On the flexibility of agglomeration based physical space discontinuous Galerkin discretizations

Francesco Bassi, Lorenzo Botti, Alessandro Colombo, Daniele Antonio Di Pietro, Pietro Tesini

To cite this version:

Francesco Bassi, Lorenzo Botti, Alessandro Colombo, Daniele Antonio Di Pietro, Pietro Tesini. On the flexibility of agglomeration based physical space discontinuous Galerkin discretizations. Journal of Computational Physics, 2011, 231 (1), pp.45-65. 10.1016/j.jcp.2011.08.18 . hal-00562219v2

HAL Id: hal-00562219

https://hal.science/hal-00562219v2

Submitted on 26 Feb 2013

On the flexibility of agglomeration based physical space discontinuous Galerkin discretizations

F. Bassi[a], L. Botti[a,b,*], A. Colombo[a], D. A. Di Pietro[c], P. Tesini[d]

[a]*Dipartimento di Ingegneria Industriale, Università di Bergamo*
via Marconi 4, 24044 Dalmine (BG), Italy
[b]*Biomedical Engineering Department, Mario Negri Institute for Pharmacological Research,*
Villa Camozzi, via Camozzi 3, Ranica (BG), Italy
[c]*IFP Energies nouvelles, 1 & 4 av. du Bois Préau, 92500 Rueil-Malmaison, France*
[d]*SKF Engineering & Research Centre, Modelling & Simulation Department,*
MDC RKs-2, SE-415 50, Göteborg, Sweden

Abstract

In this work we show that the flexibility of the discontinuous Galerkin (dG) discretization can be fruitfully exploited to implement numerical solution strategies based on the use of elements with very general shapes. Thanks to the freedom in defining the mesh topology, we propose a new h-adaptive technique based on agglomeration coarsening of a fine mesh. The possibility to enhance the error distribution over the computational domain is investigated on a Poisson problem with the goal of obtaining a mesh independent discretization.

The main building block of our dG method consists of defining discrete polynomial spaces directly on physical frame elements. For this purpose we orthonormalize with respect to the L^2-product a set of monomials relocated in a specific element frame and we introduce an easy way to reduce the cost related to numerical integration on agglomerated meshes. To complete the dG formulation for second order problems, two extensions of the BR2 scheme to arbitrary polyhedral grids, including an estimate of the stabilization parameter ensuring the coercivity property, are here proposed.

Keywords: Diffusion equation, Laplacian discretization, Discontinuous Galerkin methods, Polyhedral elements, Orthonormal hierarchical basis functions, Reduced numerical integration, Meshfree, h-adaptivity
PACS: 02.70.Dh, 02.60.Lj, 41.20.Cv

---

1. Introduction

In recent years, there has been a growing interest in developing CFD solvers capable of working on unstructured polyhedral grids. Among the reasons for this

---

[*] Corresponding author

Email addresses: `francesco.bassi@unibg.it` (F. Bassi), `lorenzo.botti@unibg.it` (L. Botti), `alessandro.colombo@unibg.it` (A. Colombo), `dipietrd@ifpenergiesnouvelles.fr` (D. A. Di Pietro), `pietro.tesini@skf.com` (P. Tesini) interest are better achievable grid quality and improved accuracy of numerical solutions. Unlike standard FV methods, dG methods are pretty insensitive to the shape of mesh elements, and apparently there would be no special reason to extend the method to element shapes other that those usually employed in unstructured grids (triangles and quadrilaterals in 2D, tetrahedra, prisms, pyramids and hexahedrals in 3D). Instead, just the flexibility offered by the high-order dG discretization suggests and makes it straightforward to implement the method on elements of any shape in order to fully exploit numerical strategies that already proved successful, and new ones. The geometric h-multigrid method is an example of the former while an agglomeration-based h-adaptivity is an example of the latter.

In this context, three are three main objectives in this work. First, we propose a dG implementation capable of computing high-order numerical solutions of second order PDEs on meshes resulting from the agglomeration of elements belonging to an underlying (possibly very) fine mesh. In this approach the high-order polynomial approximation is defined on each agglomerated element, while volume and surface integrals are computed by means of quadrature rules on the elements of the underlying fine mesh. Second, we propose a simple and effective approach to the approximation of complex (possibly curved) domain boundaries based on the use of agglomerated faces. As for agglomerated elements, such faces are simply treated as the union of the underlying fine mesh faces, each with its own geometrical approximation. We will show that this alternative to high-order mesh generation techniques is sufficient to get a geometrical approximation of the domain which does not impair the accuracy of high-order solutions. Third, we investigate the possibility of driving agglomeration to obtain a new approach to h-adaptivity, hopefully leading to potentially mesh independent discretizations, a salient feature typically ascribed to meshfree methods. Compared with mesh modification strategies based on element subdivision, see e.g. Hartmann and Leicht [11], or conforming grid modification, see e.g. Dolejší [7], an h-adaptivity strategy based on agglomeration coarsening of a fine mesh may be preferable from the implementation viewpoint. The underlying mesh does not change, no nodes nor elements need to be dynamically added, moved or removed, and no hierarchic structures are required to track refinement levels.

The agglomeration process relies on standard tools originally developed in the context of geometric multigrid methods [15] which naturally demand for agglomerated meshes. Since coarse meshes can feature elements with very general shapes, the notion of reference element fails to exist and the classical approach based on polynomial basis functions defined in the reference frame does not apply. Also, since arbitrary geometrical order is assumed for the fine mesh, the coarse elements are not necessarily polyhedra, rather the bounding surface may include curved faces. We hence adopt a different point of view, proposed by Tesini [19], where discrete polynomial spaces defined in physical coordinates are proposed as the key ingredient to handle general meshes; see also Di Pietro and Ern [5, 6]. Other approaches to deal with non-standard meshes exist and deserve to be mentioned. A standard (continuous) Galerkin discretization on arbitrary polyhedral elements has been proposed by Rashid and Selimotic [18] in the context of elastostatics and elastic-plastic problems of solid mechanics. In [18], the definition of the Lagrangian basis functions in the physical frame is strictly related to the number of nodes defining the geometry of the element. The approach has been extended to dG methods by Gassner, Lörcher, Munz and Hesthaven [8] in order to obtain a quadrature free approach.

For the sake of simplicity, the focus is here on the two-dimensional case, although the technique readily extends to higher space dimensions. In what follows we denote by $\mathcal{R}$ the (fixed) fine mesh of the domain $\Omega \subset \mathbb{R}^d$, $d = 2$, and by $\mathcal{T}_h$ a coarsening of $\mathcal{R}$ obtained by agglomeration. We consider dG approximations based on the spaces

$$\mathbb{P}_d^k(\mathcal{T}_h) \overset{\text{def}}{=} \left\{v_h \in L^2(\Omega) \mid v_{h|T} \in \mathbb{P}_d^k(T),\ \forall T \in \mathcal{T}_h\right\}, \tag{1}$$

where k is a non-negative integer and $\mathbb{P}_d^k(T)$ denotes the restriction to T of the polynomial functions of two variables and total degree $\leqslant k$. Since the choice of the basis has a major impact on the conditioning of the resulting discrete problems a primary issue to address is the construction of a computationally efficient and satisfactory basis for the space $\mathbb{P}_d^k(\mathcal{T}_h)$. Straightforward implementations based on simple monomial basis functions quickly become unsuitable for high-degree polynomial approximations on highly stretched or distorted elements. The procedure employed in this work consists in defining a starting set of basis functions for each elementary space $\mathbb{P}_d^k(T)$, $T \in \mathcal{T}_h$, and in applying a modified Gram-Schmidt orthogonalization procedure to infer a new orthonormal set of basis functions, see §2.

After having introduced the discrete tools we consider the extension to general meshes of the BR2 dG discretization of the Poisson equation devised in [3]. We provide two estimates of the stabilization parameter, suitable for two different implementations of the method on agglomerated meshes, which mimic the result of Brezzi et al. [4] for simplicial meshes. The major result of §3 is to show that optimal convergence properties are maintained for the discrete polynomial spaces of §2. Moreover, our agglomeration based h-adaptive strategy is validated using the Poisson equation as a model problem and its performance in comparison to uniform h-refinement is discussed.

Finally, an important issue to address when dealing with very general elements is numerical integration. Since, in our case, the computational mesh is obtained by agglomeration of standard elements, a straightforward solution is to deploy this decomposition in conjunction with standard quadrature rules on the elements of the fine mesh $\mathcal{R}$. However, this approach turns out to be expensive for high-order approximations, as the number of quadrature points may become very large and the basis functions have to be evaluated in each physical element separately. To mitigate the computational cost, we propose to replace exact quadratures with reduced ones while ensuring that the error remains within a prescribed tolerance. In §4 numerical validation and profiling is provided for a representative set of test cases.

2. Discrete dG spaces on general meshes

2.1. Coarsening by agglomeration

Let $\Omega \subset \mathbb{R}^d$, $d = 2$, be a bounded connected open domain, and let $\mathcal{R}$ denote a (possibly non conforming) mesh of Ω composed of elements $E \in \mathcal{R}$ such that (i) there exists a finite set of reference elements $\mathcal{R}^{\text{ref}}$ and, for all $E \in \mathcal{R}$, there exists a unique $E^{\text{ref}} \in \mathcal{R}^{\text{ref}}$ and a polynomial mapping $\Psi_E : E^{\text{ref}} \to E$ such that $\Psi \in \mathbb{P}_d^m(E^{\text{ref}})$ for some $m \geqslant 1$ and $E = \Psi_E(E^{\text{ref}})$; (ii) quadrature rules of arbitrary order are available on every $E^{\mathrm{ref}} \in \mathcal{R}^{\mathrm{ref}}$. Most often, the fine mesh $\mathcal{R}$ is obtained via standard mesh generators, and is typically composed of (possibly curved) triangular and quadrangular elements, so that $\mathcal{R}^{\mathrm{ref}}$ simply contains the unit simplex and the unit square. Notice that in the latter case the space $\mathbb{Q}_d^m(E_{\mathrm{ref}})$ replaces $\mathbb{P}_d^m$ in the definition of Ψ_E. Starting from $\mathcal{R}$ we can define a coarsened mesh $\mathcal{T}_h = \{T\}$ by agglomeration. More precisely, we suppose that (i) $\mathcal{T}_h$ is a partition of Ω; (ii) every $T \in \mathcal{T}_h$ is an open bounded connected subset of Ω and there exists $\mathcal{R}_T \subset \mathcal{R}$ such that

$$\overline{T} = \bigcup_{E \in \mathcal{R}_T} \overline{E}.$$

We consider in this work broken polynomial spaces $\mathbb{P}_d^k(\mathcal{T}_h)$, $k \geqslant 0$, of the form (1) based on the coarsened mesh $\mathcal{T}_h$. In the rest of this section we describe our approach to construct a basis for $\mathbb{P}_d^k(\mathcal{T}_h)$ with convenient numerical properties.

2.2. Orthonormal hierarchical polynomial bases in the physical frame

For a given polynomial degree $k \geqslant 0$, let, for the sake of conciseness,

$$V_h \overset{\mathrm{def}}{=} \mathbb{P}_d^k(\mathcal{T}_h).$$

For all $T \in \mathcal{T}_h$, we denote by N_{dof}^T the cardinality of the local polynomial space $\mathbb{P}_d^k(T)$ and set $D_T \overset{\mathrm{def}}{=} \{1, \ldots, N_{\mathrm{dof}}^T\}$. For a given $T \in \mathcal{T}_h$, let $\Phi_T^k = \{\varphi_i^T\}_{i \in D_T}$ denote a basis for $\mathbb{P}_d^k(T)$. The functions φ_i^T, $i \in D_T$, can be extended to Ω by simply setting $\varphi_i^T = 0$ on $\Omega \backslash T$. A basis for the space V_h is then given by

$$\Phi^k \overset{\mathrm{def}}{=} \{\Phi_T^k\}_{T \in \mathcal{T}_h}. \tag{2}$$

By construction, the support of each basis function in Φ^k is contained in exactly one element. As a plain example, for all $T \in \mathcal{T}_h$ one could simply take Φ_T^k equal to the set of monomials in the physical space variable $\mathbf{x} = (x_i)_{1 \leqslant i \leqslant d}$ of total degree $\leqslant k$

$$\Phi_T^k = \{\mathbb{M}_d^{\boldsymbol{\alpha}}(\mathbf{x})\}_{\boldsymbol{\alpha} \in \mathbb{N}^2,\, \|\boldsymbol{\alpha}\|_1 \leqslant k}, \qquad \mathbb{M}_d^{\boldsymbol{\alpha}}(\mathbf{x}) \overset{\mathrm{def}}{=} \prod_{i=1}^{d} x_i^{\alpha_i}. \tag{3}$$

It is convenient to define the coordinates $\mathbf{x}$ with respect to a translated, element-specific, physical frame whose center coincides with the barycenter of T. Following this approach, local bases $\{\Phi_T^k\}_{T \in \mathcal{T}_h}$ for any $k \geqslant 0$ can be obtained irrespective of both the shape of the elements and the quadrature rule. However, this choice is only acceptable for polynomials of moderate-degree on almost isotropic elements, whereas it presents severe drawbacks when high-degree polynomials and/or stretched elements with curved faces are considered. In such a case, the condition number of the elementary mass matrices may become unacceptably large; see Figure 1(b). We show in this work that a better solution consists in constructing a set of basis functions which are orthonormal and hierarchical within physical frame elements by applying a standard orthonormalization procedure to a suitable initial guess.

For all $T \in \mathcal{T}_h$, let $\widehat{\Phi}_T^k = \{\widehat{\varphi}_i^T\}_{i \in D_T}$ denote an initial set of linearly independent basis functions. Choices for this set will be discussed in §2.3. In order to obtain a new set of orthonormal basis functions we apply the modified Gram-Schmidt (MGS) orthogonalization algorithm. The sole requirement to apply this procedure is the capability to compute the integrals of polynomial functions on an agglomerated element T, which is here achieved by exploiting the partition $\mathcal{R}_T$ into standard-shaped sub-elements. The MGS algorithm with re-orthogonalization is setup as follows: For all $T \in \mathcal{T}_h$,

```
for $i = 1$ to $N_{\mathrm{dof}}^T$ do
    for $j = 1$ to $i - 1$ do
        $r_{ij}^T \leftarrow \left(\widehat{\varphi}_i^T, \varphi_j^T\right)_T$
        $\widehat{\varphi}_i^T \leftarrow \widehat{\varphi}_i^T - r_{ij}^T \varphi_j^T$ {Remove the projection of $\widehat{\varphi}_i^T$ onto $\varphi_j^T$}
    end for
    $r_{ii}^T \leftarrow \sqrt{(\widehat{\varphi}_i^T, \widehat{\varphi}_i^T)_T}$
    $\widehat{\varphi}_i^T \leftarrow \widehat{\varphi}_i^T / r_{ii}^T$ {Normalize}
    $\varphi_i^T \leftarrow \widehat{\varphi}_i^T$
end for
```

where $(\cdot,\cdot)_T$ denotes an inner product on T. In all the numerical examples the L^2-product is used, i.e., $(v, w)_T \overset{\mathrm{def}}{=} \int_T vw$ for all v, $w \in L^2(T)$. In practice, the orthonormalization procedure can be applied more than once to get rid of numerical round-off errors. Repeating the procedure twice is generally sufficient to obtain a basis which is orthonormal up to machine precision. This result is in accordance with what reported in [9]. For all $T \in \mathcal{T}_h$, it can be shown that the set of elementary basis functions $\{\varphi_i^T\}_{i \in D_T}$ resulting from the MGS algorithm solves

$$\forall i \in D_T, \qquad \varphi_i^T = \sum_{j=1}^{i-1} a_{ij}^T \varphi_j^T + a_{ii}^T \widehat{\varphi}_i^T, \tag{4}$$

where the coefficients a_{ij}^T are determined by enforcing each new φ_i^T to be orthogonal to the $(i-1)$ basis functions already orthonormalized, whereas the coefficient a_{ii}^T is the normalizing factor for the L^2-norm of the newly created φ_i^T. For all $i \in D_T$ there holds

$$\begin{aligned}
\frac{a_{ij}^T}{a_{ii}^T} &= -\left(\widehat{\varphi}_i^T, \varphi_j^T\right)_T, \qquad j \in \{1, \ldots, i-1\}, \\
\frac{1}{a_{ii}^T} &= \sqrt{(\widehat{\varphi}_i^T, \widehat{\varphi}_i^T)_T - \sum\nolimits_{j=1}^{i-1} \left(\widehat{\varphi}_i^T, \varphi_j^T\right)_T^2},
\end{aligned}$$

and the above coefficients are related to r_{ij}^T and r_{ii}^T in the MGS algorithm by:

$$r_{ij}^T = -\frac{a_{ij}^T}{a_{ii}^T}, \qquad r_{ii}^T = \frac{1}{a_{ii}^T}.$$

Besides being orthonormal, the bases $\{\Phi^k\}_{k \geqslant 0}$ defined by (2) with local bases $\{\Phi_T^k\}_{T \in \mathcal{T}_h}$ obtained according to the above procedure are also hierarchical, i.e.,

$$\forall k \geqslant 1, \qquad \Phi^{k-1} \subset \Phi^k.$$

In fact, adding basis functions to increase the polynomial degree leaves unaltered the equations of the form (4) corresponding to lower degrees.

An important remark is that the actual outputs of the MGS algorithm are the coefficients r_{ii}^T and r_{ij}^T, which can be used to evaluate the orthonormal basis functions and their derivatives at any point in the supporting element T. Indeed, it is a simple matter to check that the spatial derivatives $\partial_k \varphi_i^T$, $k \in \{1, 2\}$, can be evaluated as follows: For all $T \in \mathcal{T}_h$,

```
for $i = 1$ to $N_{\mathrm{dof}}^T$ do
    for $j = 1$ to $i - 1$ do
        $\partial_k \widehat{\varphi}_i^T \leftarrow \partial_k \widehat{\varphi}_i^T - r_{ij}^T \partial_k \varphi_j^T$
    end for
    $\partial_k \widehat{\varphi}_i^T \leftarrow \partial_k \widehat{\varphi}_i^T / r_{ii}^T$
    $\partial_k \varphi_i^T \leftarrow \partial_k \widehat{\varphi}_i^T$
end for
```

The evaluation based on the trivial definition (4) is not recommended in practice, because the computation of the coefficient a_{ii}^T and a_{ij}^T is usually affected by round-off errors.

2.3. Initialization of the MGS algorithm

In this section we investigate how, for a given element $T \in \mathcal{T}_h$, the initial choice $\widehat{\Phi}_T^k$ affects the quality of the basis Φ_T^k resulting from the MGS algorithm. As the final local mass matrix $M_T = \left[(\varphi_i^T, \varphi_j^T)_{L^2(T)}\right]$ is expected to be unit diagonal up to round-off errors, we use the condition number of the initial local mass matrix $\widehat{M}_T = \left[(\widehat{\varphi}_i, \widehat{\varphi}_j)_{L^2(T)}\right]$ to measure the distance of the initial guess from the target. The quality of the final basis is then measured by the following quantity:

$$E_{\nabla\cdot\varphi} \overset{\mathrm{def}}{=} \max_{i \in D_T} |\mathbf{e}_i|, \qquad \mathbf{e}_i = \int_T \boldsymbol{\nabla} \varphi_i - \int_{\partial T} \varphi_i \mathbf{n}_T, \tag{5}$$

where $\mathbf{n}_T$ denotes the outward normal to T. The integrals over T and ∂T are computed numerically by means of quadrature rules on the elements of $\mathcal{R}_T$ and on their faces that are exact for the required polynomial degree. For all $i \in D_T$, $|\mathbf{e}_i|$ is a measure of the numerical conservation with respect to the (exact) integration rules, and should be zero in infinite precision by virtue of (a corollary of) the divergence theorem. This indicator can be related to the quality of the basis with respect to the conservation property, a highly appreciated feature of dG discretizations applied to PDEs in divergence from.

To evaluate the impact of round-off errors on the output of the MGS algorithm, we consider two choices for the initial basis $\widehat{\Phi}_T^k$. In what follows, we denote by $\overline{\mathbf{x}}_T$ the frame associated to an element $T \in \mathcal{T}_h$ whose axes are aligned with the principal axis of inertia of T, and whose center coincides with the barycenter of the element; see Figure 1(a) for an example. We compare the following choices for the initial local basis:

$$\text{Choice 1:} \qquad \widehat{\Phi}_T^k = \left\{ \frac{\mathbb{M}_d^{\boldsymbol{\alpha}}(\mathbf{x})}{\|\mathbb{M}_d^{\boldsymbol{\alpha}}(\mathbf{x})\|_{L^2(T)}} \right\}_{\boldsymbol{\alpha}\in\mathbb{N}^d,\, \|\boldsymbol{\alpha}\|_1\leqslant k}, \tag{6a}$$

$$\text{Choice 2:} \qquad \widehat{\Phi}_T^k = \left\{ \frac{\mathbb{M}_d^{\boldsymbol{\alpha}}(\overline{\mathbf{x}}_T)}{\|\mathbb{M}_d^{\boldsymbol{\alpha}}(\overline{\mathbf{x}}_T)\|_{L^2(T)}} \right\}_{\boldsymbol{\alpha}\in\mathbb{N}^d,\, \|\boldsymbol{\alpha}\|_1\leqslant k}. \tag{6b}$$

In both cases the initial guess is a basis of monomials, but in (6a) the variables are the coordinates of the physical frame $\mathbf{x}$ aligned with the global frame and whose origin coincides with the barycenter of T, while in (6b) the variables are the coordinates of the local element frame $\overline{\mathbf{x}}_T$ which is additionally rotated to account for the geometric anisotropy of the cell. The normalization with respect to the L^2-product enhances the scaling of the monomials. Other choices are possible but will not be considered herein.

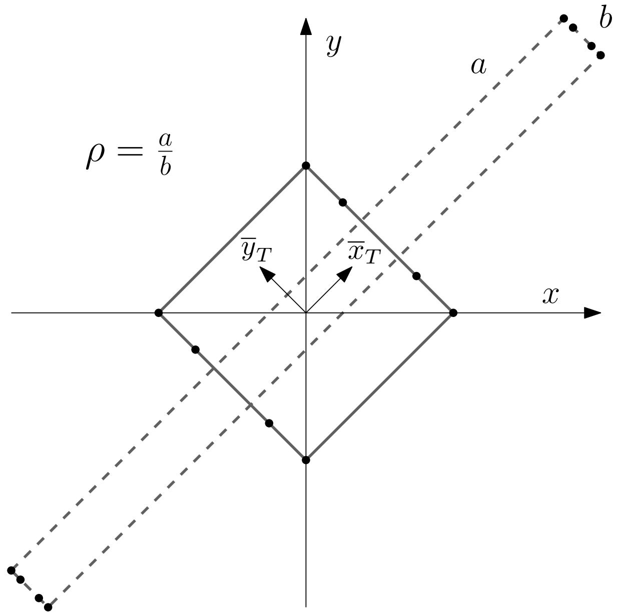
The advantages of the choice (6b) with respect to the choice (6a) are numerically verified by considering a rectangular element stretched along the bisector of the first quadrant; see Figure 1(a). As expected, Figure 1(b) shows that the condition number of the elementary mass matrix $\widehat{M_T}$ is not affected by the element aspect ratio ρ for the choice (6b), whereas it rapidly increases for the choice (6a). As shown in Figure 1(c), for the choice (6a) the error $E_{\nabla\cdot\varphi}$ for the orthonormal set of basis functions resulting from the MGS procedure grows unacceptably large as ρ increases. Conversely, the choice (6b) yields an error $E_{\nabla\cdot\varphi}$ that is almost independent of ρ. Albeit simple, this test indicates that the bad quality of the initial guess $\widehat{\Phi}_T^k$ (measured by the ill-conditioning of elementary mass matrices $\widehat{M_T}$) is eliminated by the MGS process, but the resulting orthonormalized set Φ_T^k is affected by larger numerical errors.

2.4. Numerical examples

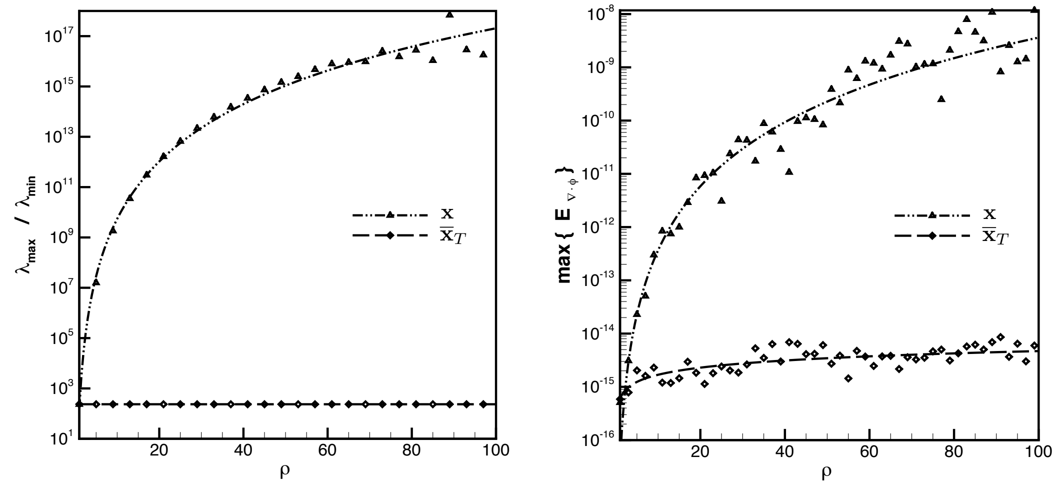
Overview of basis functions. In this section we present a qualitative overview and description of the orthonormal physical space basis function obtained by means of the procedure outlined in §2.2. The orthonormal basis functions composing the local basis Φ_T^4 on the element $T = (-1, 1)^2$ are shown in Figure 2. Up to machine precision they coincide with those generated by means of the Jacobi polynomials; see, e.g., [12]. We stress that, unlike the latter approach, our procedure yields a diagonal unit mass matrix on curved elements, independently of the nature of the reference-to-physical frame mapping. The generality of this result is of some importance, as it simplifies both the matrix assembly stage and the computation of the lifting operators defined by (9). For further details on the computation of lifting operators we refer to [1, §3.2].

In order to appreciate the geometrical flexibility of basis functions defined in the physical space, we present in Figure 3 the elements of Φ_T^4 for an L-shaped polygonal element $T = (-1, 1)^2 \backslash (-0.2, 1) \times (0, 1)$. It is interesting to remark that the slopes of the linear modes are aligned with the principal axes of inertia of the element. The quadrilateral sub-cells composing the L-shaped element considered in Figure 3 are shown in Figure 4(a).

In the context of complex domains with curved boundaries, the generation of meshes suitable for high-order and spectral element discretizations is a non-trivial



(a) Local element frame $\overline{\mathbf{x}}_T$ and global frame $\mathbf{x}$ translated in the barycenter of a rectangular element stretched along the bisector of the first quadrant



(b) Condition number of the mass matrix $\widehat{M}_T$

(c) Divergence error (5)

Figure 1: Comparison of the results obtained with the choices listed in (6) for $k = 4$

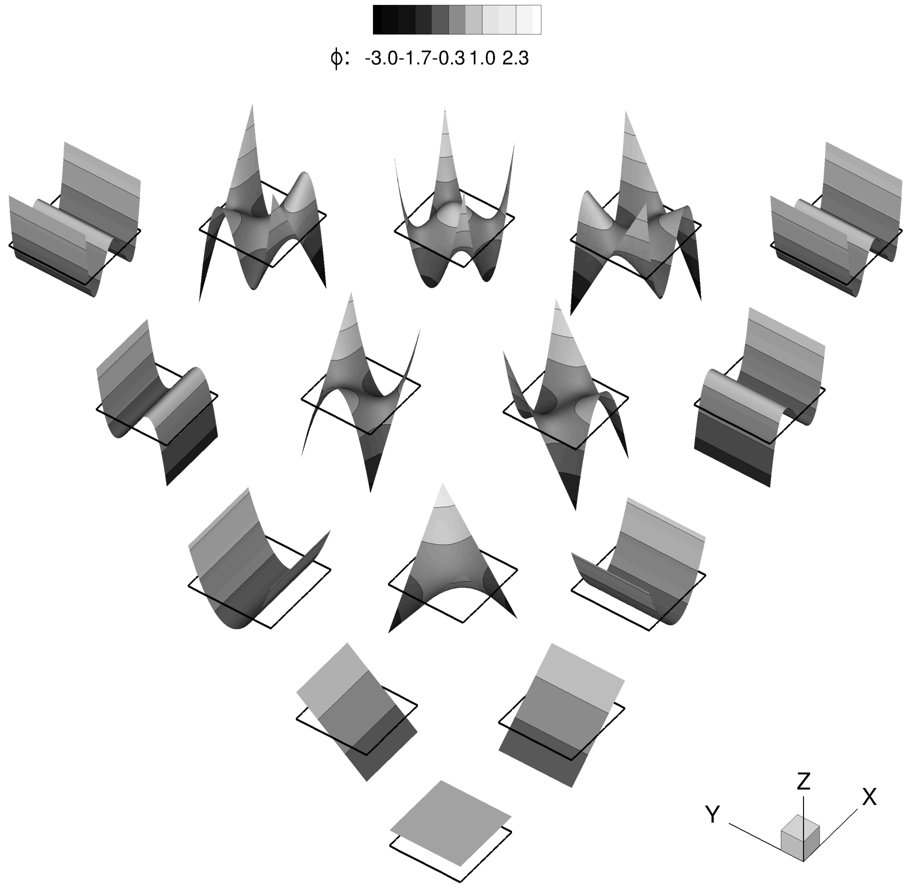


Figure 2: Elements of the orthonormal basis Φ_T^4 with T reference square $(-1,1)^2$.

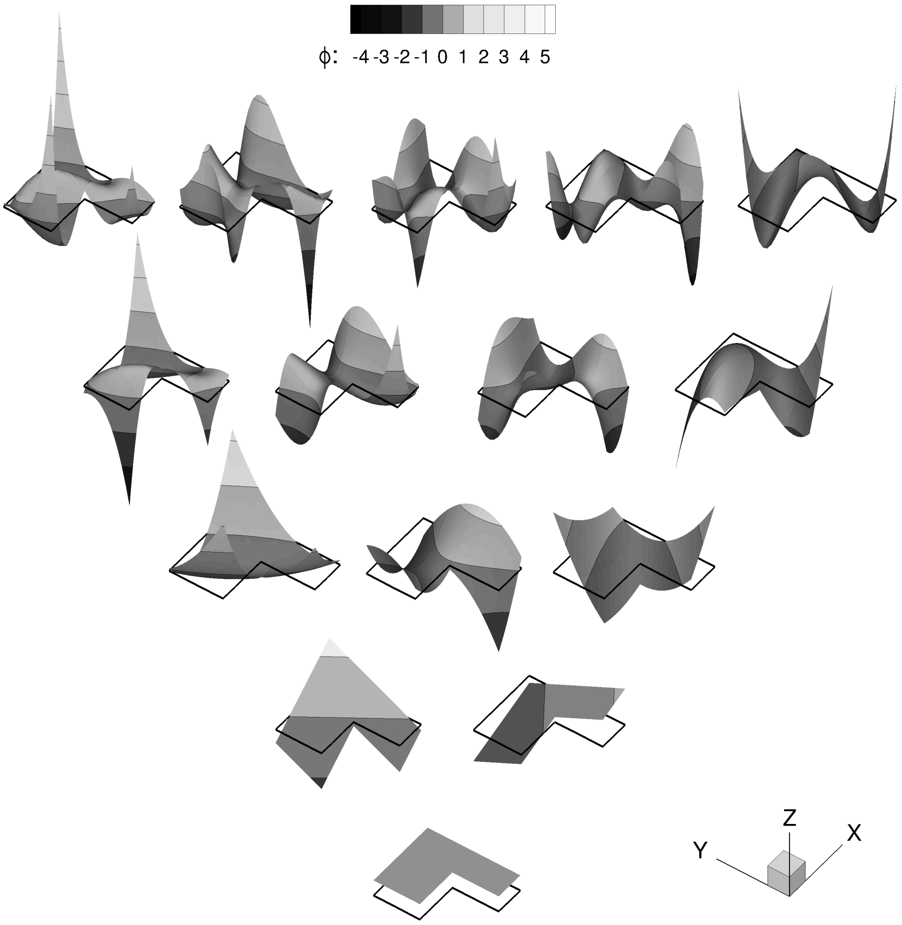
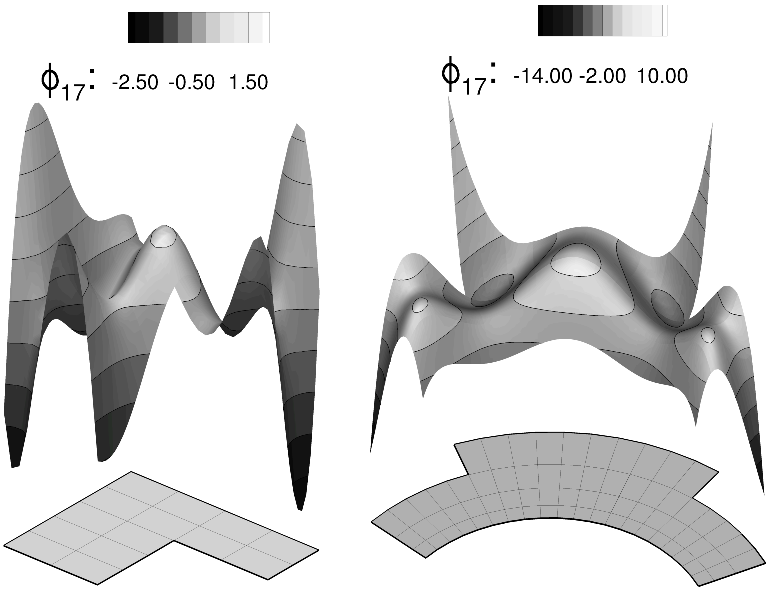


Figure 3: Elements of the orthonormal basis Φ_T^4 with T L-shaped polygonal element.



(a) Affine fine mesh $\mathcal{R}$ (b) Curved fine mesh $\mathcal{R}$

Figure 4: Examples of general elements T and of basis functions obtained from the method outlined in §2. The elements of $\mathcal{R}_T$ are represented in thin line.

task that has often limited the application of such methods in real-life computations; see, e.g. [12, §4.3.5]. The possibility to obtain coarse agglomerated grids on top of fine meshes may reduce the need for high-order boundary discretizations, see §3.4. Indeed, $\mathcal{R}$ can be chosen so as to obtain a suitable geometrical representation of the boundary of Ω since the density of the fine mesh $\mathcal{R}$ does not affect the number of degrees of freedom of the final discretization (but only the number of sub-elements required for numerical integration). An example of an agglomerated element obtained from a curved mesh $\mathcal{R}$ is shown in Figure 4(b).

3. Grid agglomeration and *h*-adaptivity

3.1. Adaptive coarsening

As we have mentioned in the introduction, general meshes can be obtained by means of standard agglomeration algorithms originally developed in the framework of geometric multigrid. We rely, in particular, on the multilevel agglomeration library MGridGen [15] developed by Moulitsas and Karypis. The coarse grid creation is divided in two conceptually distinct phases: first, a sequence of coarse graphs is created; second, starting from the coarse graphs, the refinement process takes place in order to optimize the quality of the final mesh. Several optimization parameters are provided to the user [15]. In all the numerical tests we use the default settings of the library that is, the coarse graphs are obtained by means of a globular agglomeration algorithm, while the uncoarsening/refinement phase minimizes a dual objective function. As described in [16], the weighted sum of aspect ratios and the maximum aspect ratio are considered in order to optimize the quality and the cardinality of the resulting mesh.

To drive the algorithm towards h-adaptivity, we modified the MGridGen library introducing the possibility to locally drive the agglomeration factor. In practice the number of elements of the starting mesh $\mathcal{R}$ that concur in forming an element of $\mathcal{T}_h$ is specified for each $E \in \mathcal{R}$ providing an indication of $\text{card}(R_T)$. We point out that, in practice, the above requirement cannot be satisfied exactly for every element $T \in \mathcal{T}_h$, and the agglomeration indication should be rather considered as an upper bound for $\text{card}(\mathcal{R}_T)$. Although a lower bound for $\text{card}(\mathcal{R}_T)$ can also be enforced, this may result in poor mesh quality for $\mathcal{T}_h$, and this option is therefore not considered in this work.

3.2. The BR2 method on general meshes

To test the performance of dG discretizations on general meshes we consider the BR2 method of [3] for the model problem

$$\begin{cases} -\triangle u = f & \text{in } \Omega, \\ u = 0 & \text{on } \partial\Omega. \end{cases} \tag{7}$$

For an integer $k \geqslant 1$, we look for a discrete solution in the discrete space

$$V_h = \mathbb{P}_d^k(\mathcal{T}_h).$$

We consider hereafter two different implementations of the method corresponding to different definitions of faces for general meshes. In both cases the resulting method is coercive, but the amount of stabilization (and the accuracy of the solution) vary.

Definition 1 (Facet)**.** *We define a* facet σ *of an agglomerated element* $T \in \mathcal{T}_h$ *as a portion of* ∂T *such that there exist* $E \in \mathcal{R}_T$ *and a (hyperplanar) face* σ^{ref} *of the corresponding reference element* E^{ref} *such that* σ *is the image of* σ^{ref} *through the mapping* Ψ_E*. Facets are collected in the set* $\mathcal{F}_h^\sigma$*.*

Definition 2 (Mesh face)**.** *We define a* mesh face Σ *of an agglomerated element* $T \in \mathcal{T}_h$ *as a portion of* ∂T *such that either* $\Sigma = \partial T \cap \partial\Omega$ *or there exists* $T' \in \mathcal{T}_h$*,* $T' \neq T$*, such that* $\Sigma = \partial T \cap \partial T'$*. Mesh faces are collected in the set* $\mathcal{F}_h^\Sigma$*. Moreover, for every face* Σ *we introduce the set* $\sigma_h^\Sigma \subset \mathcal{F}_h^\sigma$ *collecting the facets partitioning* Σ*, i.e.,*

$$\overline{\Sigma} = \bigcup_{\sigma \in \sigma_h^\Sigma} \overline{\sigma},$$

Example 1 (Mesh faces and facets)**.** *Consider the case in Figure 5. The number of mesh faces of the elements A, B, C is* 4, 6 *and* 4*, respectively; see Figure 5(b). On the other hand, as depicted in Figure 5(a), the number of facets is* 26, 20 *and* 22 *respectively.*

In what follows we discuss the impact of each definition on the BR2 method. For the sake of brevity we let $\mathcal{F}_h$ be such that

$$\text{Choice 1:} \qquad \mathcal{F}_h = \mathcal{F}_h^\Sigma, \tag{8a}$$

$$\text{Choice 2:} \qquad \mathcal{F}_h = \mathcal{F}_h^\sigma, \tag{8b}$$

and denote by F the generic element of $\mathcal{F}_h$ (a *face*). We introduce the set of boundary faces $F \in \mathcal{F}_h^{\mathrm{b}}$ such that $F \subset \partial\Omega$ and let $\mathcal{F}_h^{\mathrm{i}} \overset{\mathrm{def}}{=} \mathcal{F}_h \backslash \mathcal{F}_h^{\mathrm{b}}$ denote the set of internal faces. For all $F \in \mathcal{F}_h^{\mathrm{b}}$, $\mathbf{n}_F$ denotes the unit outward normal to Ω, whereas, for all $F \in \mathcal{F}_h^{\mathrm{i}}$ such that $F \subset \partial T_1 \cap \partial T_2$, $\mathbf{n}_F$ is defined as the unit normal pointing out of T_1 (the order of the elements sharing F is arbitrary but fixed). For all $F \in \mathcal{F}_h^{\mathrm{i}}$ and all $v_h \in V_h$ we introduce the jump and average operators defined as follows:

$$[\![v_h]\!] \overset{\mathrm{def}}{=} v_{h|T_1} - v_{h|T_2}, \qquad \{v_h\} \overset{\mathrm{def}}{=} \frac{1}{2}(v_{h|T_1} + v_{h|T_2}).$$

On boundary faces, we conventionally set $[\![v_h]\!] = \{v_h\} = v_h$ (other definitions may be considered to account for different boundary conditions). When applied to vector functions, the jump and average operators act componentwise.

For all $F \in \mathcal{F}_h$ we define the (local) lifting operator $\mathbf{r}_F^k : L^2(F) \to [\mathbb{P}_d^k(\mathcal{T}_h)]^d$, such that, for all $\phi \in L^2(F)$,

$$\int_\Omega \mathbf{r}_F^k(\phi)\cdot\boldsymbol{\tau}_h = \int_F \phi\{\boldsymbol{\tau}_h\}\cdot\mathbf{n}_F \qquad \forall \boldsymbol{\tau}_h \in [\mathbb{P}_d^k(\mathcal{T}_h)]^d. \tag{9}$$

We also introduce the global lifting

$$\mathbf{R}_h^k(\phi) \overset{\mathrm{def}}{=} \sum_{F \in \mathcal{F}_h} \mathbf{r}_F^k([\![\phi]\!]). \tag{10}$$

The bilinear form associated with the BR2 method is given by

$$a_h(u_h, v_h) \overset{\mathrm{def}}{=} \int_\Omega \big(\nabla_h u_h - \mathbf{R}_h^k(u_h)\big)\cdot\big(\nabla_h v_h - \mathbf{R}_h^k(v_h)\big) - \int_\Omega \mathbf{R}_h^k(u_h)\cdot\mathbf{R}_h^k(v_h) + s_h(u_h, v_h), \tag{11}$$

with

$$s_h(u_h, v_h) \overset{\mathrm{def}}{=} \sum_{F \in \mathcal{F}_h} \eta_F \int_\Omega \mathbf{r}_F^k([\![u_h]\!])\cdot\mathbf{r}_F^k([\![v_h]\!]). \tag{12}$$

We consider the following discretization of (7):

$$\text{Find } u_h \in V_h \text{ s.t. } a_h(u_h, v_h) = \int_\Omega f v_h \text{ for all } v_h \in V_h. \tag{13}$$

We observe that the global lifting is independent of the choice (8). Indeed, for all $v_h \in V_h$, there holds

$$\mathbf{R}_h^k(v_h) = \sum_{\Sigma \in \mathcal{F}_h^\Sigma} \mathbf{r}_\Sigma^k([\![v_h]\!]) = \sum_{\Sigma \in \mathcal{F}_h^\Sigma} \sum_{\sigma \in \sigma_h^\Sigma} \mathbf{r}_\sigma^k([\![v_h]\!]) = \sum_{\sigma \in \mathcal{F}_h^\sigma} \mathbf{r}_\sigma^k([\![v_h]\!]).$$

As a result, the BR2 bilinear forms defined according to (8a) or (8b) only differ in the stabilization term s_h. When dealing with general polyhedral meshes, it is important to accurately tune the stabilization parameters η_F, $F \in \mathcal{F}_h$. As a matter of fact, over-penalization may result in ill-conditioned linear systems and therefore spoil the efficiency and the accuracy of the numerical solution [1]. In the next section we propose a lower bound for the stabilization parameter η_F accounting for the differences arising from the different choices listed in (8).

3.3. Stability on general meshes

For all $F \in \mathcal{F}_h$ we introduce the set of elements in whose boundary F is contained, i.e.

$$\mathcal{T}_F \stackrel{\text{def}}{=} \{T \in \mathcal{T}_h \,|\, F \subset \partial T\},$$

which contains exactly two elements if $F \in \mathcal{F}_h^{\text{i}}$ and exactly one if $F \in \mathcal{F}_h^{\text{b}}$. Symmetrically, for all $T \in \mathcal{T}_h$, the set

$$\mathcal{F}_T \stackrel{\text{def}}{=} \{F \in \mathcal{F}_h \,|\, F \subset \partial T\} \tag{14}$$

collects the faces composing the boundary of T. Finally, for all $F \in \mathcal{F}_h$, we define

$$\mathcal{F}_F \stackrel{\text{def}}{=} \bigcup_{T \in \mathcal{T}_F} \mathcal{F}_T,$$

that is to say, the set of faces of the elements whose boundary contains F. Since we denote with F a generic face, $\mathcal{F}_T$ and $\mathcal{F}_F$ have the meaning of sets of facets, see Definition 1, or sets of mesh faces, see Definition 2.

Theorem 1 (Coercivity on general meshes)**.** *Assume that*

$$\forall F \in \mathcal{F}_h, \quad \eta_F > \underline{\eta}_F \stackrel{\text{def}}{=} \max_{T \in \mathcal{T}_F} \operatorname{card}(\mathcal{F}_T). \tag{15}$$

Then, there exists $C > 0$ independent of the meshsize such that

$$\forall v_h, \qquad a_h(v_h, v_h) \geqslant C |\!|\!| v_h |\!|\!|^2,$$

with a_h defined by (11)*, and*

$$|\!|\!| v_h |\!|\!|^2 \stackrel{\text{def}}{=} \|\nabla_h v_h\|^2_{[L^2(\Omega)]^d} + \sum_{F \in \mathcal{F}_h} \|\mathbf{r}_F^k([\![v_h]\!])\|^2_{[L^2(\Omega)]^d}. \tag{16}$$

Even if the proof of Theorem 1 is a trivial extension of the coercivity result obtained by Brezzi et al. [4], we include it hereafter for the sake of completeness.

Proof. Let $v_h \in V_h$. Using the fact that $\operatorname{supp}(\mathbf{r}_F^k([\![v_h]\!]) = \bigcup_{T \in \mathcal{T}_F} T$ we infer

$$\begin{aligned}
\|\mathbf{R}_h^k([\![v_h]\!])\|^2_{L^2(\Omega)^d} &= \sum_{T \in \mathcal{T}_h} \int_T \left| \sum_{F \in \mathcal{F}_T} \mathbf{r}_F^k([\![v_h]\!]) \right|^2 \leqslant \sum_{T \in \mathcal{T}_h} \sum_{F \in \mathcal{F}_T} \operatorname{card}(\mathcal{F}_T) \int_T \left| \mathbf{r}_F^k([\![v_h]\!]) \right|^2 \\
&\leqslant \sum_{T \in \mathcal{T}_h} \sum_{F \in \mathcal{F}_T} \underline{\eta}_F \|\mathbf{r}_F^k([\![v_h]\!])\|^2_{L^2(T)^d} = \sum_{F \in \mathcal{F}_h} \underline{\eta}_F \|\mathbf{r}_F^k([\![v_h]\!])\|^2_{L^2(\Omega)^d}.
\end{aligned}$$

Moreover, using the above result together with the Cauchy–Schwarz and Young inequalities yields for all $\epsilon > 0$

$$\begin{aligned}
a_h(v_h, v_h) &\geqslant (1 - \epsilon) \|\nabla_h v_h\|^2_{L^2(\Omega)^d} + \sum_{F \in \mathcal{F}_h} \eta_F \|\mathbf{r}_F^k([\![v_h]\!])\|^2_{L^2(\Omega)^d} - \frac{1}{\epsilon} \|\mathbf{R}_h^k([\![v_h]\!])\|^2_{L^2(\Omega)^d} \\
&\geqslant (1 - \epsilon) \|\nabla_h v_h\|^2_{L^2(\Omega)^d} + \sum_{F \in \mathcal{F}_h} \left(\eta_F - \frac{\underline{\eta}_F}{\epsilon} \right) \|\mathbf{r}_F^k([\![v_h]\!])\|^2_{L^2(\Omega)^d}.
\end{aligned}$$

The conclusion follows since ϵ is arbitrary. □

Whether we are working with mesh faces as in (8a) or with facets as in (8b), Theorem 1 states that coercivity with respect to the $|\!|\!|\cdot|\!|\!|$-norm is attained setting η_F according to (15). Even if η_F should be greater the maximum number of faces of the elements sharing F, in all the computations presented in this work we modify the stabilization parameter on internal faces and set

$$\eta_F > 1 + \frac{1}{2}\,\mathrm{card}(\mathcal{F}_F \backslash \{F\}), \tag{17}$$

as depicted in the agglomerated mesh of Figure 5. Since it is well known that the values of η_F provided by theory are abundant, the choice of taking the average number of faces of the elements sharing F allows to obtain a more uniform distribution of the penalization parameter. Clearly, on a general mesh, $\mathrm{card}(\mathcal{F}_F \backslash \{F\})$ is larger for the choice (8b), resulting in larger values of the stabilization parameters η_F, $F \in \mathcal{F}_h$. When using facets as in (8b) the stabilization term reads

$$s_h(u_h, v_h) = \sum_{\sigma \in \mathcal{F}_h^\sigma} \eta_\sigma \int_\Omega \mathbf{r}_\sigma^k([\![u_h]\!]) \cdot \mathbf{r}_\sigma^k([\![v_h]\!]) = \sum_{\sigma \in \mathcal{F}_h^\sigma} \eta_\sigma \int_\sigma \{\mathbf{r}_\sigma^k([\![u_h]\!])\} \cdot \mathbf{n}_\sigma [\![v_h]\!]. \tag{18}$$

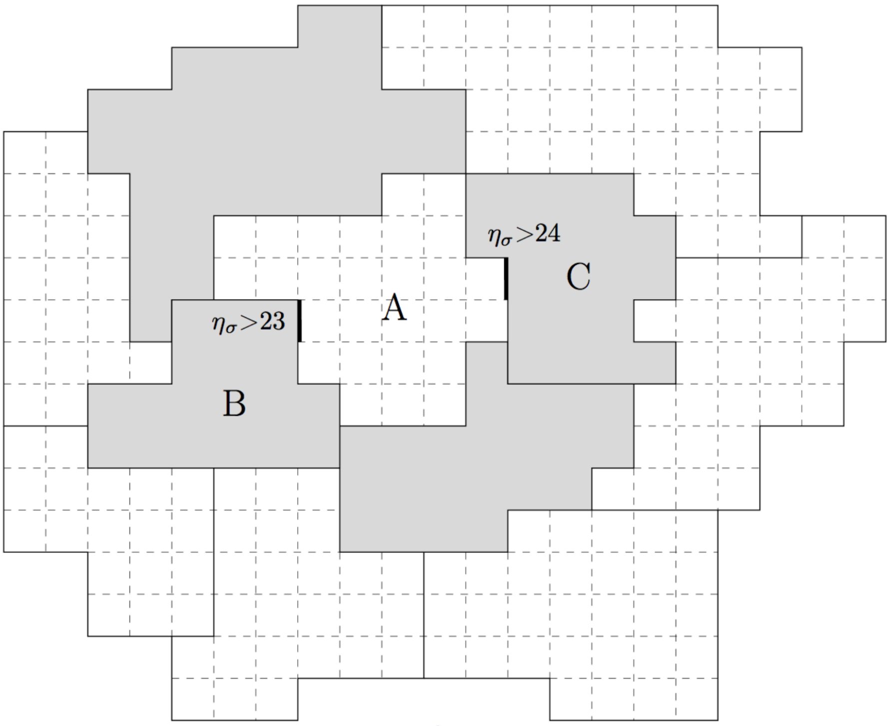
On the other hand, when working with mesh faces as in (8a), we have

$$\begin{aligned} s_h(u_h, v_h) &= \sum_{\Sigma \in \mathcal{F}_h^\Sigma} \eta_\Sigma \int_\Omega \mathbf{r}_\Sigma^k([\![u_h]\!]) \cdot \mathbf{r}_\Sigma^k([\![v_h]\!]) = \sum_{\Sigma \in \mathcal{F}_h^\Sigma} \eta_\Sigma \int_\Sigma \{\mathbf{r}_\Sigma^k([\![u_h]\!])\} \cdot \mathbf{n}_\Sigma [\![v_h]\!] \\ &= \sum_{\Sigma \in \mathcal{F}_h^\Sigma} \eta_\Sigma \int_\Sigma \{\sum_{\sigma \in \sigma_h^\Sigma} \mathbf{r}_\sigma^k([\![u_h]\!])\} \cdot \mathbf{n}_\Sigma [\![v_h]\!] = \sum_{\Sigma \in \mathcal{F}_h^\Sigma} \sum_{\sigma \in \sigma_h^\Sigma} \eta_\Sigma \int_\Sigma \{\mathbf{r}_\sigma^k([\![u_h]\!])\} \cdot \mathbf{n}_\Sigma [\![v_h]\!] \\ &= \sum_{\sigma \in \mathcal{F}_h^\sigma} \eta_\Sigma \int_\Sigma \{\mathbf{r}_\sigma^k([\![u_h]\!])\} \cdot \mathbf{n}_\Sigma [\![v_h]\!], \end{aligned} \tag{19}$$

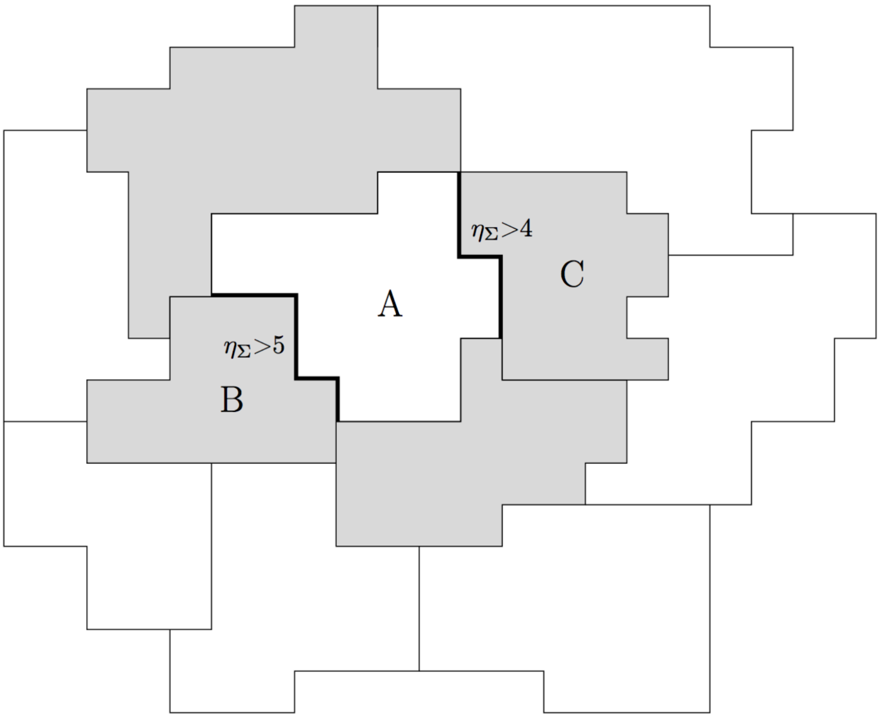
where for all $\sigma \in \mathcal{F}_h^\sigma$ we have denoted by $\Sigma \in \mathcal{F}_h^\Sigma$ the unique mesh face such that $\sigma \subset \Sigma$. Comparing the right-hand side of Eq. (18) and the last line of Eq. (19) we notice that the integral inside the sum over facets is over a facet in the former case and over a mesh face in the latter. Hence, the larger value of the stabilization parameter in the former case compensates for the smaller contribution to the stabilization term of the integral over facets with respect to the integral over mesh faces. The numerical results shown in Section §3.4 indicate that the two choices provide very similar results and thus both forms of the stabilization term can be used with confidence. Choosing one of the two forms of the stabilization parameter is just a matter of implementation convenience.

3.4. Numerical examples

Approximation properties. We now consider polygonal grids obtained by means of an agglomeration process performed on a very fine 200 × 200 uniform quadrilateral grid of a square $[-1, 1]^2$. We use the library MGridGen [15] as if we were to obtain a sequence of coarse meshes for a multigrid algorithm and we tune the agglomeration parameters so to recover a sequence of h-refined grid of 64, 255, 1028, and 4122

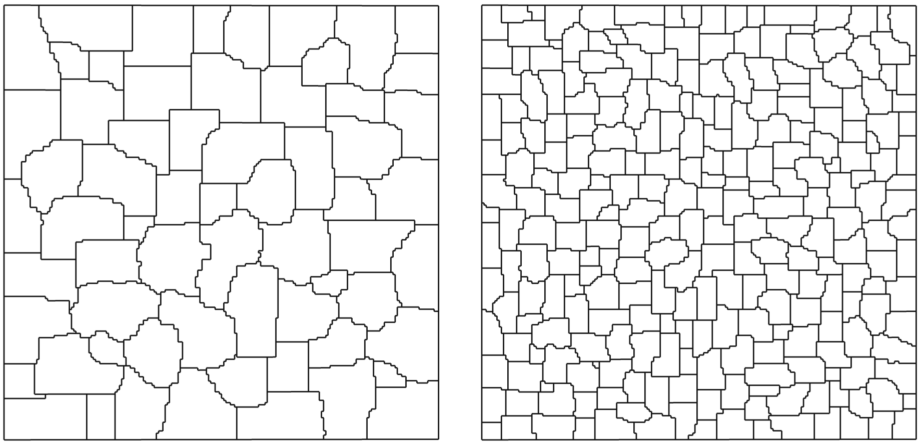


(a) Facets and values of η_σ



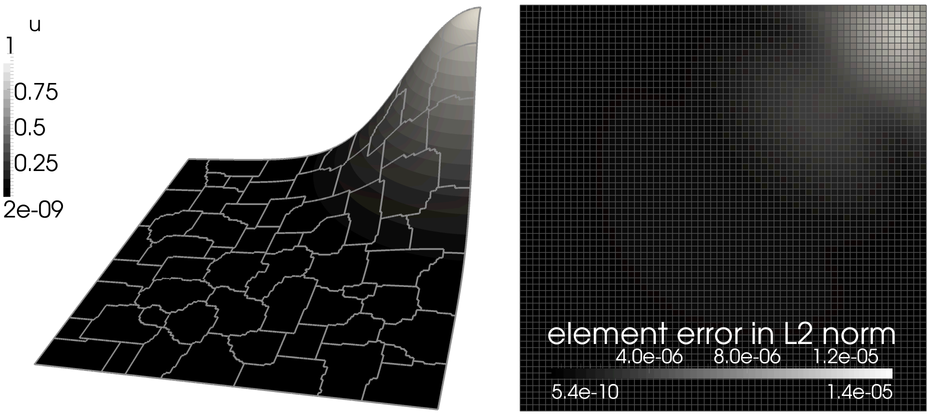
(b) Mesh faces and values of η_Σ

Figure 5: Example of polygonal mesh with mesh faces and facets defined according to Definition (1) and (2) respectively. The neighbors of the element A are shaded.



(a) 64 elements (b) 255 elements

Figure 6: Polyhedral meshes obtained by agglomeration of a 200 × 200 uniform quadrilateral grid.



(a) $k = 6$, 64 elements (b) Error distribution, 4096 elements Cartesian grid

Figure 7: BR2 solution for the test case of Karniadakis and Sherwin [12].

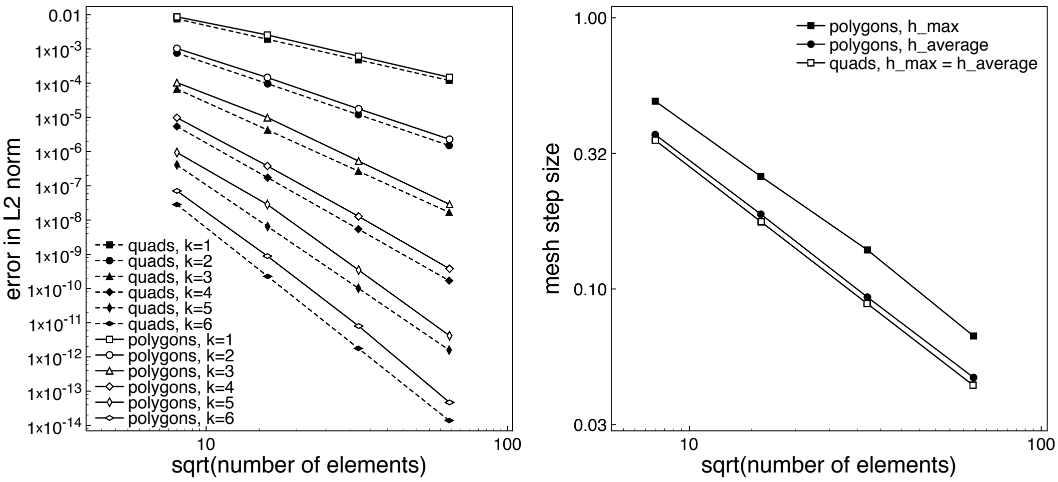


Figure 8: *Left.* Convergence of the L^2-orthogonal projection on uniform quadrilateral grids (dashed lines) and polygonal grids obtained by agglomeration (solid lines), $k \in \{1, \ldots, 6\}$. *Right.* Maximum and average mesh size h on uniform grids and polygonal grids.

polygonal elements, see Figure 6. This behavior tries to replicate the 4-fold mesh density increase typical of h-refined uniform quadrilateral grids.

The polygonal grids are used to assess the h-convergence rates of the L^2-orthogonal projection and of the BR2 dG discretization introduced in §3.2 and to compare the errors in L^2-norm with these obtained on a sequence of uniform quadrilateral grids. In all the tests here performed orthonormal physical space basis function are obtained by means of the MGS procedure outlined in §2.2. We use the exact solution to problem (7) proposed by Karniadakis and Sherwin [12],

$$u = e^{-2.5[(x-1)^2+(y-1)^2]}. \tag{20}$$

The solution behavior for a $k = 6$ approximation over the 64 elements grid of Figure 6 and the error distribution over a fine quadrilateral elements grid can be appreciated in Figure 7. The results are summarized in Figures 8 and 9. Although the error computed on uniform quadrilateral grids is almost invariably smaller than the error computed in the case of polygonal grids, the theoretical convergence rates are maintained. The better approximation properties of uniform grids are well known, and this result was somehow to be expected. We remark, however, that the possibility to devise dG discretizations on coarse grids obtained by agglomeration is remarkable in the context of solution strategies based on multigrid algorithms. Moreover, in the context of adaptive mesh refinement, the ability to locally modify the mesh size is of primary importance, as we discuss in the following section.

Decoupling geometry discretization from solution approximation. Former papers, [2, 14], have emphasized the importance of an accurate geometrical approximation of curved boundaries in dG methods, and it is now generally accepted that curved boundaries representations should be accurate enough so as not to impair the accuracy of high-degree solution approximations.

We consider the general case where $\mathcal{T}_h$ is a tessellation of an approximation Ω_h of the domain Ω, rather than of Ω itself. More specifically, letting $\partial\Omega_h \overset{\text{def}}{=}$

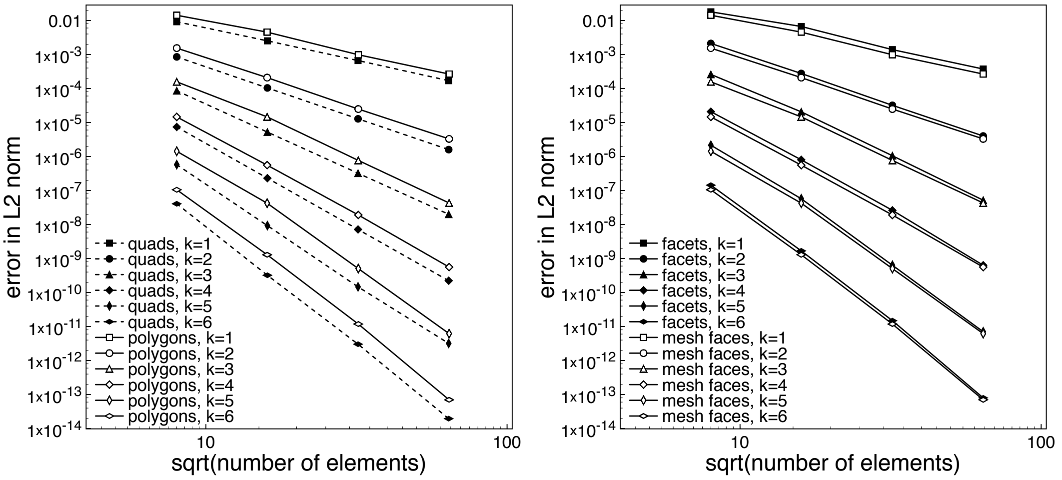


Figure 9: BR2 solution for the test case of Karniadakis and Sherwin [12]. *Left.* Convergence rates on uniform quadrilateral grids (dashed lines) and polygonal grids obtained by agglomeration (solid lines). *Right.* Convergence rates on polygonal grids obtained by agglomeration when mesh faces or, alternatively, facets are used for penalization.

$\bigcup_{F\in\mathcal{F}_h^{\mathrm{b}}} F$, there holds $\partial\Omega_h \neq \partial\Omega$. In this case, setting the boundary conditions of $\partial\Omega$ on $\partial\Omega_h$ results in a lack of (strong) consistency with respect to the unknown exact solution. In order to preserve optimal convergence properties for high-order approximations, the consistency error related to the geometrical representation of $\partial\Omega$ must not overwhelm the approximation error. While h-refinement near the boundary can be a practical solution when dealing with low-order approximations, a better approach when using high-degree polynomials on relatively coarser meshes is to improve the quality of the geometrical representation (at least) in the elements adjacent to the boundary. However, the generation of high-order meshes is by no means a trivial task, especially for highly stretched meshes like those employed in CFD problems.

Instead of directly working with increasingly higher-order approximations of curved boundaries, here we show that agglomeration suggests an alternative approach based on the geometrical approximation of the underlying fine mesh $\mathcal{R}$. In practice we propose to represent each boundary mesh face Σ as the collection of the (possibly) low-order representations of the facets composing Σ. In this framework, the boundary representation can be improved by refining $\mathcal{R}$ while keeping the same number of agglomerated elements in $\mathcal{T}_h$. In this case, there is an additional cost for numerical integration, since the cardinality of the set $\mathcal{R}_T$ for an element $T \in \mathcal{T}_h$ next to the boundary will increase, and so the number of quadrature nodes. Clearly, refining $\mathcal{R}$ will improve not only the location of $\partial\Omega_h$, but also the accuracy of the computed boundary normals and curvature. The former and the latter aspects being important in case of Dirichlet and Neumann boundary conditions, respectively.

To validate our approach we consider the test case of Gobbert and Yang [10] and possibly overturn their conclusions. To assess the k-convergence rate of the BR2 method when dealing with curved boundaries, we consider a second-order fine mesh $\mathcal{R}$ approximating the unit annulus $\Omega = \{0.5 < x^2 + y^2 < 1.5\}$ and we test against

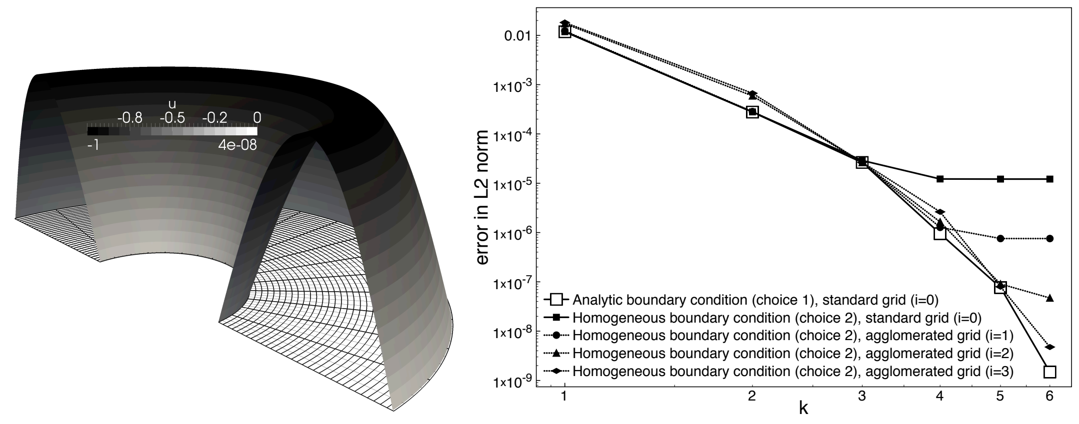


Figure 10: Test case proposed by Gobbert and Yang [10]. *Left*, Solution on a 32×32 grid (thick lines) agglomerated on top of a 128×32 quadratic quadrilateral mesh (thin lines), $k = 6$ approximation (half of the domain is sketched). *Right*, k-convergence of the BR2 discretization using a 32×32 agglomerated grid and various underlying fine meshes and boundary conditions, see text for details.

the following exact solution of the Poisson equation:

$$u = \cos(\pi\sqrt{x^2 + y^2}), \tag{21}$$

with suitable forcing term f; see Figure 10. The solution (21) vanishes on the exact boundary $\partial\Omega$. We consider the following discretizations of the homogeneous boundary condition:

$$\text{Choice 1:} \qquad u|_{\partial\Omega_h} = \cos(\pi\sqrt{x^2 + y^2}), \tag{22a}$$

$$\text{Choice 2:} \qquad u|_{\partial\Omega_h} = 0. \tag{22b}$$

The choice (22a) neglects any influence of the domain discretization as we are using the exact solution on $\partial\Omega_h$; instead, the choice (22b) introduces a consistency error since, in our case, $\partial\Omega_h \neq \partial\Omega$. For $i \in \{0, \ldots, 3\}$, we construct an agglomerated mesh $\mathcal{T}_{h,i}$ composed of 32×32 elements starting from a fine mesh $\mathcal{R}_i$ composed of $(32 \cdot 2^i) \times 32$ eight-nodes quadrilateral elements (circumferential × radial number of elements). In this case, for all $i \in \{0, \ldots, 3\}$ and all $T \in \mathcal{T}_{h,i}$, $\text{card}(\mathcal{R}_T) = 2^i$. As a result, the number of agglomerated elements does not change, but $\partial\Omega_h$ approximates $\partial\Omega$ more closely by increasing i. The convergence results shown in Figure 10 demonstrate the ability of the boundary condition in (22b) (which is the one used in practice) to provide, increasing i, convergence results closer to the exponential convergence resulting from choice (22a). Instead, the lack of consistency for $i = 0$ is clearly appreciable as the discretization is unable to provide an error in the L^2-norm lower that 10^{-5} using the boundary condition in (22b).

Application of h-adaptivity to the Poisson equation. We present here some convergence results for the BR2 method outlined in §3.2 using the exact solution (20) and performing agglomeration-based adaptive mesh refinement (AMR). We considered here approximations based on the space of piecewise affine functions, i.e. we let $V_h = \mathbb{P}^1_d(\mathcal{T}_h)$. Thanks to the possibility to define discrete polynomial spaces over agglomerated cells the grid can be easily adapted to the solution behavior. To this end, we define for each element $E \in \mathcal{R}$ an indication of the agglomeration rate $\overline{\text{card}}(\mathcal{R}_T)$ representing a strict upper bound for the number of sub-elements in an aggregate element. The resulting grid is then such that, for all $T \in \mathcal{T}_h$, there holds

$$\text{card}\,(\mathcal{R}_T) \leqslant \overline{\text{card}}(\mathcal{R}_T).$$

Starting from a constant value over $\mathcal{R}$, the agglomeration rate is modified during the refinement process according to the local error in the L^2-norm with respect to the analytical solution. Clearly, since the minimum admissible value of the agglomeration rate is one, the size of a refined element is bounded from below by the local element size of the underling grid.

We let

$$\begin{aligned} &\forall T \in \mathcal{T}_h, && e_T \overset{\text{def}}{=} \|u - u_h\|_{L^2(T)} \\ &\forall T \in \mathcal{T}_h,\ \forall E \in \mathcal{R}_T, && e_E \overset{\text{def}}{=} \|u - u_h\|_{L^2(E)} + e_T. \end{aligned}$$

The definition of e_E is intended to increase the probability that the element $E \in \mathcal{R}$ with the largest error lies inside the aggregate element $T \in \mathcal{T}_h$ with the largest error. The actual error metric is obtained by normalizing the above quantities, i.e. we let

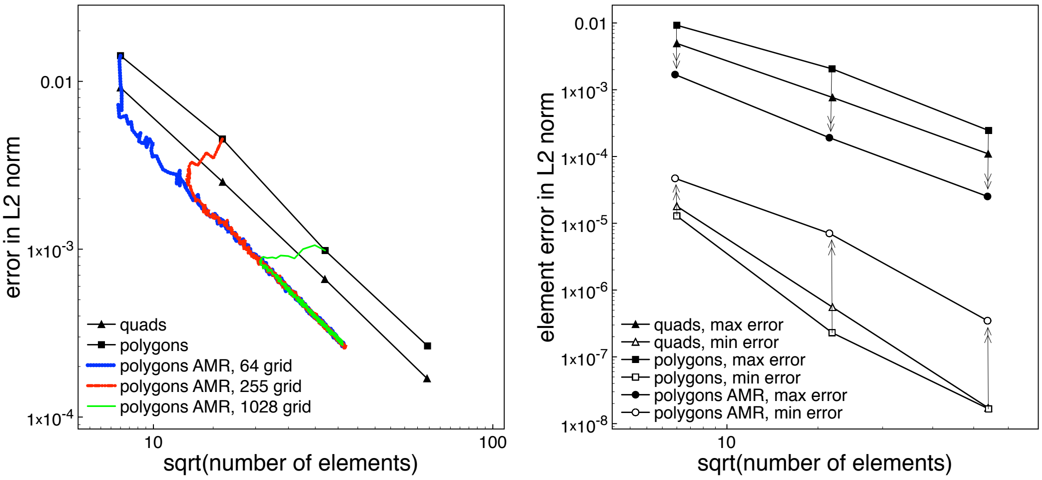
$$\begin{aligned} &\forall E \in \mathcal{R}, && \epsilon_E = \frac{e_E - \min_{E \in \mathcal{R}} e_E}{\max_{E \in \mathcal{R}} e_E - \min_{E \in \mathcal{R}} e_E}, \\ &\forall T \in \mathcal{T}_h, && \epsilon_T = \frac{e_T - \min_{T \in \mathcal{T}_h} e_T}{\max_{T \in \mathcal{T}_h} e_T - \min_{T \in \mathcal{T}_h} e_T}, \end{aligned}$$

so that ϵ_E and ϵ_T are both contained in the interval $[0, 1]$. Two different adaptation strategies borrowed from the open source library `libMesh` [13] are considered. The first one is best suited to optimize the discretization error while maintaining fixed computation requirements in terms of memory and execution time, while the second one can be used in place of global h-refinement in order to perform convergence studies and evaluate the discretization accuracy.

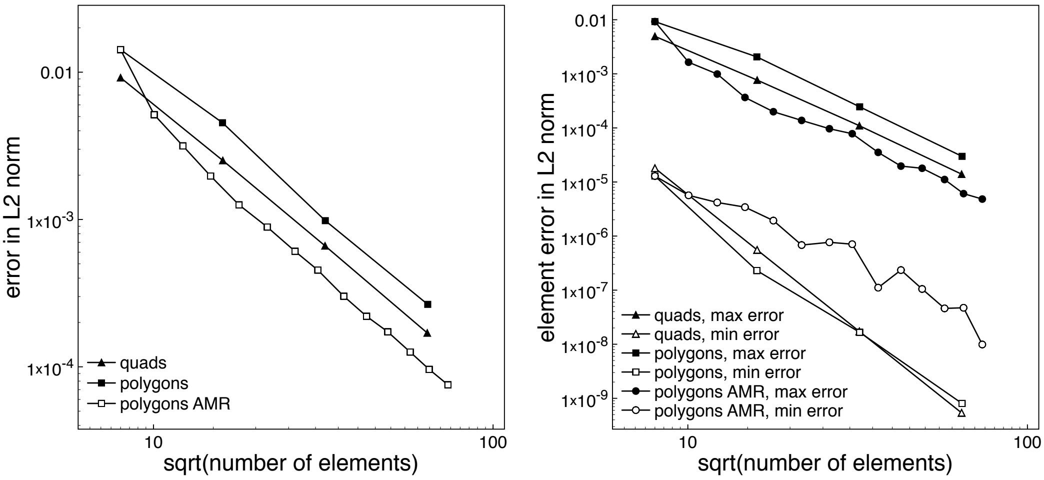
Example 2 (First strategy)**.** *Refinement and coarsening instructions are identified comparing the normalized cell errors with user defined parameters, say* f_{ref} *and* f_{crs}*, corresponding to the relative errors triggering refinement and coarsening respectively. At each adaptation the agglomeration rates* $\overline{\text{card}}(\mathcal{R}_T)$*,* $T \in \mathcal{T}_h$*, are modified according to the following procedure:*

```
{Refinement}
for $T \in \mathcal{T}_h$ do
    if $\epsilon_T \geqslant (1 - f_{\text{ref}})$ then
        for $E \in \mathcal{R}_T$ do
            if $\epsilon_E \geqslant (1 - f_{\text{ref}})$ then
                $\overline{\text{card}}(\mathcal{R}_T) = \frac{\text{card}(\mathcal{R}_T)}{2}$
            end if
        end for
    end if
end for
```

```
{Coarsening}
for $T \in \mathcal{T}_h$ do
    if $\epsilon_T \leqslant f_{\text{crs}}$ then
        for $E \in \mathcal{R}_T$ do
            if $\epsilon_E \leqslant f_{\text{crs}}$ then
                if $\overline{\text{card}}(\mathcal{R}_T)/\text{card}(\mathcal{R}_T) \leqslant 2$ then
                    $\overline{\text{card}}(\mathcal{R}_T) = 2\,\text{card}(\mathcal{R}_T)$
                end if
            end if
        end for
    end if
end for
```

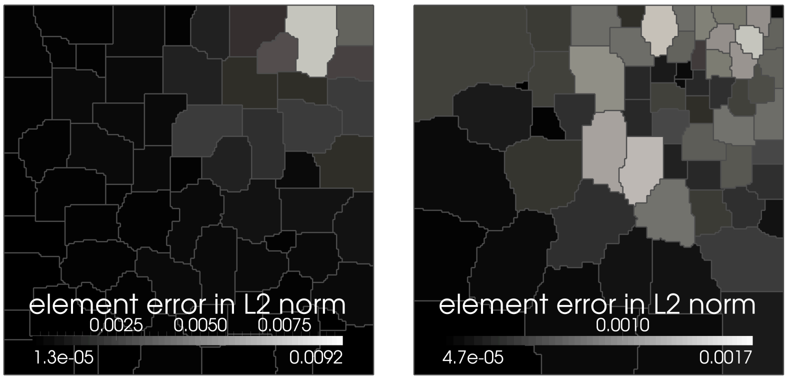


(a) Strategy of Example 2

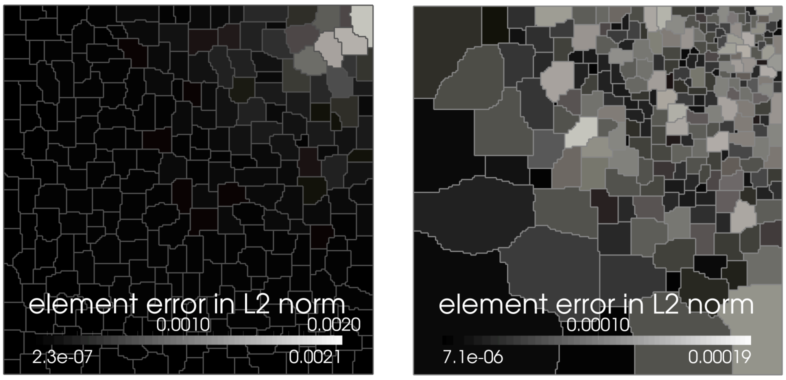


(b) Strategy of Example 3

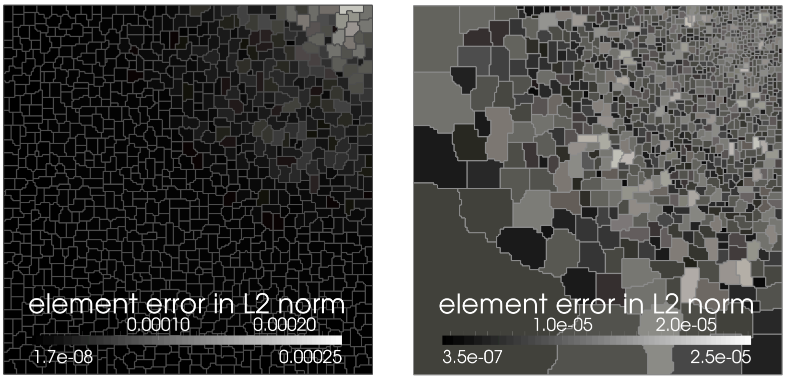
Figure 11: BR2 solution for the test case of Karniadakis and Sherwin [12], $k = 1$. Comparison of the convergence rates using uniform Cartesian meshes, uniform polygonal meshes, and adaptively refined polygonal meshes. *Top row.* Three distinct adaptive processes are considered starting from 64, 255, and 1028 polygonal elements. *Bottom row.* Single adaptive process starting from 64 polygonal elements. *Left.* Error in L^2-norm. *Right.* Maximum and minimum error in L^2-norm computed over the mesh elements. The arrows indicate the action of AMR from starting polygonal grids to adapted grids with the same card($\mathcal{T}_h$).



(a) card($\mathcal{T}_h$) = 64, uniform (b) card($\mathcal{T}_h$) = 63, adaptive

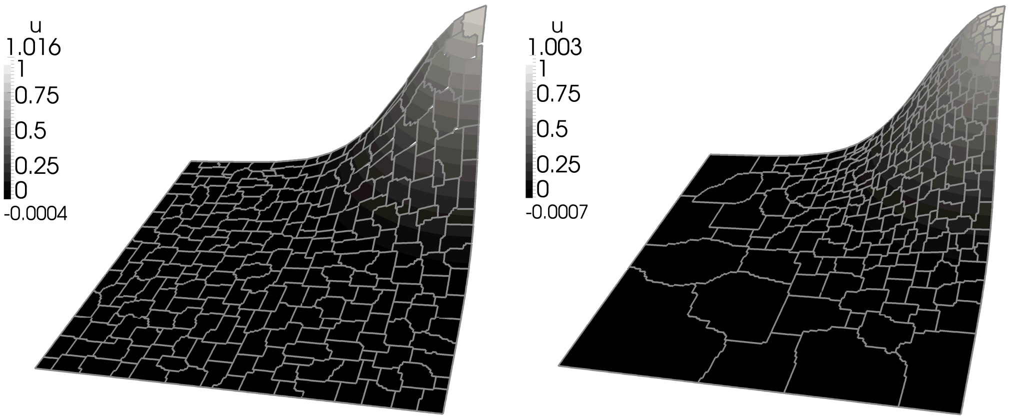


(c) card($\mathcal{T}_h$) = 255, uniform (d) card($\mathcal{T}_h$) = 249, adaptive



(e) card($\mathcal{T}_h$) = 1028, uniform (f) card($\mathcal{T}_h$) = 1018, adaptive

Figure 12: Polygonal grids on top of a 200×200 uniform quadrilateral grid, first degree polynomial approximation and distribution of the error in the L^2-norm



(a) $k = 1$, 255 elements, uniform (b) $k = 1$, 249 elements, adaptive

Figure 13: BR2 solution for the test case of Karniadakis and Sherwin [12].

In the coarsening step, line 6 accounts for the potential inability of the agglomeration process to saturate the agglomeration rate, possibly resulting in $\mathrm{card}(\mathcal{R}_T) \ll \overline{\mathrm{card}}(\mathcal{R}_T)$*.*

Example 3 (Second strategy)**.** *The number of elements to be refined is chosen as a fraction* f_{ref} *of* $\mathrm{card}(\mathcal{T}_h)$*. The elements are sorted in increasing order according to* ϵ_T *and numbered starting from zero to reflect the element position* Pos_T *in the sorted elements vector. At each iteration of the adaptation strategy refinement is obtained modifying the agglomeration rates* $\overline{\mathrm{card}}(\mathcal{R}_T)$*,* $T \in \mathcal{T}_h$*, as follows:*

```
for $T \in \mathcal{T}_h$ do
    if $\mathrm{Pos}_T \geqslant (1 - f_{\mathrm{ref}})\,\mathrm{card}(\mathcal{T}_h)$ then
        $\overline{\mathrm{card}}(\mathcal{R}_T) = \frac{\mathrm{card}(\mathcal{R}_T)}{2}$
    end if
end for
```

It is interesting to notice that, in both cases, coarsening and refinement act on the real number of elements, $\mathrm{card}(\mathcal{R}_T)$*, of the starting mesh* $\mathcal{R}$ *that concur in forming an element of* $\mathcal{T}_h$ *and not on the agglomeration rate* $\overline{\mathrm{card}}(\mathcal{R}_T)$*.*

In practice, the above strategies can be applied by replacing the quantities ϵ_T, $T \in \mathcal{T}_h$, and ϵ_E, $E \in \mathcal{R}$, by *a posteriori* error estimates.

The results obtained applying the strategy of Example 2 with $f_{\mathrm{ref}} = 0.1$ and $f_{\mathrm{crs}} = 0.15$ to the 64, 255 and 1028 polygonal elements grids in Figure 6 are collected in Figure 11(a). It is interesting to note that both $\mathrm{card}(\mathcal{T}_h)$ and the L^2-norm of the error decrease in the first place thanks to the combined action of coarsening and refinement; then, once the error over the domain is redistributed, the AMR process maintain a second order convergence rate. Independently from the initial mesh, the strategy is able to guide the discretization towards analogue error values as the lines associated to different starting grids are perfectly overlapping once the same $\mathrm{card}(\mathcal{T}_h)$ is reached. The global L^2-norm of the error is always smaller on the grids resulting from adaptive h-refinement. On the right panel of Figure 11(a) we sketch the values of $\max_{T \in \mathcal{T}_h} e_T$ and $\min_{T \in \mathcal{T}_h} e_T$ that would have been obtained arresting the refinement process as soon as $\mathrm{card}(\mathcal{T}_h)$ is comparable with the number of elements in the starting mesh. The resulting h-adapted grids as well as the polygonal grids considered as starting point are shown in Figure 12. A comparison between uniform and adaptive meshes featuring the same number of elements shows the difference in the maximum and minimum error values, their location as well as the better distribution of the error over the domain in the adaptive case. The solution behavior over the uniform 255 and the adapted 249 agglomerated elements grids of Figure 12 can be compared in Figure 13.

The results obtained applying the strategy of Example 3 with $f_{\text{ref}} = 0.17$ to the 64 polygonal element grid in Figure 6 are reported in Figure 11(b). The number of elements increases with a constant ratio allowing to quickly increase the discretization accuracy. Up to $\text{card}(\mathcal{T}_h) = 300$, in the error redistribution phase, the second order discretization shows a third order convergence rate and, afterwards, second order convergence is maintained. Although the error in L^2-norm as well as the maximum element error are higher than the ones obtained with strategy of Example 2, the benefits of h-adaptivity are still present in this raw and fast refinement algorithm.

4. Reduced quadratures on general meshes

4.1. The reduction algorithm

Integration on general shape elements is an open field of research and, to our knowledge, only few papers have been published on this topic. The recent works of Mousavi, Xiao and Sukumar [17] and Xiao and Gimbutas [20], propose an interesting technique and certainly deserves more attention. Study and implementation of this approach will be left for future work. In the following we describe the simple approach that has been employed for numerical integration on the agglomerated elements considered in this work. This procedure consists of (i) computation of integrals on agglomerated elements as the sum of integrals on the composing sub-elements, (ii) integration on each sub-element performed by means of Gaussian quadrature defined on reference space elements. This procedure can be very expensive and a simple strategy to reduce the integration cost will be presented hereafter.

The integral of any polynomial function $v \in \mathbb{P}^k_d(\mathcal{T}_h)$ over an element $T \in \mathcal{T}_h$ can be computed as follows:

$$\int_T v(\mathbf{x})\, \mathrm{d}\mathbf{x} = \sum_{E \in \mathcal{R}_T} \int_E v(\mathbf{x})\, \mathrm{d}\mathbf{x} = \sum_{E \in \mathcal{R}_T,\, E = \Psi_E(E^{\text{ref}})} \int_{E^{\text{ref}}} (v \circ \Psi_E)(\boldsymbol{\xi}) |J_{\Psi_E}(\boldsymbol{\xi})|\, \mathrm{d}\boldsymbol{\xi}, \quad (23)$$

where $\mathbf{x}$ and $\boldsymbol{\xi}$ are physical and reference space coordinates respectively, and J_{Ψ_E} is the Jacobian of the mapping function Ψ_E. From (23) it is clear that the polynomial degree q of the integrand in the last term of (23) results from the product of the polynomial degrees of v and Ψ_E, respectively, plus the polynomial degree j of $|J_{\Psi_E}|$ according to the equation

$$q = km + j. \quad (24)$$

The value of q rapidly increases when considering high order polynomials on curved elements, and so does the number of quadrature nodes required to compute the integral exactly. However, the numerical experiments of this section indicate that exact integration is only required for very stretched and curved elements. In other, more common, situations it is possible to reduce the exactness of the quadrature rule without compromising the accuracy. In order to make sure that a desired precision is achieved, we introduce the following measure for the integration error on the element E:

$$\forall T \in \mathcal{T}_h, \forall E \in \mathcal{R}_T, \qquad \epsilon_{\mathrm{i},E} \overset{\mathrm{def}}{=} |m^*_{ii} - m^{ex}_{ii}|$$

where m^*_{ii} denotes the value of the ith diagonal entry of the local mass matrix computed with the reduced quadrature rule, whereas m^{ex}_{ii} its the expected value as a result of exact integration. Then we use on the elements of $\mathcal{R}_T$ an integration rule with the minimum degree of exactness needed to satisfy the condition:

$$\max_{i \in D_T} \epsilon_{\mathrm{i},E} \leqslant \mathrm{tol}, \qquad \forall T \in \mathcal{T}_h, \tag{25}$$

where tol is the maximum error accepted in the diagonal entries of the mass matrix. As will be pointed out in §4.2, the reduction algorithm can also benefit from the imposition of a minimum admissible degree of exactness defined as a function of the degree of the polynomial expansion to be integrated.

The effectiveness of the above procedure is assessed in a parametric study on a conventional element. We consider the set of basis functions for the $\mathbb{P}^{10}_d(T)$ space on the quadrangle T whose sides are defined by cubic polynomials. According to (24), the exact integration of the terms in the mass matrix requires a degree of exactness of $2 \times 10 \times 3 + 5 = 65$ for a tensor product quadrature rule. However, Figure 14 shows that exact integration is not required. Even setting tol $= 10^{-10}$ the degree of exactness can be significantly reduced also for highly stretched or curved elements.

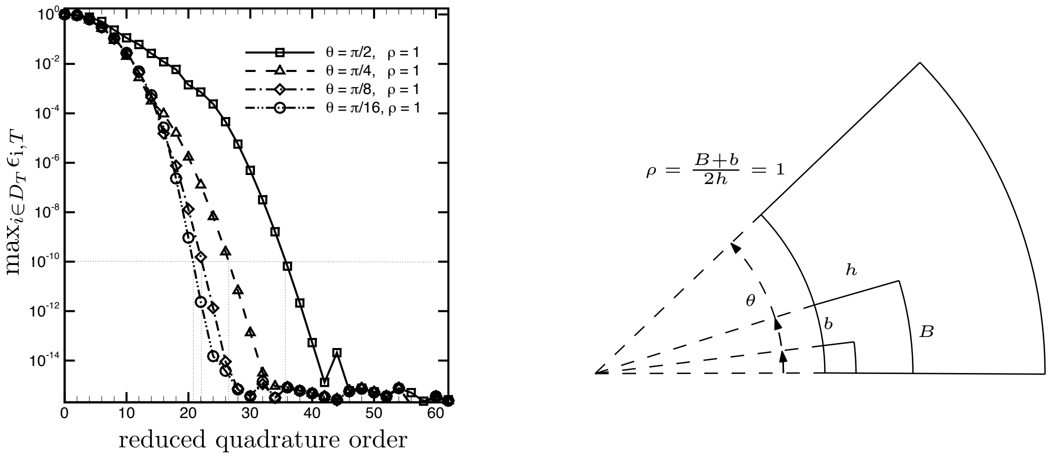
4.2. Numerical examples

To numerically assess the benefits of using the reduced quadrature procedure introduced in §4.1 we consider the dG discretization of the Poisson equation (13) with Dirichlet boundary conditions and the forcing term set according to the analytical solution of Equation (20). The performance and the accuracy obtained with exact and reduced numerical integration are compared using a 255 polygonal elements grid built on top of a fine 40000 quadrilateral elements mesh, see Figure 6. A standard grid with 256 quadrilateral elements is used as a baseline to evaluate the efficiency of the BR2 discretization on agglomerated grids. Polynomial expansions up to $k = 6$ are considered.

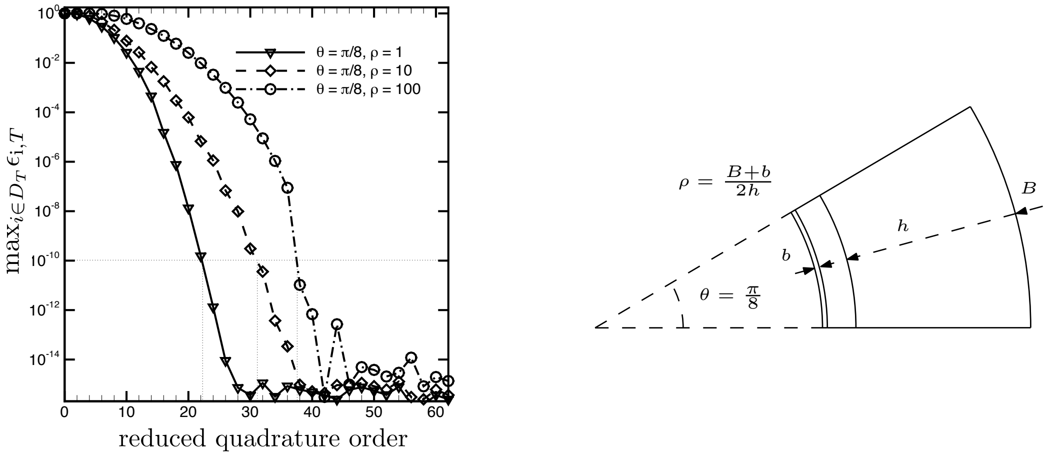
In order to derive an efficient implementation of the BR2 discretization we apply the global and local lifting operators definitions to the bilinear form in (11) to obtain the following equivalent form

$$a_h(u_h, v_h) = \int_\Omega \nabla_h u_h \cdot \nabla_h v_h - \sum_{F \in \mathcal{F}_h} \int_F \left(\{\nabla_h u_h\} \cdot \mathbf{n}_F \, [\![v_h]\!] + [\![u_h]\!] \, \{\nabla_h v_h\} \cdot \mathbf{n}_F \right) \\ + \sum_{F \in \mathcal{F}_h} \eta_F \int_F \{\mathbf{r}^k_F([\![u_h]\!])\} \cdot \mathbf{n}_F [\![v_h]\!]. \tag{26}$$

Despite the same number of terms to be integrated, this alternative formulation reduces the numerical integration cost since Gaussian quadrature formulas on faces require less quadrature points compared with element integrals. Moreover, higher



(a) Varying element curvature at constant aspect ratio



(b) Varying element aspect ratio at constant curvature

Figure 14: Integration error vs. quadrature order.

gains are expected on agglomerated grids as the average number of facets composing the mesh faces can be considerably smaller than the number of sub-elements composing the polygons. For example, the mean number of sub-elements composing the polygons of the 255 elements grid here considered is 156 while the average number of facets composing the mesh faces is 14. Therefore, while exact integration is roughly 156 times more expensive on $T \in \mathcal{T}_h$ than on a standard mesh element, exact integration on $F \in \mathcal{F}_h$ (computed as the sum of integrals over each facet) turns out to be 14 times more expensive than integration on a standard mesh face. Since three of the four terms in (26) need to be integrated over mesh faces it is useful and straightforward to extend the quadrature reduction procedure introduced in §4.1 to face integrals. To this end the degree of exactness computed according to (25) is used not only for quadrature formulas on E but also for face integrals on $\sigma = \Sigma \cap \partial E$.

Poisson Eq. dG discretization, Exact and Reduced numerical integration							
	Ex. Int.	Red. Int., adaptive *tol*			Red. Int., fixed *tol*		
k	$\|u - u_h\|_{L^2(\Omega)}$	*tol*	min(*doe*)	$\|u - u_h\|_{L^2(\Omega)}$	*tol*	min(*doe*)	$\|u - u_h\|_{L^2(\Omega)}$
1	$4.53624e{-3}$	10^{-1}	1	$4.53089e{-3}$	10^{-1}	1	$4.53089e{-3}$
2	$2.08739e{-4}$	10^{-2}	1	$2.08738e{-4}$	10^{-1}	2	$2.08738e{-4}$
3	$1.46011e{-5}$	10^{-2}	1	$1.46011e{-5}$	10^{-1}	3	$1.46011e{-5}$
4	$5.57018e{-7}$	10^{-3}	1	$5.57017e{-7}$	10^{-1}	4	$5.57017e{-7}$
5	$4.23520e{-8}$	10^{-4}	1	$4.23519e{-8}$	10^{-1}	5	$4.23515e{-8}$
6	$1.28866e{-9}$	10^{-4}	1	$1.28876e{-9}$	10^{-1}	6	$1.28866e{-9}$

Table 1: Reduced numerical integration. BR2 discretization of the Poisson equation, see §3.2, on the 255 polygonal elements mesh obtained on top of a 40000 quadrilateral elements mesh, see Figure 6. *Adaptive tol*, the tolerance on the mass matrix integration error is defined according to the exact error in L^2 norm, see §4.1 for details. *Fixed tol*, the tolerance is fixed and the minimum degree of exactness (*doe*) of the quadrature formula is set to k. $\|u - u_h\|_{L^2(\Omega)}$ is computed with exact numerical integration.

Two different reduction strategies are considered. First the tolerance on the mass matrix error, see Definition (25), is varied according to the exact error in the L^2-norm (the order of magnitude of tol is computed as the smaller integer number obtained halving the order of magnitude of the error in L^2-norm), see Table 1. Second we select a fixed tolerance tol $= 10^{-1}$ and limit the minimum degree of exactness of the quadrature rule to the order k of the polynomial expansion. The former strategy provides the best reduction capabilities while the latter is best suited to be applied in real-life computations. In this context the tolerance must be chosen according to the accuracy prescribed for the problem at hand as the accuracy associated to the discretization is not known *a priori*. The limit on the minimum degree of exactness is safely chosen to avoid undue under integration and induces a speed up of the reduction procedure as the number of quadrature rules to be evaluated decreases. For both strategies considered here the difference in the error in the L^2-norm is negligible and does not affect the theoretical convergence of the scheme, see Table 1.

The results reported in Table 2, show that the reduction in the number of quadrature points is larger when high-order expansions are considered. In this case a larger

Exact and Reduced numerical integration, number of quadrature points								
	Red. Int., adaptive *tol*				Red. Int., fixed *tol*			
k	fq^R/fq^E	eq^R/eq^E	fq^R/fq^{std}	eq^R/eq^{std}	fq^R/fq^E	eq^R/eq^E	fq^R/fq^E	eq^R/eq^{std}
1	0.63	0.47	8.51	72.0	0.63	0.46	8.51	72.0
2	0.67	0.46	9.05	69.7	0.67	0.44	9.05	69.4
3	0.55	0.29	7.48	42.9	0.50	0.25	6.79	39.1
4	0.60	0.36	8.15	55.0	0.60	0.36	8.15	56.2
5	0.66	0.35	9.00	53.7	0.50	0.25	6.79	39.1
6	0.58	0.30	7.91	46.2	0.57	0.33	7.76	51.0

Table 2: Reduced numerical integration, see Table 1 for details. *From left to right, in each block.* Ratio between the number of quadrature points required for reduced and exact numerical integration on the 255 polygonal elements grid in Figure 6, face quadrature points (fq^R/fq^E) and element quadrature points (eq^R/eq^E) ratio. Ratio between the number of quadrature points required for reduced numerical integration on the 255 polygonal elements grid and exact numerical integration on a standard 256 quadrilateral elements grid, face quadrature points (fq^R/fq^{std}) and volume quadrature points (eq^R/eq^{std}) ratio.

Exact and Reduced integration, matrix assembly, shape function evaluation								
	Red. Int., adaptive *tol*				Red. Int., fixed *tol*			
k	t^R_{se}/t^E_{se}	t^R_{ma}/t^E_{ma}	t^R_{se}/t^{std}_{se}	t^R_{ma}/t^{std}_{ma}	t^R_{se}/t^E_{se}	t^R_{ma}/t^E_{ma}	t^R_{se}/t^{std}_{se}	t^R_{ma}/t^{std}_{ma}
1	0.63	0.52	36.8	10.9	0.63	0.52	36.9	10.9
2	0.72	0.51	49.4	14.2	0.71	0.50	49.1	14.1
3	0.62	0.33	53.4	11.1	0.59	0.30	51.2	10.1
4	0.65	0.39	63.6	16.1	0.65	0.40	64.0	16.3
5	0.53	0.39	63.9	16.9	0.59	0.29	63.9	12.5
6	0.63	0.34	77.2	16.7	0.64	0.36	78.5	16.8

Table 3: Reduced numerical integration, see Table 1 for details. *From left to right, in each block.* Ratio between the CPU time required by reduced and exact quadrature formulas for shape function evaluation (t^R_{se}/t^E_{se}) and matrix assembly (t^R_{ma}/t^E_{ma}) on the 255 polygonal elements grid in Figure 6. Ratio between the CPU time required by reduced quadrature formulas on the 255 polygonal elements grid and exact quadrature formulas on a standard 256 quadrilateral elements grid, shape function evaluation (t^R_{se}/t^{std}_{se}) and matrix assembly (t^R_{ma}/t^{std}_{ma}).

set of quadrature formulas can be considered in order to match the prescribed tolerance on the mass matrix integration error. Moreover, while a first degree expansion is linear all over an element $T \in \mathcal{T}_h$, the degree of exactness required to integrate higher order expansions might vary locally inside each sub-element composing T. We remark that the quadrature reduction procedure reduces the number of quadrature points by a factor of 3 for mesh elements and by a factor of 2 for mesh faces when high-order discretizations are considered. Compared with a standard non agglomerated grid with the same number of mesh elements, the 255 polygonal elements grid requires roughly 8 times the number of face quadrature points and, in general, more than 50 times the number of element quadrature points. However, as can be appreciated in Table 3, the performance penalty in terms of matrix assembly time is much closer to 8 than to 50 as for each mesh element we need to compute one element integral and three integrals for each element face. Overall, on the test case here considered, the performance penalty in the matrix assembly time amounts to about 10% of the average number of sub-elements in the mesh elements.

Similar considerations do not apply to the cost of shape function evaluation because the orthonormalization coefficients r_{ii}^T and r_{ij}^T must be computed using exact integration, see Table 3. However this can be considered less important as usually, in non-linear and/or time dependent computations, the shape functions are computed once and stored as long as the mesh does not change.

Even if not implemented in this work the algorithms proposed by Mousavi, Xiao and Sukumar [17] and Xiao and Gimbutas [20], should provide quadrature formulas for polygons having a computational cost comparable to that of integration on standard elements. However, since in general it is not trivial to use the same procedure to reduce the cost of face integrals, the performance of the discretization might be limited by numerical integration over facets. In this case, if the cost of integration over mesh elements becomes comparable or cheaper than the cost of integration over mesh faces, the bilinear form (11) might provide a more efficient implementation than the form (26).

On account of the above considerations, we remark that the quadrature efficiency is still an open issue that needs further investigation in order to fruitfully exploit the flexibility of the agglomeration based dG discretizations here proposed.

Acknowledgments

Lorenzo Botti, Alessandro Colombo acknowledge the financial support of the European Union, under the ADIGMA project. Daniele Di Pietro acknowledges the support of the VFSitCom ANR project. The authors gratefully acknowledge reviewers' comments and suggestions that helped to improve the quality of the manuscript.

[1] Bassi, F., Crivellini, A., Di Pietro, D. A., and Rebay, S. An artificial compressibility flux for the discontinuous Galerkin solution of the incompressible Navier-Stokes equations. *J. Comput. Phys. 218* (2006), 794–815.

[2] Bassi, F., and Rebay, S. High-order accurate discontinuous finite element solution of the 2D Euler equations. *J. Comput. Phys. 138* (1997), 251–285.

[3] Bassi, F., Rebay, S., Mariotti, G., Pedinotti, S., and Savini, M. A high-order accurate discontinuous finite element method for inviscid and viscous turbomachinery flows. In *Proceedings of the 2nd European Conference on Turbomachinery Fluid Dynamics and Thermodynamics* (Antwerpen, Belgium, March 5–7 1997), R. Decuypere and G. Dibelius, Eds., Technologisch Instituut, pp. 99–108.

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

[5] Di Pietro, D. A., and Ern, A. Discrete functional analysis tools for discontinuous Galerkin methods with application to the incompressible Navier-Stokes equations. *Math. Comp.* (2010), 1303–1330.

[6] Di Pietro, D. A., and Ern, A. *Mathematical Aspects of Discontinuous Galerkin Methods*. Maths & Applications. Springer-Verlag, 2011. In press.

[7] Dolejší, V., and Felcman, J. Anisotropic mesh adaptation for numerical solution of boundary value problems. *Numerical Methods for Partial Differential Equations 20* (2004), 576–608.

[8] Gassner, G. J., Lörcher, F., Munz, C.-D., and Hesthaven, J. S. Polymorphic nodal elements and their application in discontinuous Galerkin methods. *J. Comput. Phys. 228* (2009), 1573–1590.

[9] Giraud, L., Langou, J., and Rozloznik, M. On the loss of orthogonality in the Gram-Schmidt orthogonalization process. Technical Report No. TR/PA/03/25, CERFACS, 2003.

[10] Gobbert, K., and Yang, S. Numerical Demonstration of Finite Element Convergence for Lagrange Elements in COMSOL Multiphysics. In *Proceedings of the COMSOL Conference 2008 Boston* (2008).

[11] Hartmann, R., and Leicht, T. Error estimation and anisotropic mesh refinement for 3D laminar aerodynamic flow simulations. *J. Comput. Phys. 229* (2010), 7344–7360.

[12] Karniadakis, G. E., and Sherwin, S. *Spectral/hp Element Methods for Computational Fluid Dynamics*. Numerical Mathematics and Scietific Computation. Oxford University Press, USA, 2005.

[13] Kirk, B., Peterson, J. W., Stogner, R. H., and Carey, G. F. libmesh: A c++ library for parallel adaptive mesh refinement/coarsening simulations. *Eng. Comput. 22* (2006), 237–254.

[14] Krivodonova, L., and Berger, M. High-order accurate implementation of solid wall boundary conditions in curved geometries. *J. Comput. Phys. 211* (2006), 492–512.

[15] Moulitsas, I., and Karypis, G. MGridGen/ParmGridGen, Serial/Parallel library for generating coase meshes for multigrid methods. Technical Report Version 1.0, University of Minnesota, Department of Computer Science/Army HPC Research Center, 2001. http://www-users.cs.umn.edu/∼moulitsa/software.html.

[16] Moulitsas, I., and Karypis, G. Multilevel algorithms for generating coarse grids for multigrid methods. In *Supercomputing 2001 Conference Proceedings* (2001).

[17] Mousavi, S., Xiao, H., and Sukumar, N. Generalized Gaussian quadrature rules on arbitrary polygons. *Int. J. Numer. Meth. Engng.*, 82 (2010), 99–113.

[18] Rashid, M. M., and Selimotic, M. A three-dimensional finite element method with arbitrary polyhedral elements. *Int. J. Numer. Meth. Engng. 67* (2006), 226–252.

[19] Tesini, P. *An* h-*Multigrid Approach for High-Order Discontinuous Galerkin Methods*. PhD thesis, Università degli Studi di Bergamo, January 2008.

[20] Xiao, H., and Gimbutas, Z. A numerical algorithm for the construction of efficient quadrature rules in two and higher dimensions. *Computers & Mathematics with Applications*, 59 (2010), 663–676.