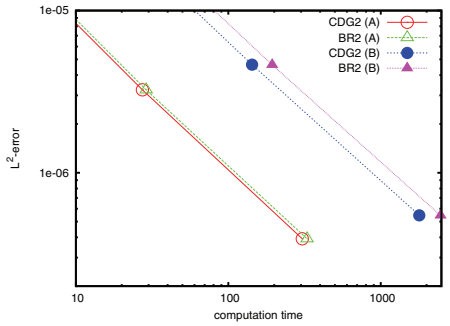


FIG. 5. *Results for Navier–Stokes equations on an unstructured triangular grid. The stabilization parameters are* $\eta = 0$ *and* χ *according to Theorem* 2.



A

Scheme	CPU time	L^2-error
CDG2	305	3.91e-07
BR2	329	3.92e-07

B

Scheme	CPU time	L^2-error
CDG2	1790	5.47e-07
BR2	2469	5.46e-07

FIG. 6. *Comparison on affine* (A) *and nonaffine* (B) *quadrilateral grids. Problem is from section* 5.2 *on the quadrilateral domain with corners* $(0,0),(1,0),(1,1),(0,1)$ (A) *and* $(0.4,0),(1,0),(1,1.4),(0.1)$ (B). *The graph (left) contains levels* 4 *and* 5 *of the simulation cycle and the table (right) contains the numbers of the level* 5 *run.*

for linear problems. Furthermore, we show that these conditions can be successfully used for nonlinear Navier–Stokes equations. Numerical examples for Poisson, linear heat, and nonlinear Navier–Stokes equations show that CDG2 is more efficient than CDG in terms of L^2-error versus CPU time. On very regular grids the CDG2 method is identical to the BR2 method whereas the spatial operator of the CDG2 is slightly cheaper to evaluate.

Our results indicate that even on triangular grids using orthonormal basis functions the CDG2 method is more efficient than BR2, even though in this case the lifting operator is cheap to evaluate. In Figure 6 we show results on two quadrilateral grids, the second one requiring nonaffine element transformations. We use a natural extension of the bounds in Theorem 2 to define χ. Note that in this situation the mass matrix is no longer diagonal and the lower complexity of the CDG2 method in this case clearly leads to a more efficient method. Due to the still missing theoretical justification of bounds for the parameter values this situation still requires more thorough testing.

Acknowledgments. The authors would like to thank the reviewers and the editor for their various important remarks and suggestions. Computations have been performed using the DUNE library, in particular the DUNE-FEM package.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

REFERENCES

[1] M. Ainsworth and R. Rankin, *Constant free error bounds for nonuniform order discontinuous Galerkin finite-element approximation on locally refined meshes with hanging nodes*, IMA J. Numer. Anal., 31 (2011), pp. 254–280.

[2] D. N. Arnold, F. Brezzi, B. Cockburn, and L. D. Marini, *Unified analysis of discontinuous Galerkin methods for elliptic problems*, SIAM J. Numer. Anal., 39 (2002), pp. 1749–1779.

[3] S. Balay, J. Brown, K. Buschelman, V. Eijkhout, W. Gropp, D. Kaushik, M. Knepley, L. Curfman McInnes, B. Smith, and H. Zhang, *PETSc Users Manual*, Technical Report ANL-95/11 - Revision 3.1, Argonne National Laboratory, Argonne, IL, 2010.

[4] 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 (1997), pp. 267–279.

[5] F. Bassi, S. Rebay, G. Mariotti, S. Pedinotti, and M. Savini, *A high-order accurate discontinuous finite element method for inviscid turbomachinery flows*, in Proceedings of the 2nd European Conference on Turbomachinery Fluid Dynamics and Thermodynamics, Technologisch Instituut, Antwerp, Belgium, 1997, pp. 99–108.

[6] P. Bastian, M. Blatt, A. Dedner, C. Engwer, R. Klöfkorn, R. Kornhuber, M. Ohlberger, and O. Sander, *A generic grid interface for parallel and adaptive scientific computing.* II. *Implementation and tests in DUNE*, Computing, 82 (2008), pp. 121–138.

[7] P. Bastian, M. Blatt, A. Dedner, C. Engwer, R. Klöfkorn, M. Ohlberger, and O. Sander, *A generic grid interface for parallel and adaptive scientific computing.* I. *Abstract framework*, Computing, 82 (2008), pp. 103–119.

[8] F. Brezzi, G. Manzini, D. Marini, P. Pietra, and A. Russo, *Discontinuous Galerkin approximations for elliptic problems*, Numer. Methods Partial Differential Equations, 16 (2000), pp. 365–378.

[9] E. Burman and A. Ern, *Implicit-Explicit Runge-Kutta Schemes and Finite Elements with Symmetric Stabilization for Advection-Diffusion Equations*, preprint, 2010; available online at http://hal.archives-ouvertes.fr/hal-00530378/fr/.

[10] E. Burman and P. Zunino, *A domain decomposition method based on weighted interior penalties for advection-diffusion-reaction problems*, SIAM J. Numer. Anal., 44 (2006), pp. 1612–1638.

[11] B. Cockburn, S. Hou, and C.-W. Shu, *TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws.* IV. *The multidimensional case*, Math. Comp., 54 (1990), pp. 545–581.

[12] B. Cockburn, S. Y. Lin, and C.-W. Shu, *TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws.* III. *One-dimensional systems*, J. Comput. Phys., 84 (1989), pp. 90–113.

[13] B. Cockburn and C.-W. Shu, *The local discontinuous Galerkin method for time-dependent convection-diffusion systems*, SIAM J. Numer. Anal., 35 (1998), pp. 2440–2463.

[14] A. Dedner and R. Klöfkorn, *The compact discontinuous Galerkin method for elliptic problems*, in Finite Volumes for Complex Applications V, ISTE, London, 2008, pp. 761–776.

[15] A. Dedner and R. Klöfkorn, *A generic stabilization approach for higher order discontinuous Galerkin methods for convection dominated problems*, J. Sci. Comput., 47 (2011), pp. 365–388.

[16] A. Dedner, R. Klöfkorn, M. Nolte, and M. Ohlberger, *A generic interface for parallel and adaptive scientific computing: Abstraction principles and the* Dune-Fem *module*, Computing, 89 (2010), pp. 165–196.

[17] D. A. Di Pietro, A. Ern, and J.-L. Guermond, *Discontinuous Galerkin methods for anisotropic semidefinite diffusion with advection*, SIAM J. Numer. Anal., 46 (2008), pp. 805–831.

[18] J. Douglas, Jr., and T. Dupont, *Interior penalty procedures for elliptic and parabolic Galerkin methods*, in Computing Methods in Applied Sciences, Lecture Notes in Phys. 58, Springer, Berlin, 1976, pp. 207–216.

[19] Y. Epshteyn and B. Rivière, *Estimation of penalty parameters for symmetric interior penalty Galerkin methods*, J. Comput. Appl. Math., 206 (2007), pp. 843–872.

[20] A. Ern, A. F. Stephansen, and P. Zunino, *A discontinuous Galerkin method with weighted averages for advection-diffusion equations with locally small and anisotropic diffusivity*, IMA J. Numer. Anal., 29 (2009), pp. 235–256.

[21] R. Eymard, G. Henry, R. Herbin, F. Hubert, R. Klöfkorn, and G. Manzini, 3*D benchmark on discretization schemes for anisotropic diffusion problems on general grids*, in Proceedings of the 6th International Symposium on Finite Volumes for Complex Applications, to

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

appear.
[22] M. Feistauer, V. Kučera, K. Najzar, and J. Prokopová, *Analysis of space-time discontinuous Galerkin method for nonlinear convection-diffusion problems*, Numer. Math., 117 (2011), p. 251–288.
[23] M. Feistauer and V. Kučera, *A new technique for the numerical solution of the compressible Euler equations with arbitrary Mach numbers,* in Hyperbolic Problems: Theory, Numerics and Applications, Proceedings of the 11th International Conference on Hyperbolic Problems, Ecole Normale Supérieure, Lyon, France, 2006, Springer, Berlin, 2008, pp. 523–531.
[24] G. Gassner, F. Lörcher, and C.-D. Munz, *A discontinuous Galerkin scheme based on a space-time expansion.* II. *Viscous flow equations in multidimensions*, J. Sci. Comput., 34 (2008), pp. 260–286.
[25] J.-L. Guermond, R. Pasquetti, and B. Popov, *Entropy viscosity method for nonlinear conservation laws*, J. Comput. Phys., 230 (2011), pp. 4248–4267.
[26] K. Harriman, P. Houston, B. Senior, and E. Süli, *hp-version discontinuous Galerkin methods with interior penalty for partial differential equations with nonnegative characteristic form*, in Recent Advances in Scientific Computing and Partial Differential Equations (Hong Kong, 2002), Contemp. Math. 330, AMS, Providence, RI, 2003, pp. 89–119.
[27] R. Hartmann and P. Houston, *An optimal order interior penalty discontinuous Galerkin discretization of the compressible Navier-Stokes equations*, J. Comput. Phys., 227 (2008), pp. 9670–9685.
[28] R. Herbin and F. Hubert, *Benchmark on discretization schemes for anisotropic diffusion problems on general grids*, in Finite Volumes for Complex Applications V, ISTE, London, 2008, pp. 659–692.
[29] V. Hernandez, J. Roman, and V. Vidal, *SLEPc: A scalable and flexible toolkit for the solution of eigenvalue problems*, ACM Trans. Math. Software, 31 (2005), pp. 351–362.
[30] R. Klöfkorn, *Benchmark* 3*D: The compact discontinuous Galerkin* 2 *scheme*, in Proceedings of the 6th International Symposium on Finite Volumes for Complex Applications, to appear.
[31] D. A. Knoll and D. E. Keyes, *Jacobian-free Newton-Krylov methods: A survey of approaches and applications*, J. Comput. Phys., 193 (2004), pp. 357–397.
[32] D. Kröner, *Numerical Schemes for Conservation Laws*, Wiley & Teubner, Stuttgart, 1997.
[33] E. Lindblad, D. Valiev, B. Müller, J. Rantakokko, P. Lötstedt, and M. Liberman, *Implicit-explicit Runge-Kutta method for combustion simulation*, in European Conference on Computational Fluid Dynamics ECCOMAS CFD, P. Wesseling, E. Oñate, and J. Periaux, eds., TU Delft, Delft, The Netherlands, 2006.
[34] H. Liu and J. Yan, *The Direct Discontinuous Galerkin (DDG) method for diffusion with interface corrections*, Commun. Comput. Phys., 8 (2010), pp. 541–564.
[35] L. Pareschi and G. Russo, *Implicit-explicit Runge-Kutta schemes and applications to hyperbolic systems with relaxation*, J. Sci. Comput., 25 (2005), pp. 129–155.
[36] J. Peraire and P.-O. Persson, *The compact discontinuous Galerkin (CDG) method for elliptic problems*, SIAM J. Sci. Comput., 30 (2008), pp. 1806–1824.
[37] W. Reed and T. Hill, *Triangular Mesh Methods for the Neutron Transport Equation*, Technical Report LA-UR-73-479, Los Alamos National Laboratory, Los Alamos, NM, 1973.
[38] V. Sobotíková and M. Feistauer, *Effect of numerical integration in the DGFEM for nonlinear convection-diffusion problems*, Numer. Methods Partial Differential Equations, 23 (2007), pp. 1368–1395.
[39] E. Toro, *Riemann Solvers and Numerical Methods for Fluid Dynamics. A Practical Introduction*, 2nd ed., Springer-Verlag, Berlin, 1999.
[40] J. J. Yoh and X. Zhong, *New hybrid Runge-Kutta methods for unsteady reactive flow simulation*, AIAA J., 42 (2004), pp. 1593–1600.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.