Goal–Oriented Adaptive Composite Discontinuous Galerkin Methods for Incompressible Flows

S. Giani

School of Engineering and Computing Sciences, Durham University, South Road, Durham, DH1 3LE, UK

stefano.giani@durham.ac.uk.

P. Houston

School of Mathematical Sciences, University of Nottingham, University Park, Nottingham NG7 2RD, UK

Paul.Houston@nottingham.ac.uk

Abstract

In this article we consider the application of goal–oriented mesh adaptation to problems posed on complicated domains which may contain a huge number of local geometrical features, or micro-structures. Here, we exploit the composite variant of the discontinuous Galerkin finite element method based on exploiting finite element meshes consisting of arbitrarily shaped element domains. Adaptive mesh refinement is based on constructing finite element partitions of the domain consisting of agglomerated elements which belong to different levels of an underlying hierarchical tree data structure. As an example of the application of these techniques, we consider the numerical approximation of the incompressible Navier–Stokes equations. Numerical experiments highlighting the practical performance of the proposed refinement strategy will be presented.

Keywords: Composite finite element methods, discontinuous Galerkin methods, *a posteriori* error estimation, adaptivity, incompressible flows.
2000 MSC: 65N15, 65N30, 65N50

1. Introduction

In recent years extensive work has been undertaken on the construction of finite element methods on general meshes consisting of polygonal and poly-

hedral elements. In the context of conforming finite element element methods, notable examples include the composite finite element method (CFEM) proposed in [20, 19], for example, the polygonal finite element method, cf. [25, 26], and the extended finite element method [16]. Stimulated by work on mimetic finite difference methods, cf. [8, 28], for example, the virtual element method has been proposed in [27]. Closely related techniques have been developed in the context of fictitious domain methods, whereby overlapping meshes are employed, see, for example, [9, 10, 11, 23].

Discontinuous Galerkin finite element methods (DGFEMs), cf. [14, 15, 21, 24, 3, 13], naturally admit the use of general finite element meshes, since the underlying finite element space does not require the enforcement of any inter-element continuity constraints. Moreover, DGFEMs may exploit local polynomial spaces defined in the physical frame, without the need to employ element mappings from a reference frame; thereby, the order of convergence of the underlying method is *independent* of the element shape employed, cf. [12, 15, 22, 29]. In particular, Bassi *et al.* [4, 5, 6] have considered the application of DGFEMs on general meshes consisting of general agglomerated elements. In our own work, stimulated by the articles [20, 19], we have developed the discontinuous Galerkin composite finite element method (DGCFEM) for problems posed on complicated computational domains which may contain local geometrical features, or micro-structures, see [1, 18]; for the application of these ideas to the construction of domain decomposition preconditioners, we refer to [17, 2]. The key feature of CFEMs/DGCFEMs is that they allow for the construction of coarse finite element meshes, consisting of general element shapes, which provide an accurate description of the computational domain Ω, even in the presence of micro-structures. In this manner, the minimal dimension of the underlying composite finite element space is *independent* of the number of geometric features present in Ω. Thereby, computations may be undertaken on coarse finite element meshes in order to yield numerical approximations which are sufficiently accurate in terms of attaining *engineering accuracy* in an efficient manner. More precisely, by admitting such coarse initial meshes, adaptive refinement of the computational mesh may be guided in order to resolve specific features of the computed numerical solution which must be accurately resolved to attain the desired level of error control, with respect to a given quantity of interest. Indeed, this may, of course, lead to refinement of elements near the boundary of the domain, but the granularity of these elements will be determined on the basis of *a posteriori* error indicators, rather than *a priori* considerations. The general

idea of CFEMs is to construct the underlying finite element spaces based on first generating a hierarchy of meshes, such that the finest mesh does indeed provide an accurate representation of the underlying computational domain, followed by the introduction of appropriate prolongation operators which determine how the finite element basis functions on the coarse mesh are defined in terms of those on the fine grid. In this way, CFEMs naturally lend themselves for exploitation within adaptive mesh refinement strategies in order that the discretization error is controlled in a reliable fashion. Indeed, refinement of the grid within a CFEM can be easily undertaken by simply constructing the underlying computational mesh by employing elements at different levels within the mesh tree hierarchy. These ideas have been exploited in the context of *hp*–version energy norm error estimation for DGCFEMs in our recent paper [18]. The purpose of the current article is to consider the extension of this approach to goal–oriented error estimation for DGCFEMs. For simplicity, we focus on the discretization of incompressible fluid flow problems, though we stress that this general framework may be applied to a wide range of application areas where complicated computational domains are employed.

The structure of this article is as follows. Section 2 introduces the composite finite element spaces, based on the ideas developed in [1, 20, 19]. In Section 3 we introduce the incompressible Navier–Stokes equations as a prototype partial differential equation and formulate the DGCFEM discretization. Goal–oriented *a posteriori* error estimates are formulated in Section 4, together with the design of an appropriate adaptive refinement strategy. The practical performance of the proposed error indicators for a range of two–dimensional problems is studied in Section 5. Finally, in Section 6 we summarize the work presented in this article and draw some conclusions.

2. Construction of the composite finite element spaces

Throughout this article, we write $\Omega \subset \mathbb{R}^d$, $d > 1$, to denote a bounded, connected Lipschitz domain, with boundary $\partial\Omega$. Moreover, it is implicitly assumed that Ω may be a complicated domain, in the sense that it may contain micro-structures. The purpose of this section is to briefly outline the construction of the hierarchy of CFEM meshes needed for the discretization of partial differential equations posed over Ω; for a more detailed discussion, we refer to [1], cf. also [18, 20, 19].

In order to construct the physical meshes, which will later be referred to as the *composite finite element* meshes, we first construct a hierarchy of (overlapping) reference meshes. To this end, we define $\hat{\mathcal{M}}_H \equiv \hat{\mathcal{M}}_{h_1}$ to be an overlapping mesh in the sense that it does not necessarily conform to the boundary of the underlying computational domain Ω. Thereby, we have that $\Omega \subset \Omega_H = \big(\bigcup_{\hat{\kappa}\in\hat{\mathcal{M}}_H} \hat{\kappa}\big)^\circ$ and $\hat{\kappa}^\circ \cap \Omega \neq \emptyset$ for all $\hat{\kappa} \in \hat{\mathcal{M}}_H$, where, for a closed set $D \subset \mathbb{R}^d$, D° denotes the interior of D. The hierarchy of reference meshes $\hat{\mathcal{M}}_{h_i}$, $i = 2, 3, \ldots, \ell$, are now constructed based on adaptively refining $\hat{\mathcal{M}}_H$; the sequence of refinement steps are stored in a tree structure, denoted by $\hat{\mathfrak{T}}$. In particular, the elements $\hat{\kappa}$ which belong to the initial coarse mesh $\hat{\mathcal{M}}_H$ are stored as the roots of the tree $\hat{\mathfrak{T}}$; we assign these elements with a level number $L = 1$. Elements $\hat{\kappa}$ at the bottom of the tree, i.e., those which do not possess any children, and which intersect the boundary of Ω, i.e., so that $\hat{\kappa}^\circ \cap \partial\Omega \neq \emptyset$, are marked for refinement; the resulting child elements of $\hat{\kappa}$ which intersect Ω are then stored at level $L = 2$, and assigned $\hat{\kappa}$ as their parent element. This process is continuously repeated until the elements within the tree structure $\hat{\mathfrak{T}}$ provide a sufficiently accurate description of the boundary of the computational domain. This is understood in the following sense: for each $\hat{\kappa} \in \hat{\mathfrak{T}}$ satisfying

$$\text{children}(\hat{\kappa}) = 0 \quad \wedge \quad \hat{\kappa}^\circ \cap \partial\Omega \neq \emptyset,$$

we require that $h_{\hat{\kappa}} < \texttt{TOL}$, where $h_{\hat{\kappa}} = \text{diam}(\hat{\kappa})$, $\text{children}(\hat{\kappa})$ denotes the number of children that element $\hat{\kappa}$ possesses, and $\texttt{TOL}$ is a user defined tolerance which characterizes the required resolution of the boundary, cf. [19]. This general strategy is summarized in Algorithm 2.1, cf. [1, 18]. For the purposes of this article, we employ local isotropic refinement of the elements in the computational mesh, though anisotropic refinement may also be admitted.

Writing $\ell = L_{\max}$ to denote the maximum level of refinement within the hierarchical tree $\hat{\mathfrak{T}}$, the reference meshes $\{\hat{\mathcal{M}}_{h_i}\}_{i=1}^{\ell}$, may be defined in the following manner

$$\hat{\mathcal{M}}_{h_i} \;=\; \Big\{\hat{\kappa} \in \hat{\mathfrak{T}} : \text{level}(\hat{\kappa}) = i \vee (\text{level}(\hat{\kappa}) \leq i \wedge \text{children}(\hat{\kappa}) = 0)\Big\},$$

where $\text{level}(\hat{\kappa})$ denotes the level of the element $\hat{\kappa}$ in $\hat{\mathfrak{T}}$. We stress that Algorithm 2.1 simply provides a generic refinement algorithm which may be employed to generate the sequence of reference meshes $\{\hat{\mathcal{M}}_{h_i}\}_{i=1}^{\ell}$, though alternative sequences of hierarchical meshes may be exploited within the

Algorithm 2.1 Construction of the hierarchical refinement tree $\hat{\mathfrak{T}}$

Store the elements $\hat{\kappa} \in \hat{\mathcal{M}}_H$ as the root nodes in the hierarchical tree $\hat{\mathfrak{T}}$.
Set the level number $L = 1$
for all $\hat{\kappa} \in \hat{\mathfrak{T}}$ such that children$(\hat{\kappa}) = 0$ **do**
    if $\hat{\kappa}^\circ \cap \partial\Omega \neq \emptyset$ **then**
        Refine $\hat{\kappa} = \bigcup_{i=1}^{n_{\hat{\kappa}}} \hat{\kappa}_i$, $n_{\hat{\kappa}} \geq 1$.
        for $i = 1, \ldots, n_{\hat{\kappa}}$ **do**
            if $\hat{\kappa}_i \cap \Omega \neq \emptyset$ **then**
                Store $\hat{\kappa}_i$ in $\hat{\mathfrak{T}}$ with parent $\hat{\kappa}$ and level number $L + 1$.
            end if
        end for
    end if
end for
Perform additional refinement of elements in $\hat{\mathfrak{T}}$ to undertake appropriate mesh smoothing, e.g., to limit the number of hanging nodes per element face, etc.
if The elements $\hat{\kappa} \in \mathfrak{T}$, with children$(\hat{\kappa}) = 0$, are *sufficiently* fine, in the sense that they are 'close' to the boundary $\partial\Omega$, cf. above. **then**
    `STOP`.
else
    Set $L = L + 1$, and `GOTO` 3.
end if

CFEM framework. On the basis of the finest reference mesh $\hat{\mathcal{M}}_{h_\ell}$, we may define the corresponding finest physical mesh $\mathcal{M}_{h_\ell}$ in the following manner.

Given that the above stopping criterion is satisfied, vertex points which are 'close' to the boundary $\partial\Omega$, i.e., vertices $\hat{x}_v$ which satisfy $\text{dist}(\hat{x}_v, \partial\Omega) \ll h_{\hat{\kappa}}$, $\hat{x}_v \in \hat{\kappa}$, are moved onto the boundary of the computational domain. During this procedure, any elements present in the tree $\hat{\mathfrak{T}}$ which lie outside of Ω after vertex node movement are removed from the tree; for distinction, we refer to this 'cropped' tree by $\mathfrak{T}$, though we stress that in practice, it is only necessary to store one such tree structure. On the basis of the tree data structure $\mathfrak{T}$, the physical meshes are constructed based on agglomerating elements which share a common parent within a given level of the mesh tree hierarchy $\mathfrak{T}$. More precisely, we first introduce the following notation: for $\kappa_{\mathcal{C}} \in \mathfrak{T}$ with level$(\kappa_{\mathcal{C}}) = j$, we write $\mathfrak{F}_i^j(\kappa_{\mathcal{C}})$, $j \geq i$, to denote the unique element $\kappa_{\mathcal{P}} \in \mathfrak{T}$ with level$(\kappa_{\mathcal{P}}) = i$ who is directly related to $\kappa_{\mathcal{C}}$ in the sense

that $\kappa_{\mathcal{C}} \subset \kappa_{\mathcal{P}}$; i.e., $\kappa_{\mathcal{C}}$ has resulted from subsequent refinement of $\kappa_{\mathcal{P}}$. In the trivial case when $j = i$, $\mathfrak{F}_i^j(\kappa_{\mathcal{C}}) = \kappa_{\mathcal{C}}$. Thereby, the physical meshes $\{\mathcal{M}_{h_i}\}_{i=1}^{\ell}$ may be constructed from $\mathfrak{T}$ as follows:

$$\begin{aligned}\mathcal{M}_{h_i} \;=\; & \{\kappa : (\kappa \in \mathfrak{T} \wedge \text{level}(\kappa) \leq i \wedge \text{children}(\kappa) = 0) \\ & \vee (\kappa = \cup_{\kappa' \in \mathfrak{T}} \kappa' : \text{children}(\kappa') = 0 \ \wedge \mathfrak{F}_i^j(\kappa') = P, j = \text{level}(\kappa') \\ & \wedge P \text{ is identical for all members of this set})\}\,.\end{aligned}$$

We point out that the physical meshes $\{\mathcal{M}_{h_i}\}_{i=1}^{\ell}$ may consist of general polygonal/polyhedral element domains. We refer to the coarsest physical mesh $\mathcal{M}_{h_1}$ as the CFE mesh, and accordingly write $\mathcal{M}_{\texttt{CFE}} \equiv \mathcal{M}_{h_1}$. As a simple example, we consider the simple case of a square $(-1, 1)^2$, which has had a diamond, with vertices $(0, \pm 3/5)$ and $(\pm 1/5, 0)$, removed. The corresponding reference and physical meshes $\{\hat{\mathcal{M}}_{h_i}\}_{i=1}^{\ell}$ and $\{\mathcal{M}_{h_i}\}_{i=1}^{\ell}$, respectively, are depicted in Figure 1 for $\ell = 3$.

On the basis of the (polygonal/polyhedral) mesh $\mathcal{M}_{\texttt{CFE}}$, given the polynomial degree $\mathfrak{p} \geq 1$, we may define the corresponding composite finite element space $V(\mathcal{M}_{\texttt{CFE}}, \mathfrak{T}, \mathfrak{p})$ by

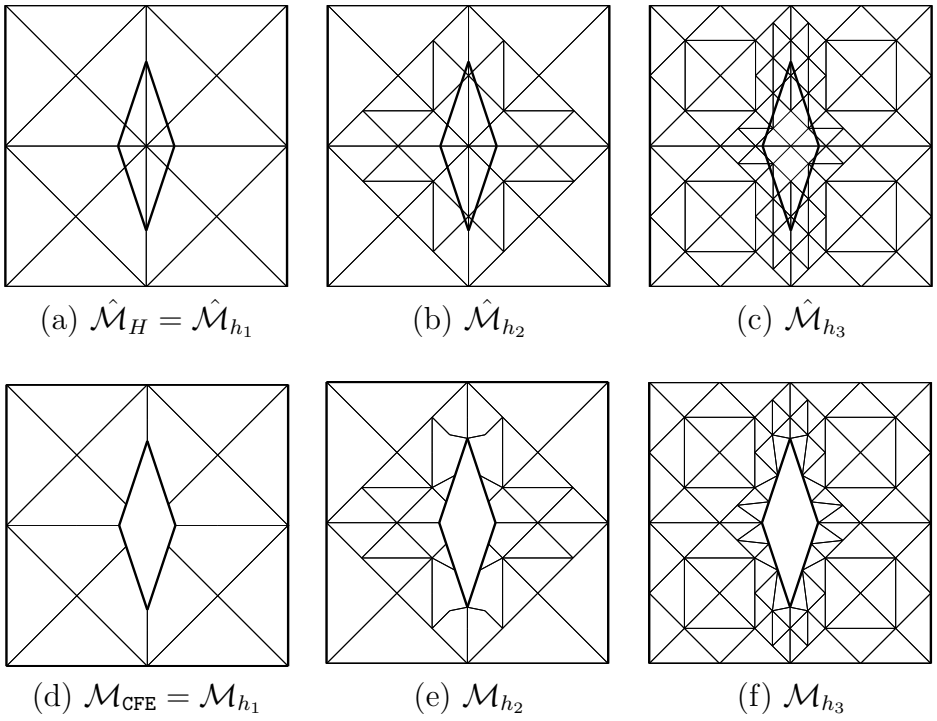
$$V(\mathcal{M}_{\texttt{CFE}}, \mathfrak{T}, \mathfrak{p}) = \{u \in L_2(\Omega) : u|_\kappa \in \mathcal{P}_{\mathfrak{p}}(\kappa),\ \kappa \in \mathcal{M}_{\texttt{CFE}}\},$$

where $\mathcal{P}_{\mathfrak{p}}(\kappa)$ denotes the space of polynomials of total degree $\mathfrak{p}$ over κ.

Remark 1. *We note that the construction of the elemental polynomial spaces present in the composite finite element space $V(\mathcal{M}_{\texttt{CFE}}, \mathfrak{T}, \mathfrak{p})$ may be constructed in a number of different ways. For example, in [1, 18] we first define polynomial bases on the finest level physical mesh $\mathcal{M}_{h_\ell}$, which consists of standard element shapes, i.e., triangles/quadrilaterals in 2D, etc; then polynomial bases may be defined on the coarse level polygonal/polyhedral elements by employing an appropriate prolongation operator, cf. [20, 19]. Alternatively, bases may be defined directly in the physical space, using either the bounding box technique employed in [12], or the Gram-Schmidt orthogonalization technique developed in [4, 5, 6], for example.*

3. Model problem and DGCFEM discretization

Consider the flow of an incompressible fluid confined in a domain $\Omega \in \mathbb{R}^d$, $d > 1$, with boundary $\partial\Omega = \partial\Omega_{\text{D}} \cup \partial\Omega_{\text{N}}$. Here, we impose a Dirichlet boundary condition on $\partial\Omega_{\text{D}}$ and a natural Neumann condition on $\partial\Omega_{\text{N}}$. By



(a) $\hat{\mathcal{M}}_H = \hat{\mathcal{M}}_{h_1}$ (b) $\hat{\mathcal{M}}_{h_2}$ (c) $\hat{\mathcal{M}}_{h_3}$

(d) $\mathcal{M}_{\texttt{CFE}} = \mathcal{M}_{h_1}$ (e) $\mathcal{M}_{h_2}$ (f) $\mathcal{M}_{h_3}$

Figure 1: Hierarchy of meshes (inner diamond is marked in bold): (a)–(c) Reference meshes; (d)–(f) Corresponding physical meshes.

introducing the Reynolds number Re, we consider the non–dimensionalized Navier–Stokes equations: find the velocity $\mathbf{u}$ and pressure p such that

$$-\frac{1}{Re}\nabla^2\mathbf{u} + \nabla\cdot(\mathbf{u}\otimes\mathbf{u}) + \nabla p = \mathbf{0}, \quad \text{in } \Omega, \tag{1}$$

$$\nabla\cdot\mathbf{u} = 0, \quad \text{in } \Omega, \tag{2}$$

with boundary conditions

$$\mathbf{u} = \mathbf{g}_{\mathrm{D}} \quad \text{on } \partial\Omega_{\mathrm{D}}, \qquad \frac{1}{Re}\frac{\partial\mathbf{u}}{\partial\mathbf{n}} - p\mathbf{n} = 0 \quad \text{on } \partial\Omega_{\mathrm{N}}, \tag{3}$$

where $\mathbf{n}$ is the unit outward normal vector to the boundary $\partial\Omega$.

In order to define the DGCFEM discretization of the incompressible Navier–Stokes equations defined by (1)–(3), we first introduce some notation. We denote by $\mathcal{F}^{\mathcal{I}}(\mathcal{M}_{\texttt{CFE}})$ the set of all interior faces of the partition $\mathcal{M}_{\texttt{CFE}}$ of

Ω, and by $\mathcal{F}^{\mathcal{B}_{\rm D}}(\mathcal{M}_{\tt CFE})$ the set of (Dirichlet) boundary faces of $\mathcal{M}_{\tt CFE}$ which lie on $\partial\Omega_{\rm D}$. Furthermore, we define $\mathcal{F}(\mathcal{M}_{\tt CFE}) = \mathcal{F}^{\mathcal{I}}(\mathcal{M}_{\tt CFE}) \cup \mathcal{F}^{\mathcal{B}_{\rm D}}(\mathcal{M}_{\tt CFE})$. The boundary $\partial\kappa$ of an element κ and the sets $\partial\kappa \setminus \partial\Omega$ and $\partial\kappa \cap \partial\Omega_{\rm D}$ will be identified in a natural way with the corresponding subsets of $\mathcal{F}(\mathcal{M}_{\tt CFE})$.

Next, we define average and jump operators. To this end, let κ^+ and κ^- be two adjacent elements of $\mathcal{M}_{\tt CFE}$, and $\mathbf{x}$ be an arbitrary point on the interior face $F = \partial\kappa^+ \cap \partial\kappa^- \subset \mathcal{F}^{\mathcal{I}}(\mathcal{M}_{\tt CFE})$. Given scalar–, vector–, and matrix–valued functions q, $\mathbf{v}$, and $\underline{\tau}$, respectively, that are smooth inside each element $\kappa^\pm$, by $(q^\pm, \mathbf{v}^\pm, \underline{\tau}^\pm)$ we denote the traces of $(q, \mathbf{v}, \underline{\tau})$ on F taken from within the interior of $\kappa^\pm$, respectively. Then, we introduce the averages at $\mathbf{x} \in F$:

$$\{\!\{q\}\!\} = (q^+ + q^-)/2, \quad \{\!\{\mathbf{v}\}\!\} = (\mathbf{v}^+ + \mathbf{v}^-)/2, \quad \{\!\{\underline{\tau}\}\!\} = (\underline{\tau}^+ + \underline{\tau}^-)/2.$$

Similarly, the jumps at $\mathbf{x} \in F$ are given by

$$\begin{aligned} [\![q]\!] &= q^+\mathbf{n}_{\kappa^+} + q^-\mathbf{n}_{\kappa^-}, & [\![\mathbf{v}]\!] &= \mathbf{v}^+ \cdot \mathbf{n}_{\kappa^+} + \mathbf{v}^- \cdot \mathbf{n}_{\kappa^-}, \\ \underline{[\![\mathbf{v}]\!]} &= \mathbf{v}^+ \otimes \mathbf{n}_{\kappa^+} + \mathbf{v}^- \otimes \mathbf{n}_{\kappa^-}, & [\![\underline{\tau}]\!] &= \underline{\tau}^+\mathbf{n}_{\kappa^+} + \underline{\tau}^-\mathbf{n}_{\kappa^-}. \end{aligned}$$

On boundary faces $F \subset \partial\Omega$, we set $\{\!\{q\}\!\} = q$, $\{\!\{\mathbf{v}\}\!\} = \mathbf{v}$, $\{\!\{\underline{\tau}\}\!\} = \underline{\tau}$, $[\![q]\!] = q\mathbf{n}$, $[\![\mathbf{v}]\!] = \mathbf{v} \cdot \mathbf{n}$, $\underline{[\![\mathbf{v}]\!]} = \mathbf{v} \otimes \mathbf{n}$, and $[\![\underline{\tau}]\!] = \underline{\tau}\mathbf{n}$.

Writing $\mathbf{V}_h = [V(\mathcal{M}_{\tt CFE}, \mathfrak{T}, \mathfrak{p})]^d$ and $Q_h = V(\mathcal{M}_{\tt CFE}, \mathfrak{T}, \mathfrak{p} - 1)$ we approximate the Navier–Stokes problem (1)–(3) by the mixed DGCFEM: find $(\mathbf{u}_h, p_h) \in \mathbf{V}_h \times Q_h$ such that

$$\begin{cases} A_h(\mathbf{u}_h, \mathbf{v}_h) + C_h(\mathbf{u}_h, \mathbf{v}_h) + B_h(\mathbf{v}_h, p_h) &= F_h(\mathbf{v}_h), \\ -B_h(\mathbf{u}_h, q_h) &= G_h(q_h) \end{cases} \tag{4}$$

for all $(\mathbf{v}_h, q_h) \in \mathbf{V}_h \times Q_h$. The forms A_h, B_h, and C_h are discontinuous Galerkin forms that discretize the Laplacian, the incompressibility constraint, and the nonlinear convective terms, respectively, with corresponding right-

hand sides F_h and G_h. These forms are given by

$$\begin{aligned}
A_h(\mathbf{u},\mathbf{v}) =& \frac{1}{Re}\left(\sum_{\kappa\in\mathcal{M}_{\mathtt{CFE}}}\int_\kappa \nabla\mathbf{u}:\nabla\mathbf{v}\,d\mathbf{x} + \sum_{F\in\mathcal{F}(\mathcal{M}_{\mathtt{CFE}})}\int_F \sigma\underline{[\![\mathbf{u}]\!]}:\underline{[\![\mathbf{v}]\!]}\,ds\right.\\
&\left. - \sum_{F\in\mathcal{F}(\mathcal{M}_{\mathtt{CFE}})}\int_F \left(\{\!\{\nabla\mathbf{v}\}\!\}:\underline{[\![\mathbf{u}]\!]} + \{\!\{\nabla\mathbf{u}\}\!\}:\underline{[\![\mathbf{v}]\!]}\right)ds\right),\\
B_h(\mathbf{v},q) =& -\sum_{\kappa\in\mathcal{M}_{\mathtt{CFE}}}\int_\kappa q\,\nabla\cdot\mathbf{v}\,d\mathbf{x} + \sum_{F\in\mathcal{F}(\mathcal{M}_{\mathtt{CFE}})}\int_F \{\!\{q\}\!\}[\![\mathbf{v}]\!]\,ds,\\
C_h(\mathbf{u},\mathbf{v}) =& -\sum_{\kappa\in\mathcal{M}_{\mathtt{CFE}}}\int_\kappa \mathfrak{F}(\mathbf{u}):\nabla\mathbf{v}\,d\mathbf{x} + \sum_{\kappa\in\mathcal{M}_{\mathtt{CFE}}}\int_{\partial\kappa\setminus\partial\Omega}\mathcal{H}(\mathbf{u}^+,\mathbf{u}^-,\mathbf{n})\cdot\mathbf{v}^+\,ds\\
&+\int_{\partial\Omega}\mathcal{H}(\mathbf{u}^+,\mathbf{u}_\Gamma(\mathbf{u}^+),\mathbf{n})\cdot\mathbf{v}^+\,ds,\\
F_h(\mathbf{v}) =& \int_\Omega \mathbf{f}\cdot\mathbf{v}\,d\mathbf{x} + \sum_{F\in\mathcal{F}^{\mathcal{B}_{\mathrm{D}}}(\mathcal{M}_{\mathtt{CFE}})}\int_F(\sigma\mathbf{g}_{\mathrm{D}}\cdot\mathbf{v} - (\mathbf{g}_{\mathrm{D}}\otimes\mathbf{n}):\nabla\mathbf{v})ds,\\
G_h(q) =& -\sum_{F\in\mathcal{F}^{\mathcal{B}_{\mathrm{D}}}(\mathcal{M}_{\mathtt{CFE}})}\int_F q\,\mathbf{g}_{\mathrm{D}}\cdot\mathbf{n}\,ds.
\end{aligned}$$

Here, the flux $\mathfrak{F}$ is given by $\mathfrak{F}(\mathbf{u}) = \mathbf{u}\otimes\mathbf{u}$; moreover, $\mathcal{H}(\cdot,\cdot,\cdot)$ denotes the Lax-Friedrichs flux defined as

$$\mathcal{H}(\mathbf{v},\mathbf{w},\mathbf{n}) := \frac{1}{2}\left(\mathfrak{F}(\mathbf{v})\cdot\mathbf{n} + \mathfrak{F}(\mathbf{w})\cdot\mathbf{n} - \alpha(\mathbf{w}-\mathbf{v})\right),$$

where $\alpha := 2\max(|\mathbf{v}\cdot\mathbf{n}|,|\mathbf{w}\cdot\mathbf{n}|)$. The boundary function $\mathbf{u}_\Gamma$ is given by $\mathbf{u}_\Gamma(\mathbf{u}) = \mathbf{g}_{\mathrm{D}}$ on $\partial\Omega_{\mathrm{D}}$ and $\mathbf{u}_\Gamma(\mathbf{u}) = \mathbf{u}^+$ on $\partial\Omega_{\mathrm{N}}$. Finally, the interior penalty parameter σ is defined by $\sigma|_F = \gamma h_F^{-1}$ for $F\in\mathcal{F}(\mathcal{M}_{\mathtt{CFE}})$, where $\gamma>0$ is a sufficiently large constant, and h_F is a representative length scale associated to the face F; see [1] and [12] for details.

Writing

$$\mathcal{N}((\mathbf{u},p),(\mathbf{v},q)) = A_h(\mathbf{u},\mathbf{v}) + C_h(\mathbf{u},\mathbf{v}) + B_h(\mathbf{v},p) - B_h(\mathbf{u},q) - F_h(\mathbf{v}) - G_h(q),$$

the DGCFEM (4) may be written in the following compact manner: find $(\mathbf{u}_h,p_h)\in\mathbf{V}_h\times Q_h$ such that

$$\mathcal{N}((\mathbf{u}_h,p_h),(\mathbf{v}_h,q_h)) = 0 \quad \forall(\mathbf{v}_h,q_h)\in\mathbf{V}_h\times Q_h.$$

Remark 2. *We point out that the elementwise integrals over a general polygonal/polyhedral element $\kappa \in \mathcal{M}_{\mathtt{CFE}}$, arising in the definition of the DGCFEM* (4)*, are computed based on employing standard integration schemes over the fine (regularly shaped) elements present at the foot of the refinement tree $\mathfrak{T}$, which, when agglomerated, form κ, cf. [5, 12], for example.*

4. *A posteriori* error estimation and mesh adaptation

For the purposes of this article we consider a goal-oriented *a posteriori* error indicator based on employing the general dual–weighted–residual (DWR) framework introduced by R. Rannacher and his collaborators; see [7], for example. To this end, given a (linear, for simplicity) target functional J, we state the following *a posteriori* error estimate.

Proposition 1. *Let $(\mathbf{u}, p)$ and $(\mathbf{u}_h, p_h)$ denote the solutions of* (1)–(3) *and* (4)*, respectively, then the following error representation formula holds:*

$$J(\mathbf{u}, p) - J(\mathbf{u}_h, p_h) = -\mathcal{N}((\mathbf{u}_h, p_h), (\mathbf{z} - \mathbf{z}_h, r - r_h)) \equiv \sum_{\kappa \in \mathcal{M}_{\mathtt{CFE}}} \eta_\kappa$$

for all $(\mathbf{z}_h, r_h) \in \mathbf{V}_h \times Q_h$. Here, $(\mathbf{z}, r)$ denotes the solution of the corresponding dual/adjoint problem: find $(\mathbf{z}, r)$ such that

$$\mathcal{M}((\mathbf{u}, p), (\mathbf{u}_h, p_h); (\mathbf{w}, s), (\mathbf{z}, r)) = J(\mathbf{w}, s) \quad \forall (\mathbf{w}, s),$$

where

$$\begin{aligned} &\mathcal{M}((\mathbf{u}, p), (\mathbf{u}_h, p_h); (\mathbf{u} - \mathbf{u}_h, p - p_h), (\mathbf{w}, s)) \\ &\quad = \int_0^1 \mathcal{N}'[\theta(\mathbf{u}, p) + (1 - \theta)(\mathbf{u}_h, p_h)]((\mathbf{u} - \mathbf{u}_h, p - p_h), (\mathbf{w}, s))\, \mathrm{d}\theta, \end{aligned}$$

for some $(\mathbf{w}, s)$. Here, $\mathcal{N}'[(\mathbf{w}, s)]((\cdot, \cdot), (\mathbf{v}, q))$ denotes the Fréchet derivative of $(\mathbf{u}, p) \to \mathcal{N}((\mathbf{u}, p), (\mathbf{v}, q))$, for $(\mathbf{v}, q)$ fixed, at some $(\mathbf{w}, s)$.

We now briefly outline the construction of a sequence of adaptively refined meshes $\{\mathcal{M}_{\mathtt{CFE},l}\}_{l \geq 0}$, with $\mathcal{M}_{\mathtt{CFE},0} = \mathcal{M}_{\mathtt{CFE}}$. From an implementation point of view, the error indicators η_κ, $\kappa \in \mathcal{M}_{\mathtt{CFE},l}$, $l \geq 0$, are computed based on approximating the dual solution $(\mathbf{z}, r)$ on the composite finite element mesh $\mathcal{M}_{\mathtt{CFE},l}$, $l \geq 0$, using polynomials one degree higher than those employed for the computation of $(\mathbf{u}_h, p_h)$. On the basis of the size of the

modulus of the local error indicators, i.e., $|\eta_\kappa|$, the elements in the mesh $\mathcal{M}_{\texttt{CFE},l}$, $l \geq 0$, may be marked for refinement/derefinement using a standard marking strategy; here, for this purpose, we employ the fixed fraction strategy with refinement and derefinement parameters set to 15% and 5%, respectively. Refinement/derefinement of the polygonal/polyhedral elements present in $\mathcal{M}_{\texttt{CFE},l}$, $l \geq 0$, can be easily realized based on employing (agglomerated) elements at different levels within the mesh tree hierarchy $\mathfrak{T}$, cf. the definition of the physical meshes $\mathcal{M}_{h_i}$, $i = 1, \ldots, \ell$. More precisely, suppose that element $\kappa \in \mathcal{M}_{\texttt{CFE},l}$, $l \geq 0$, with level$(\kappa) = k$, say, is marked for subdivision, then assuming that children$(\kappa) = n_\kappa \neq 0$, writing $\kappa_{\mathfrak{C}_i}$, $i = 1, \ldots, n_\kappa$, to denote the children of κ stored in the tree $\mathfrak{T}$, we subdivide $\kappa = \cup_{i=1}^{n_\kappa} \kappa_{\mathcal{C}_i}$, where

$$\kappa_{\mathcal{C}_i} \;\; = \;\; \cup_{\kappa' \in \mathfrak{T}} \kappa' : (\text{children}(\kappa') = 0 \wedge \mathfrak{F}_{k+1}^{j}(\kappa') = \kappa_{\mathfrak{C}_i}, \text{level}(\kappa') = j),$$

$i = 1, \ldots, n_\kappa$. On the other hand if children$(\kappa) = 0$, then, given that κ is a standard–shaped element, refinement of this element domain may be undertaken in the usual manner, giving rise to elements at a potentially deeper level than ℓ in the mesh tree $\mathfrak{T}$. In addition, mesh smoothing is undertaken to ensure that the resulting mesh is 1-irregular, cf. [1].

5. Numerical Experiments

In this section we present a series of numerical examples to demonstrate the practical performance of the proposed adaptive algorithm based on employing composite finite element spaces.

5.1. Example 1: Flow through a square with micro-structures

In this first example, we select Ω to be the square $(0, 5)^2$ in two dimensions, which has had a uniform set of 25 circular holes removed, cf. Figure 2(a), which depicts the initial mesh. Here, the flow enters the domain from the left-hand side of the boundary, i.e., along $x = 0$, where we assume Poiseuille flow enters Ω. No slip Dirichlet boundary conditions are imposed on the top and bottom parts of Ω, i.e., along $y = 5$ and $y = 0$, respectively, as well as on the circular holes. Finally, a Neumann condition is imposed along $x = 5$. Setting $Re = 100$, we select the functional of interest to be equal to the value of the pressure p at the point $(2.0, 2.5)$, i.e., $J(\mathbf{u}, p) = p(2.0, 2.5)$; on the basis of a fine grid computation, we find that $J(\mathbf{u}, p) \approx 6.02846\text{E-}3$.

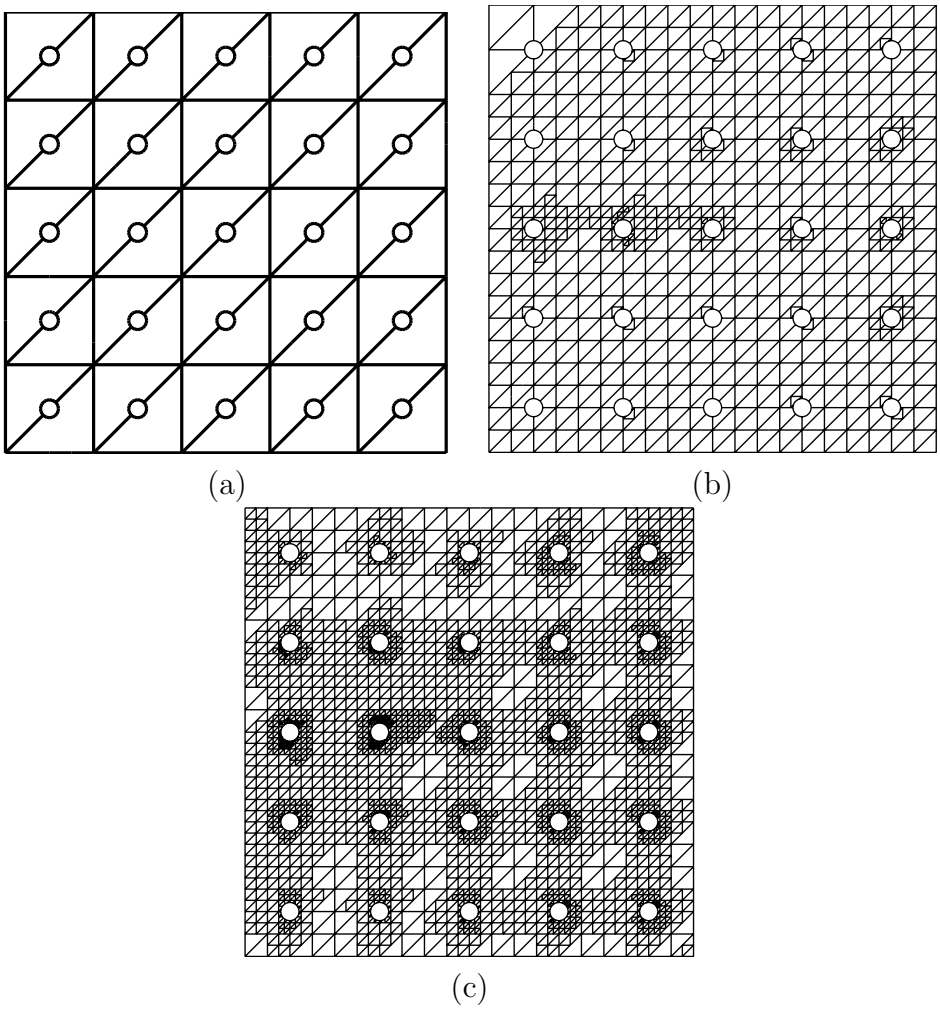


Figure 2: Example 1. (a) Initial composite mesh $\mathcal{M}_{\mathtt{CFE}}$ consisting of 50 polygonal elements; (b) Composite mesh after 6 adaptive refinements with 1061 elements; (c) Composite mesh after 9 adaptive refinements with 4386 elements.

We select the initial composite finite element mesh $\mathcal{M}_{\mathtt{CFE}}$, based on employing Algorithm 2.1, to consist of 50 polygonal elements, cf. Figure 2(a). In Table 1, we demonstrate the performance of the proposed adaptive strategy with $\gamma = 10$ and the polynomial degree $\mathfrak{p} = 2$; here, we show the number of elements in composite mesh $\mathcal{M}_{\mathtt{CFE}}$, the number of degrees of freedom in underlying finite element space $\mathbf{V}_h \times Q_h$, the true error in the functional

No of Eles	No of Dofs	$J(\mathbf{u},p)-J(\mathbf{u}_h,p_h)$	$\sum_{\kappa\in\mathcal{M}_{\mathrm{CFE}}}\eta_\kappa$	θ
50	750	-2.205E-2	-1.197E-2	0.54
74	1110	-2.470E-2	-9.202E-3	0.37
125	1875	-1.875E-2	-7.074E-3	0.38
215	3225	-1.479E-2	-5.749E-3	0.39
347	5205	-8.501E-3	-5.742E-3	0.68
677	10155	-2.191E-3	-1.535E-3	0.70
1061	15915	-1.754E-3	-1.289E-3	0.73
1558	23370	-4.560E-4	-3.922E-4	0.86
2531	37965	-1.228E-4	-8.521E-5	0.69
4386	65790	-3.801E-5	-3.801E-5	1.00

Table 1: Example 1: Adaptive algorithm.

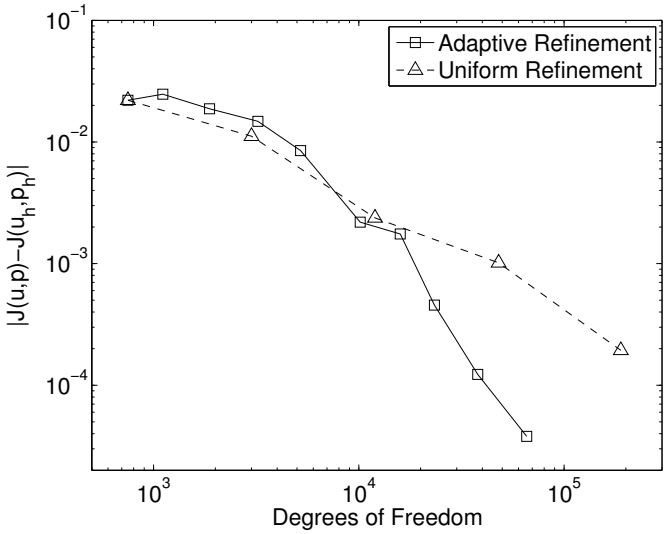


Figure 3: Example 1: Comparison between uniform and adaptive mesh refinement.

$J(\mathbf{u},p)-J(\mathbf{u}_h,p_h)$, the computed error representation formula $\sum_{\kappa\in\mathcal{M}_{\mathrm{CFE}}}\eta_\kappa$, and the effectivity index $\theta=\sum_{\kappa\in\mathcal{M}_{\mathrm{CFE}}}\eta_\kappa/(J(\mathbf{u},p)-J(\mathbf{u}_h,p_h))$. Here, we see that, even on such coarse finite element meshes, the quality of the computed error representation formula is relatively good, in the sense that the effectivity indices are not too far away from unity. We also notice that

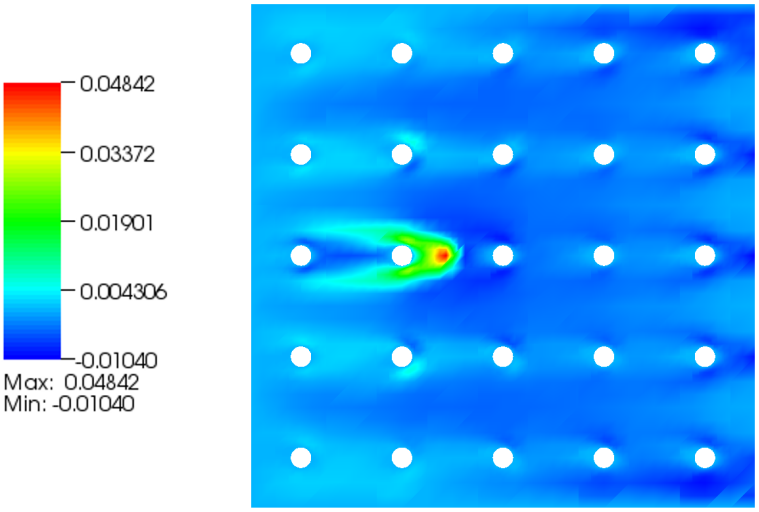


Figure 4: Example 1. Pressure component of the dual solution.

practical/engineering accuracy can be attained using a very small number of degrees of freedom; indeed, fewer degrees of freedom are necessary than what would be required to accurately mesh the domain Ω using standard element shapes, i.e., triangles/quadrilaterals. The results presented in Table 1 are plotted in Figure 3. In addition, here we compare the performance of the proposed adaptive refinement strategy with uniform mesh refinement. We observe that initially both strategies lead to a comparable error in the computed target functional of interest J, for a given number of degrees of freedom; however, as both refinement procedures continue, the adaptive algorithm leads to around an order of magnitude improvement in the error in J for a comparable number of degrees of freedom.

Figures 2(b) & 2(c) show the composite finite element meshes after 6 and 9 refinement steps, respectively. Here, we observe that initially refinement is concentrated in the vicinity of the point of interest; see in particular, the pressure component of the dual solution in Figure 4. However, as the refinement strategy continues, we observe that although much of the refinement is still concentrated in this same region, additional refinement is also undertaken in the vicinity of all of the circular holes present in Ω.

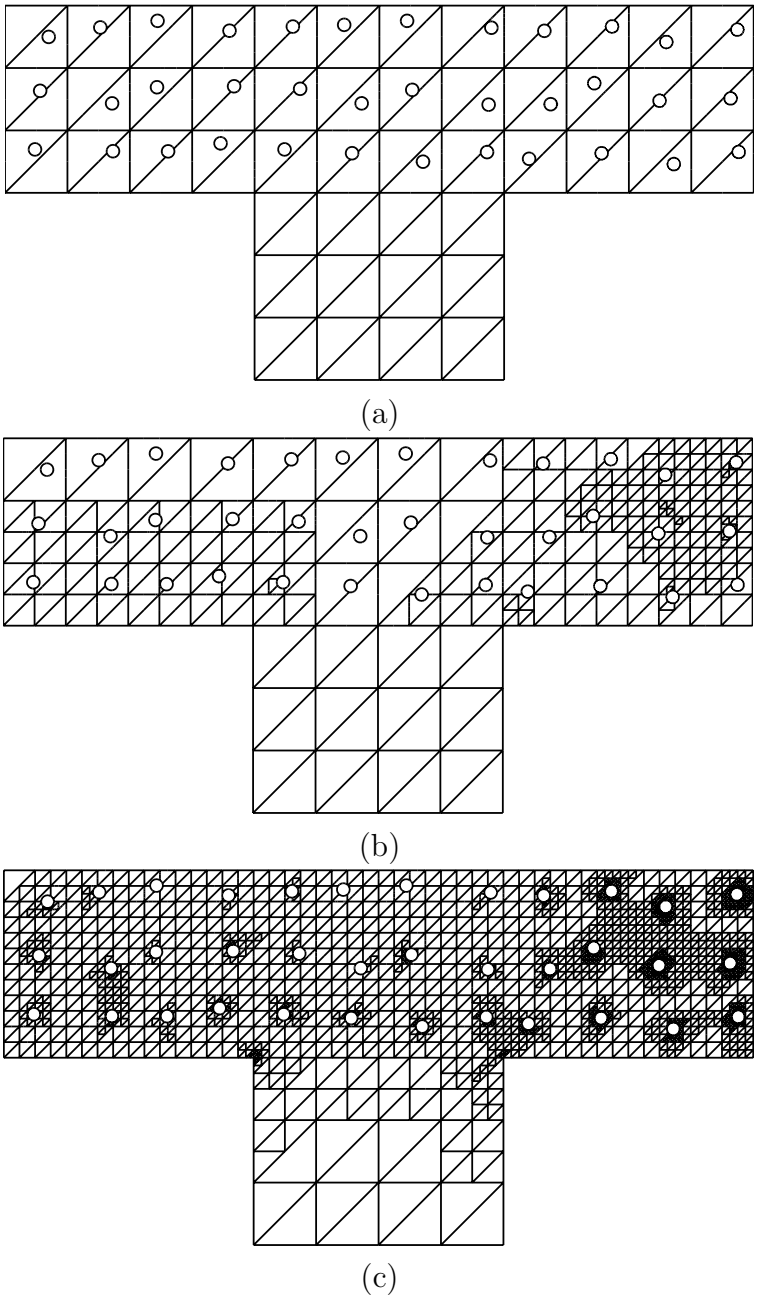


Figure 5: Example 2. (a) Initial composite finite element mesh consisting of 96 polygonal elements; (b) Composite mesh after 3 adaptive refinements with 408 elements; (c) Composite mesh after 7 adaptive refinements with 3118 elements.

No of Eles	No of Dofs	$J(\mathbf{u},p)-J(\mathbf{u}_h,p_h)$	$\sum_{\kappa\in\mathcal{M}_{\mathtt{CFE}}}\eta_\kappa$	θ
96	1440	-2.849E-2	-2.534E-2	0.89
159	2385	-1.702E-2	-1.430E-2	0.84
240	3600	-5.755E-3	-3.419E-3	0.59
408	6120	-2.974E-3	-1.554E-3	0.52
660	9900	-1.592E-3	-7.969E-4	0.50
1108	16620	-8.644E-4	-3.853E-4	0.45
1901	28515	-5.008E-4	-2.842E-4	0.57
3118	46770	-2.068E-4	-1.468E-4	0.71
5196	77940	-5.390E-5	-4.716E-5	0.87
8708	130620	-1.172E-5	-1.172E-5	1.00

Table 2: Example 2: Adaptive algorithm.

5.2. Example 2: Flow through a complicated T–pipe domain

In this second example we consider a variant of the previous numerical experiment. To this end, the computational domain Ω is defined to be a T–shaped pipe, which has had a series of randomly located holes removed from the horizontal section; see Figure 5(a) which shows the initial composite mesh consisting of 96 polygonal elements. Here, the inflow boundary is specified at the bottom of the vertical section of the pipe, i.e., along $y=-3$, $4\leq x\leq 8$, where Poiseuille flow enters Ω; the left-hand and right-hand side boundaries of the horizontal portion of the pipe, located at $x=0$, $0\leq y\leq 3$ and $x=12$, $0\leq y\leq 3$, are identified as Neumann boundaries. No slip boundary conditions are imposed on the remaining walls of the T–pipe geometry, together with the boundaries of the circular holes. As before, we set $Re=100$ and select $J(\mathbf{u},p)=p(11,1.5)\approx 2.27564\text{E-}3$.

In Table 2 we tabulate the results of the proposed adaptive refinement strategy, based on employing an initial composite mesh consisting of 96 elements with $\gamma=10$ and $\mathfrak{p}=2$, as before. As in the previous numerical experiment, we again notice that the effectivity indices θ are relatively good, given the coarse nature of the finite element meshes employed. Indeed, as the mesh is refined, we observe that θ improves and approaches unity. Again, here we observe that a sufficiently accurate (in terms of engineering accuracy) approximation to the target functional of interest may be computed with very few degrees of freedom. The results of Table 2 are depicted in Figure 6. As in the previous example, we again compare the performance of the

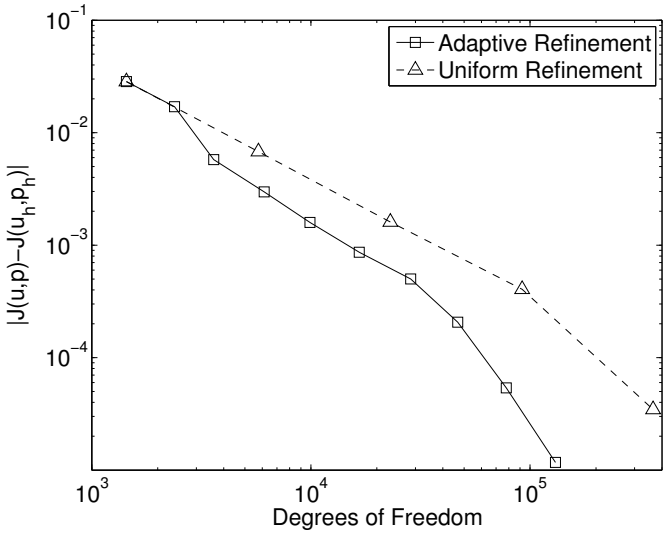


Figure 6: Example 2: Comparison between uniform and adaptive mesh refinement.

proposed adaptive refinement strategy with uniform mesh refinement. Here, we observe that adaptive refinement is always superior to uniform refinement, in the sense that the error in the computed target functional of interest J is smaller when adaptive refinement is employed, for a given number of degrees of freedom. Indeed, adaptive refinement leads to over an order of magnitude improvement in the error in J for a comparable number of degrees of freedom than the corresponding quantity computed in the case when uniform refinement is exploited.

In Figures 5(b) & 5(c) we show the composite finite element meshes after 3 and 7 adaptive refinement steps, respectively. In this setting, we now observe that the grid is strongly refined in the vicinity of the holes located to the right of the vertical inlet pipe where the point value of interest is located. As the refinement proceeds, some refinement of the holes to the left of the inlet pipe is observed, as well as near the reentrant corners where the vertical and horizontal sections of the T–pipe meet. The pressure component of the dual solution is depicted in Figure 7.

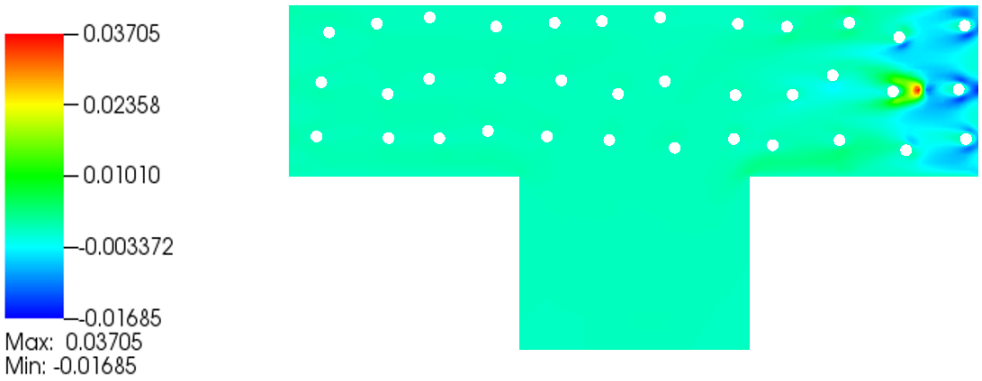


Figure 7: Example 2. Pressure component of the dual solution.

6. Conclusion and outlook

In this article we have considered the application of goal–oriented adaptive mesh refinement to the composite variant of the standard DGFEM for problems posed on complicated domains. As a prototype partial differential equation, we have focused our attention on the numerical approximation of the incompressible Navier–Stokes equations. The numerical experiments presented in this article clearly highlight the practical performance of this approach. Indeed, by admitting the use of general polygonal/polyhedral elements, the dimension of the underlying (composite) finite element space is independent of the features present in the computational domain. Thereby, coarse numerical approximations can be computed in an efficient manner to yield solutions which are sufficiently accurate in terms of engineering accuracy. While we have only considered incompressible fluid flows, this approach can be universally applied to a wide range of application areas, where the geometry imposes strong limitations on the number of degrees of freedom required to yield a sufficiently accurate mesh representation of the computational domain. Finally, we note that the extension of this work to include *hp*–mesh refinement follows in a natural fashion; for related work undertaken in the context of energy norm error estimation, we refer to our recent article [18].

Acknowledgments

PH acknowledges the financial support of the Leverhulme Trust.

References

[1] P. Antonietti, S. Giani, P. Houston, *hp*–Version composite discontinuous Galerkin methods for elliptic problems on complicated domains, SIAM J. Sci. Comput. 35 (2013) A1417–A1439.

[2] P. Antonietti, S. Giani, P. Houston, Domain decomposition preconditioners for discontinuous Galerkin methods for elliptic problems on complicated domains, J. Sci. Comput. (In press).

[3] D. Arnold, F. Brezzi, B. Cockburn, L. Marini, Unified analysis of discontinuous Galerkin methods for elliptic problems, SIAM J. Numer. Anal. 39 (2001) 1749–1779.

[4] F. Bassi, L. Botti, A. Colombo, An attempt to be mesh free, agglomeration based physical frame DG discretizations, Math. Models Methods Appl. Sci. (Submitted for publication).

[5] F. Bassi, L. Botti, A. Colombo, D. Di Pietro, P. Tesini, On the flexibility of agglomeration based physical space discontinuous Galerkin discretizations, J. Comput. Phys. 231 (2012) 45–65.

[6] F. Bassi, L. Botti, A. Colombo, S. Rebay, Agglomeration based discontinuous Galerkin discretization of the Euler and Navier-Stokes equations, Comput. & Fluids 61 (2012) 77–85.

[7] R. Becker, R. Rannacher, An optimal control approach to a-posteriori error estimation in finite element methods, in: A. Iserles (Ed.), Acta Numerica, Cambridge University Press, 2001, pp. 1–102.

[8] F. Brezzi, A. Buffa, K. Lipnikov, Mimetic finite differences for elliptic problems, M2AN Math. Model. Numer. Anal. 43 (2009) 277–295.

[9] E. Burman, P. Hansbo, Fictitious domain finite element methods using cut elements: I. A stabilized Lagrange multiplier method, Comput. Methods Appl. Mech. Engrg. 199 (2010) 2680–2686.

[10] E. Burman, P. Hansbo, An interior-penalty-stabilized Lagrange multiplier method for the finite-element solution of elliptic interface problems, IMA J. Numer. Anal. 30 (2010) 870–885.

[11] E. Burman, P. Hansbo, Fictitious domain finite element methods using cut elements: II. A stabilized Nitsche method, Appl. Numer. Math. 62 (2012) 328–341.

[12] A. Cangiani, E. Georgoulis, P. Houston, *hp*-Version discontinuous Galerkin methods on polygonal and polyhedral meshes, Math. Models Methods Appl. Sci. (In press).

[13] B. Cockburn, An introduction to the discontinuous Galerkin method for convection-dominated problems, in: Advanced numerical approximation of nonlinear hyperbolic equations (Cetraro, 1997), Springer, Berlin, 1998, pp. 151–268.

[14] B. Cockburn, G. Karniadakis, C.W. Shu (Eds.), Discontinuous Galerkin methods, Springer-Verlag, Berlin, 2000. Theory, computation and applications, Papers from the 1st International Symposium held in Newport, RI, May 24–26, 1999.

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

[16] T.P. Fries, T. Belytschko, The extended/generalized finite element method: an overview of the method and its applications, Internat. J. Numer. Methods Engrg. 84 (2010) 253–304.

[17] S. Giani, P. Houston, Domain decomposition preconditioners for discontinuous Galerkin discretizations of compressible fluid flows, Numer. Math. Theor. Meth. Appl. (In press).

[18] S. Giani, P. Houston, *hp*–Adaptive composite discontinuous Galerkin methods for elliptic problems on complicated domains, Num. Meth. Part. Diff. Eqs. (In press).

[19] W. Hackbusch, S. Sauter, Composite finite elements for problems containing small geometric details. Part II: Implementation and numerical results, Comput. Visual Sci. 1 (1997) 15–25.

[20] W. Hackbusch, S. Sauter, Composite finite elements for the approximation of PDEs on domains with complicated micro-structures, Numer. Math. 75 (1997) 447–472.

[21] J. Hesthaven, T. Warburton, Nodal discontinuous Galerkin methods, volume 54 of *Texts in Applied Mathematics*, Springer, New York, 2008. Algorithms, analysis, and applications.

[22] K. Lipnikov, D. Vassilev, I. Yotov, Discontinuous Galerkin and mimetic finite difference methods for coupled Stokes-Darcy flows on polygonal and polyhedral grids, Numer. Math. (2013) 1–40.

[23] A. Massing, Analysis and implementation of Finite Element Methods on overlapping and Fictitious Domains, Ph.D. thesis, University of Oslo, 2012.

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

[25] N. Sukumar, A. Tabarraei, Conforming polygonal finite elements, Internat. J. Numer. Methods Engrg. 61 (2004) 2045–2066.

[26] A. Tabarraei, N. Sukumar, Extended finite element method on polygonal and quadtree meshes, Comput. Methods Appl. Mech. Engrg. 197 (2007) 425–438.

[27] L. Beirão da Veiga, F. Brezzi, A. Cangiani, G. Manzini, L. Marini, A. Russo, Basic principles of virtual element methods, Math. Models Methods Appl. Sci. 23 (2013) 199–214.

[28] L. Beirão da Veiga, K. Lipnikov, G. Manzini, Arbitrary-order nodal mimetic discretizations of elliptic problems on polygonal meshes, SIAM J. Numer. Anal. 49 (2011) 1737–1760.

[29] D. Wirasaet, E. Kubatko, C. Michoski, S. Tanaka, J. Westerink, C. Dawson, Discontinuous Galerkin methods with nodal and hybrid modal/nodal triangular, quadrilateral, and polygonal elements for nonlinear shallow water flow, Comput. Methods Appl. Mech. Engrg. (In review).