INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2010; **62**:90–118
Published online 26 February 2009 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.2016

A posteriori error estimation and anisotropy detection with the dual-weighted residual method

Thomas Richter*, †

Institut für Angewandte Mathematik, INF 294, Universität Heidelberg, 69120 Heidelberg, Germany

SUMMARY

In this work we develop a new framework for *a posteriori* error estimation and detection of anisotropies based on the dual-weighted residual (DWR) method by Becker and Rannacher. The common approach for anisotropic mesh adaptation is to analyze the Hessian of the solution. Eigenvalues and eigenvectors indicate dominant directions and optimal stretching of elements. However, this approach is firmly linked to energy norm error estimation. Here, we extend the DWR method to anisotropic finite elements allowing for the direct estimation of directional errors with regard to given output functionals. The resulting meshes reflect anisotropic properties of both the solution and the functional. For the optimal measurement of the directional errors, the coarse meshes need some alignment with the dominant anisotropies. Numerical examples will demonstrate the efficiency of this method on various three-dimensional problems including a well-known Navier–Stokes benchmark. Copyright © 2009 John Wiley & Sons, Ltd.

Received 23 October 2007; Revised 12 January 2009; Accepted 16 January 2009

KEY WORDS: finite element methods; adaptivity; error estimation; mesh adaptation; anisotropy; Navier–Stokes

1. INTRODUCTION

The dual-weighted residual (DWR) method [1, 2] for *a posteriori* error estimation and adaptive remeshing or mesh refinement has been essentially established in the last years. Optimal mesh refinement using isotropic elements is well understood and has been demonstrated in various applications, such as fluid dynamics [3], reactive flow problems [4], or optimization and parameter identification problems [5].

However, for three-dimensional problems, adaptive mesh refinement rapidly reaches its limits if we only utilize isotropic meshes. This problem is easily understood considering e.g. singularities in elliptic equations aroused by reentrant edges. The solution on an element K touching such a

*Correspondence to: Thomas Richter, Institut für Angewandte Mathematik, INF 294, Universität Heidelberg, 69120 Heidelberg, Germany.
†E-mail: richter@uni-hd.de, thomas.richter@iwr.uni-heidelberg.de

singularity behaves like $u \sim r^{\alpha}$ with $|\nabla^k u| \sim r^{\alpha-k}$. The polynomial interpolation $i_h u$ of degree p exhibits the interpolation error

$$\|\nabla(u - i_h u)\|_K \sim h^{p-\beta} \|\nabla^{p+1-\beta} u\|_K$$

with a $\beta \in [0, p]$. For 90° reentrant edges, we get $\alpha = \frac{3}{4}$ and the error on an element K_E at the edge E with mesh size h_E is given by

$$\|\nabla(u - i_h u)\|_{K_E}^2 \sim h_E^{2p-2\beta} \int_{K_E} r^{3/2-2p-2+2\beta} \, \mathrm{d}x \sim h_E^{5/2}$$

The accumulated error of all the $N \sim h_E^{-1}$ elements along this edge sums up to $\|\nabla e\|_\Omega \sim h_E^{3/4}$. Without considering additional error and elements within the domain we get a lower bound of the approximation order independent of the polynomial degree:

$$\|\nabla e\|_\Omega \geqslant N^{-3/4}$$

The reason for this bad result is the enormous dissipation of elements along the singularity. A fine mesh size h_E is only necessary in the radial direction. Using anisotropically stretched elements, $O(1)$ elements along the edge would be sufficient for the same approximation quality.

The transition to an anisotropic finite element method (FEM) brings along several difficulties. We will be using a hierarchical mesh structure based on hexahedra with local splitting of elements instead of remeshing. To make anisotropic refinement possible, we allow for real bisection of elements into two, four or eight new elements. Data structures for the handling of three-dimensional meshes are very complex, especially the concept of hanging nodes has to be significantly extended. We will give details on our realization of anisotropic meshes in Section 2. Using this structure we cannot resolve arbitrary anisotropies within the domain. However, it is well suited for boundary layers and anisotropies due to non-smooth geometry and all anisotropies aligned with the coarse mesh elements. Further, this structure easily allows the use of efficient geometric multigrid methods to solve the arising linear systems.

Using anisotropic meshes for the FEM, approximation and stability issues have to be analyzed. Apel [6] introduced anisotropic interpolants and has derived interpolation estimates valid on a large class of possible meshes. With these estimates most of the error analysis for finite elements can be directly transferred to anisotropic meshes.

Considering the Navier–Stokes equations, stability problems can arise. Not all inf–sup stable finite elements will also be stable on arbitrary anisotropic meshes. See [7, 8] for an overview of the anisotropic stability of different elements. We will be using the local projection method [9] for stabilizing the inf–sup condition as well as for stabilizing the convective term. With equal order finite elements for both pressure and velocity, stability is gained by adding fluctuations of the discrete functions with respect to a coarse finite element space. This discretization of the Navier–Stokes equations is described in Section 6.

Focus of this work is the development of automatic anisotropic mesh refinement methods. The usual approach for an *a posteriori* identification of anisotropies is the heuristic approach to analyze the Hessian of the solution. The eigenvectors show the dominant directions of the error, the eigenvalues the optimal stretching of the elements. This approach goes back to interpolation estimates on anisotropic elements like [10] and is now well established for anisotropic mesh generation, see [11–14]. While yielding good results in several applications, the approach is firmly linked to the local minimization of interpolation errors. These are not necessarily local bounds

DOI: 10.1002/fld

for the approximation error and if at all they can only give hints on energy norm errors. In many applications, certain functional values of the solution are investigated. With the DWR method [1] the error in functionals of the solution can be estimated. In [15] and [16] a combination of both, the DWR method with Hessian-based anisotropy recovery, is examined for discontinuous Galerkin methods. While the error is estimated with the DWR method and this method is also utilized for picking elements for refinement, the direction of the refinement is determined using the Hessian-based approach. After deciding which elements are to be refined, recovery techniques for higher derivatives of the primal and adjoint solution based either on projection errors or on jump evaluations are used for determining the optimal element stretching and alignment. Although this method uses adjoint-based error estimation, the correct balancing of primal and adjoint information for the extraction of the anisotropy is not given. The authors of [17, 18] use a similar approach. Here the error is estimated with the DWR approach and the directional information is gained by solving a local optimization problem involving the Hessian of the adjoint solution.

All these methods do not yield a optimal balancing of primal and adjoint anisotropy information. Either, the anisotropy is determined by analyzing the primal solution or the adjoint solution is given by the functional, but not both.

Aim of this paper is to present a unified approach for *a posteriori* error estimation including the detection of anisotropies. Instead of splitting the process of error estimation and anisotropy detection, we will directly estimate directional errors using the DWR method.

In the subsequent section we will introduce the FEM with anisotropic meshes and some notations. Differing from the standard FEM on locally refined meshes, we will allow for more general meshes. The third section is a short recapitulation of the DWR method on isotropic meshes. The central fourth section contains the main results and is devoted to the DWR method on anisotropic meshes. We will introduce a new concept for the detection of anisotropic solution patterns and adaptation of the meshes. Then, we will consider several numerical examples to demonstrate the application of this method. Section 5 describes basic test cases to study various effects of the proposed method. In the final sixth section, we describe the stabilized anisotropic discretization of the Navier–Stokes equations. Finally, the proposed method is applied to a well-known Navier–Stokes benchmark problem.

2. FINITE ELEMENTS ON ANISOTROPIC MESHES

In the following we are solving systems of partial differential equations given in the weak formulation. On a polygonal domain $\Omega \subset \mathbb{R}^d$ with $d=2,3$, find $u:\Omega \to \mathbb{R}^c$, $c \in \mathbb{N}$, with $u \in V$

$$A(u)(\phi)=0 \quad \forall \phi \in V \tag{1}$$

where $A(\cdot)(\cdot)$ is a semi-/bi-linear form, linear in the second argument and usually $V=[H_0^1(\Omega)]^c$. For simplicity, we only treat the case of homogeneous Dirichlet data $u=0$ on the boundary $\partial\Omega$ of Ω.

The finite element solution u_h of (1) is given in a subspace $V_h \subset V$, a space of piecewise polynomial functions defined on a triangulation Ω_h of Ω.

2.1. Anisotropic finite element meshes

Key to the anisotropic FEM is the implementation of the mesh structure. We use a hierarchical approach based on quadrilaterals in two and hexahedra in three dimensions. Starting with a coarse

DOI: 10.1002/fld

mesh, finer grids are realized by refinement of elements: a number of mesh elements is split into smaller ones. The coarse mesh has to fulfill certain regularity conditions for finite element meshes:

M1: The domain Ω is split into open elements $\Omega_h = \{K\}$ with

$$\Omega = \bigcup_{K \in \Omega_h} \bar{K}$$

For $K \neq K'$, we have $K \cap K' = \emptyset$ and $\bar{K} \cap \bar{K}'$ is either empty, or a common face, or a common line or a common point of both elements.

M2: For every element $K \in \Omega_h$, there exists a transformation $T_K : (0,1)^d \mapsto K$ with $\det(T_K)(\hat{x}) > 0$ for all $\hat{x} \in (0,1)^d$. Usually T_K will be a polynomial of the same degree as the finite element space (iso-parametric FEM). The transformation T_K can be split into

$$T_K := \mathscr{R}_K \circ \mathscr{S}c_K \circ \mathscr{S}h_K \circ \mathscr{P}_K$$

where $\mathscr{R}_K$ is a rotation and translation, $\mathscr{S}c_K$ an (anisotropic) scaling, $\mathscr{S}h_K$ a shearing and $\mathscr{P}_K$ the nonlinear part of the transformation. We further demand (uniform for all $K \in \Omega_h$) that the shearing and nonlinear part of the transformation are bounded.

M3: The jump in anisotropy between adjacent elements is limited by a constant $\kappa \in \mathbb{R}$. Further, every mesh will have a *patch structure*: each 2^d neighboring elements result from the isotropic refinement (full bisection) of one common father.

Remark 1

1. [M1] is the standard regularity assumption for finite element meshes. For allowing local mesh refinement we will introduce hanging nodes.
2. Assumption [M2] ensures shape regularity of the finite elements. By reducing the allowed shearing and nonlinearity, we especially get a minimum and maximum interior angle condition. However, we still allow for arbitrary anisotropies.
3. Combining [M2] and [M3] we get sufficient conditions for anisotropic local interpolation estimates [6, 19]. The exact definition of κ will be given in Section 4.1.
4. If a mesh fulfills these three requirements, every refined mesh will also comply with [M1] and [M2]. For [M1] this is obvious. It can be shown that the impact of the nonlinear part $\mathscr{P}_K$ decreases exponentially with mesh refinement [19]. The shearing part of the transformation will not change with refinement.

Usual hierarchical finite element meshes allow the splitting of a quad into four and a hex into eight hexahedra. Here, we allow for more general bisection of elements. See Figure 1 for possibilities of element refinement. With local mesh refinement we introduce the concept of hanging nodes: Two adjacent elements can have a different refinement level. A quad or hex can have nodes, which are on the midpoints of adjacent lines or quadrilaterals. These nodes will not be degrees of freedom (DOF), instead their value is replaced by interpolations from the neighboring points. We do not allow *recursive hanging nodes*:

M4: No hanging line or quad may depend on another hanging node.

DOI: 10.1002/fld

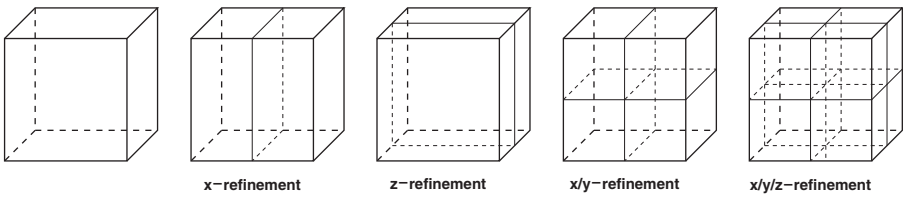

Figure 1. Some possibilities for refinement of one elements in three spacial dimensions.

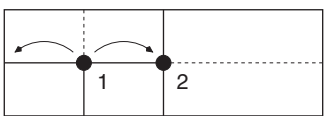

Figure 2. Recursive hanging nodes.

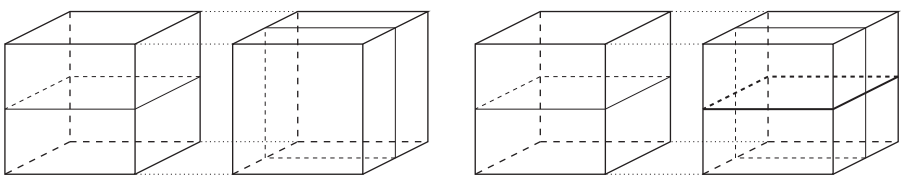

Figure 3. Two hexahedra with opposite anisotropic refinement (left). Resolved by adding further refinement (right).

In Figure 2 a situation of such a *recursive hanging node* is shown. Node 1 is hanging, but the interpolation would depend on the hanging node 2. Although resolving this case with interpolation applied in the correct sequence would be possible, we do not allow for this setting. Instead, we change the mesh by introducing an additional refinement (one of the two dotted lines). Finally, we will bar one special type of anisotropic refinement in three dimensions.

M5: No quad may belong to two hexahedra with opposite anisotropic refinement.

In Figure 3 that case is shown. The bordering quad for the two hexahedra in the left picture is not valid. To resolve this situation we have to add one additional refinement in one of the two hexahedra.

Using an *a posteriori* error estimator, we will pick elements for refinement and determine the optimal bisection direction for every element. This will be described in Section 4.2. Before carrying out the refinement we have to assure that the new mesh will comply with the mesh conditions [M4] and [M5]. This is done by modifying the selection of elements to refine. Usually we pick additional elements for refinement or we add more directions to an already chosen element.

DOI: 10.1002/fld

Algorithm 1: Anisotropic mesh adaptation

(1) Estimate directional error indicators η_K^x, η_K^y and average η_{avg} over all K
(2a) Flag element K for refinement in direction x, if 'error in direction x is large'
(2b) Flag element K for refinement in direction y, if 'error in direction y is large'
(3) While mesh is not regular
  (a) Resolve double hanging nodes by adding refinement directions
  (b) Resolve antipodally refined quadrilateral by adding refinement directions
(4) Refine the mesh

Remark 2

1. Step (2) describes the actual adaptation method. We split element K in direction x, if the directional error η_K^x of this element is larger than a given threshold. Since we check each element for every direction separately, this can result in anisotropic or isotropic refinement. Section 4.4 will give details.
2. For resolving the problem of interdependent hanging nodes, we try to remove the hanging node on the element that is more coarse. This is usually the element containing the hanging node, which is the last in the dependency chain. To resolve this problem in Figure 2, we would flag the large element on the right side for a splitting y-direction. This process leads to some over-refinement. In numerical examples in three spatial dimensions, we observe about 10% additional refinement due to mesh regularity.
3. For resolving antipodally refined quadrilaterals between two hexahedra, we have to add the second refinement direction to one of the hexahedra. See the right sketch in Figure 3.

2.2. Finite element spaces

The finite element spaces on the anisotropic meshes are defined in the usual iso-parametric manner. For two spatial dimensions using [M2], we have with $r \in \mathbb{N}$

$$V_h = \{v \in V : v_{|K} = T_K(\phi), \phi \in \text{span}\{x^i y^j : i, j = 0, \ldots, r\}\}$$

The finite element solution $u_h \in V_h$ is given by

$$A(u_h)(\phi_h) = 0 \quad \forall \phi_h \in V_h$$

and for a arbitrary function $v \in V$ we define the Ritz projection $\mathscr{R}_h v$ by

$$\mathscr{R}_h v: \quad A(\mathscr{R}_h v)(\phi) = A(v)(\phi) \quad \forall \phi \in V_h \tag{2}$$

3. THE DWR METHOD

In this section we will briefly present the DWR method and present the main results from [1], important for this work. Let $J : V \to \mathbb{R}$ be a three times directionally differentiable functional and $A(\cdot)(\cdot) : X := V \times V \to \mathbb{R}$ a three times directionally differentiable form, linear in the second argument. With $x := (u, z) \in X$, we define the functional $L : X \to \mathbb{R}$ as

$$L(x) := J(u) - A(u)(z)$$

DOI: 10.1002/fld

The stationary point $L'(x)(y)=0$ for all $y \in X$ indicates the solution $u \in V$ of the primal problem and $z \in V$ of the dual (or adjoint) problem:

$$\begin{aligned} A(u)(\phi) &= 0 \quad \forall \phi \in V \\ A'(u)(\phi, z) &= J'(u)(\phi) \quad \forall \phi \in V \end{aligned} \tag{3}$$

In [1] we find the following substantial result:

Theorem 1 (*DWR Method*)
For the Galerkin approximations $x \in X = V \times V$ and $x_h \in X_h = V_h \times V_h$ of the variational problem (3), we have the *a posteriori* error representation

$$J(u) - J(u_h) = \frac{1}{2} \min_{y_h \in X_h} L'(x_h)(x - y_h) + R^{(3)}(x - x_h)$$

where the remainder term $R^{(3)}(x - x_h)$ is given with $e := x - x_h$ as

$$R^{(3)}(e) := \frac{1}{2} \int_0^1 L'''(x_h + se)(e, e, e) s(s-1)\, \mathrm{d}s \tag{4}$$

The remainder term vanishes, if $A(\cdot)(\cdot)$ is bilinear and if the functional $J(\cdot)$ is quadratic. We will discuss the behavior of this term for specific applications.

For the actual evaluation of the DWR error estimator, we have to approximate the primal and dual interpolation errors (called the *weights* in this context).

A higher-order reconstruction of the discrete solutions u_h and z_h has been found to deliver the best results. We introduce the space $V_{2h}^{(2r)}$ of piece-wise polynomials of double degree on the mesh with double grid spacing. With the interpolation operator $i_{2h}^{(r)} : V_h \to V_{2h}^{(2r)}$ into this space, the interpolation error is approximated by

$$u - i_h u \approx i_{2h}^{(2r)} u_h - u_h \tag{5}$$

This method requires the patch structured mesh [M3], where every element is part of a set of four (or eight in three-dimensional) elements, which result from global refinement. Other possibilities for the evaluation of the weights are described in [19].

Theorem 1 holds on anisotropic meshes without modification. The remainder term can be estimated by anisotropic *a priori* estimates and is of higher order. However, this representation of the functional error cannot be used to identify the directions for refinement. Considering anisotropic finite elements, the DWR method is usable for error estimation; however, it does not yield a mechanism to generate optimal anisotropic meshes.

In the following section we will derive directional error estimators by a splitting of the DWR estimator. This allows for a uniform approach of error estimation and anisotropy detection.

4. ANISOTROPIC FINITE ELEMENT SPACES

For the following analysis we introduce partially discretized function spaces. For simplicity of the presentation, we restrict the discussion to two spatial dimensions. The partially discretized spaces

DOI: 10.1002/fld

V_x and V_y are defined as

$$V_x = \{v \in V:\ v_{|K} = T_K(\phi), \phi \in \text{span}\{x^i \psi(y) : i = 0, \ldots, r, \psi \in H^1((0,1))\}\}$$
$$V_y = \{v \in V:\ v_{|K} = T_K(\phi), \phi \in \text{span}\{\psi(x) y^i : i = 0, \ldots, r,\ \psi \in H^1((0,1))\}\}$$

In three spatial dimensions, the construction is analogous with always one discretized direction versus two directions that are not discretized. Similar to (2) we define for a function $v \in V$ the Ritz projections into the partially discretized spaces

$$\begin{aligned} \mathscr{R}_x v: &\quad A(\mathscr{R}_x v)(\phi) = A(v)(\phi) \quad \forall \phi \in V_x \\ \mathscr{R}_y v: &\quad A(\mathscr{R}_y v)(\phi) = A(v)(\phi) \quad \forall \phi \in V_y \end{aligned} \tag{6}$$

Lemma 1
The following properties hold:

(i) The spaces V_x and V_y are nested

$$V_h \subset \begin{matrix} V_x \\ V_y \end{matrix} \subset V$$

(ii) For functions $v \in V$, the Ritz projections commute

$$\mathscr{R}_h v = \mathscr{R}_x \mathscr{R}_y v = \mathscr{R}_y \mathscr{R}_x v$$

Proof
(i) is a direct consequence of the construction of the anisotropic finite element spaces. With (6) and (i) we have for the projection $\mathscr{R}_x : \mathscr{R}_y v \mapsto \mathscr{R}_x \mathscr{R}_y v$:

$$A(\mathscr{R}_x \mathscr{R}_y v)(\phi) = A(\mathscr{R}_y v)(\phi) = A(v)(\phi) \quad \forall \phi \in V_h \subset V_x$$

On the other hand using (2) we get $\mathscr{R}_h v = \mathscr{R}_x \mathscr{R}_y v$. The same is true for $\mathscr{R}_y \mathscr{R}_x v$ and (ii) holds. □

4.1. Interpolation estimates for anisotropic function spaces

In this paragraph we derive estimates for special interpolation operators $i_x : V \to V_x$ and $i_y : V \to V_y$. The operator is an adaption of ideas in [6]. The considerations in this paragraph are limited to meshes of tensor–product type. Extensions to more general meshes are technical, see [6] for an overview. By $\hat{K}$ we denote the patch of elements that share a common side with K. See Figure 4 for the construction of $\hat{K}$. We demonstrate the construction of the interpolation operator in x-direction, that is for the operator $i_x : V \to V_x$. First we assume that $u \in C(\hat{K})$ is smooth. Let χ^i for $i = 0, \ldots, r$ be the Lagrange points in x-direction of element K. In fact these points are lines in y-direction through the element K, see the three dotted thick lines in Figure 4 (here shown for the case $r = 2$). Now, let $\phi^i(x)$ be the one-dimensional nodal basis functions (polynomial of degree r) in K with $\phi^i(\chi^j) = \delta_{ij}$ for $i, j = 0, \ldots, r$.

Then, the nodal interpolation operator for smooth functions is given by

$$\mathscr{N}_x u(x, y)_{|K} = \sum_{i=0}^{r} \phi^i(x) u(\chi^i, y) \tag{7}$$

and we have $\mathscr{N}_x u(\chi^i, y) = u(\chi^i, y)$ for all $(\chi^i, y) \in \bar{K}$.

DOI: 10.1002/fld

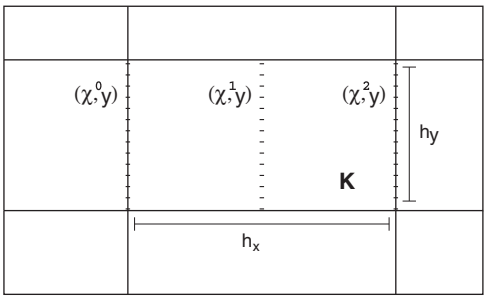

Figure 4. Patch of elements $\hat{K}$.

If we cannot access node values (due to limited regularity), we define the following averaging anisotropic interpolation operator i_x:

$$i_x u(x,y)_{|K} = \sum_{i=0}^{r} \phi^i(x)\gamma^i(y) \tag{8}$$

where the values (functions in y) $\gamma^i(y)$ are averages of u in the surrounding of the interpolation points (χ^i, y). For this averaging we use a variation of the Scott and Zhang interpolation operator [20]. An excellent overview of the extension of averaging interpolation operators to anisotropic meshes is given in [6]. Here we use a modification suitable for a partial interpolation.

The values $\gamma^i(y)$ are defined as averages of u over lines σ^i in x-direction. The line σ^i touches the basis point χ^i and is entirely in one element $K' \in \hat{K}$. For basis points in the middle of the elements (like χ^1 in Figure 4), we have $\sigma^i \subset K$ and σ^i is the line of width h_x from the left to the right part of the element K. The choice of σ^i is not unique for basis points on the (left and right) boundaries of the element K (e.g. for χ^0 in Figure 4). Here we demand that for every (χ^0, y) in $\bar{K}$, the line σ^0 is the same, in particular it does not depend on y. If the choice is not unique, we choose for σ^i the line touching χ^i through the element $K' \in \hat{K}$ with the largest mesh spacing h'_x. Notice that the choice of σ^i depends on the Lagrange point χ^i only, not on the regarded element $K \ni (\chi^i, y)$. For points $(\chi^i, y) \in \overline{K} \cap \overline{K}'$, there is just one line σ^i either in element K or in element K'. Thus the resulting interpolant (8) is continuous in the whole domain Ω.

With this definition, let $\{\psi^i\}$ be polynomials of degree r defined by

$$\int_{\sigma^i} \psi^i(s)\phi^j(s)\,\mathrm{d}s = \delta_{ij}, \quad i,j = 0,\ldots,r \tag{9}$$

We note from [6] that

$$h_x \leqslant |\sigma^i| \leqslant \kappa h_x, \quad \|\psi^i\|_{L^\infty(\sigma^i)} \sim |\sigma^i|^{-1} \leqslant h_x^{-1} \tag{10}$$

where κ is the maximum local change of anisotropy within the triangulation Ω_h:

$$\kappa = \max_{\hat{K} \in \Omega_h} \left\{ \max_{K,K' \in \hat{K}} \left\{ \frac{\alpha_K}{\alpha_{K'}} \right\} : \alpha_K = \max\left\{ \frac{h_{K,x}}{h_{K,y}}, \frac{h_{K,y}}{h_{K,x}} \right\} \right\}$$

DOI: 10.1002/fld

Now we define the averages $\gamma^i(y)$ in (8) as

$$\gamma^i(y) := \int_{\sigma^i} u(s,y)\psi^i(s)\,\mathrm{d}s$$

Usually the optimal choice of these averaging lines σ^i is crucial for obtaining stable anisotropic interpolation operators. Here, the situation is different (in fact more easy) since we interpolate into an only partially discretized space. For i_x the averaging lines σ^i are always aligned in x-direction, for i_y the lines are aligned in y-direction. The following stability result holds:

Lemma 2 (*Stability of the interpolation operator i_x*)
For a function u with sufficient regularity, it holds on meshes of tensor product type

$$\|\partial_x\partial_y^l i_x u\|_K \leqslant c\|\partial_x\partial_y^l u\|_{\hat{K}} + c\kappa^{1/2}h_y\|\partial_x\partial_y^{l+1}u\|_{\hat{K}}, \quad l\in\mathbb{N}_0$$

$$\|\partial_y^l i_x u\|_K \leqslant c\|\partial_y^l u\|_{\hat{K}}, \quad l\in\mathbb{N}_0$$

Proof
We start with the second estimate. Since σ^i does not depend on y, we get from (8)

$$\begin{aligned}
\|\partial_y^l i_x u\|_K^2 &\leqslant \sum_{i=0}^{r}\int_0^{h_x}\int_0^{h_y}\left(\phi^i(x)\int_{\sigma^i}\partial_y^l u(s,y)\psi^i(s)\,\mathrm{d}s\right)^2\mathrm{d}y\,\mathrm{d}x \\
&\leqslant \sum_{i=0}^{r}\|\phi^i\|_{L^\infty(K)}^2\|\psi^i\|_{L^\infty(\sigma^i)}^2\int_0^{h_x}\mathrm{d}x\int_0^{h_y}|\sigma^i|\int_{\sigma^i}(\partial_y^l u(s,y))^2\,\mathrm{d}s\,\mathrm{d}y \\
&\leqslant c\sum_{i=0}^{r}|\sigma^i|^{-1}h_x\|\partial_y^l u\|_{\hat{K}}^2 \leqslant c\|\partial_y^l u\|_{\hat{K}}^2
\end{aligned} \tag{11}$$

The case of the in-line derivative is more complicated. We will introduce a function $w\in V_x$ to be specified later, but satisfying $\partial_x w = 0$; thus, $w(x,y)=w(y)$ on $\hat{K}$. With the trace theorem, we have

$$\begin{aligned}
\|\partial_x\partial_y^l u\|_K^2 &\leqslant \sum_{i=0}^{r}\int_0^{h_x}\int_0^{h_y}\left(\partial_x\phi^i(x)\int_{\sigma^i}(\partial_y^l u(s,y)-\partial_y^l w(y))\psi^i(s)\,\mathrm{d}s\right)^2\mathrm{d}y\,\mathrm{d}x \\
&\leqslant c|K|\sum_{i=0}^{r}\left(\|\partial_x\phi^i\|_{L^\infty(K)}^2\|\psi^i\|_{L^\infty(\sigma^i)}^2\sup_{0\leqslant y\leqslant h_y}\left(\int_{\sigma^i}|\partial_y^l(u(s,y)-w(y))|\,\mathrm{d}s\right)^2\right) \\
&\leqslant ch_x^{-2}\kappa^{-1}(\|\partial_y^l(u-w)\|_{\hat{K}}^2 + h_x^2\|\partial_x\partial_y^l(u-w)\|_{\hat{K}}^2 + h_y^2\|\partial_y^{l+1}(u-w)\|_{\hat{K}}^2)
\end{aligned} \tag{12}$$

The results of the trace inequality will be analyzed separately. The choice of

$$w(y) = \frac{1}{|\sigma|}\int_\sigma u(s,y)\,\mathrm{d}s$$

leads with Poincare's inequality to

$$\|\partial_y^l(u-w)\|_{\hat{K}} \leqslant ch_x\|\partial_x\partial_y^l u\|_{\hat{K}}, \quad \|\partial_x\partial_y^l(u-w)\|_{\hat{K}} = \|\partial_x\partial_y^l u\|_{\hat{K}} \tag{13}$$

DOI: 10.1002/fld

In addition with

$$(u-w)(x,y)=u(x,y)-\frac{1}{|\sigma|}\int_\sigma u(s,y)\,\mathrm{d}s=\frac{1}{\sigma}\int_\sigma\int_s^x \partial_x u(t,y)\,\mathrm{d}t\,\mathrm{d}s$$

it follows that

$$\|\partial_y^{l+1}(u-w)\|_{\hat{K}}\leqslant\kappa h_x\|\partial_x\partial_y^{l+1}u\|_{\hat{K}} \tag{14}$$

Combining (11)–(14) we get the result. On elements K touching the homogeneous Dirichlet boundary, the argument holds with the choice $w=0$. □

Lemma 3 (*Estimates for the nodal interpolation operator* $\mathcal{N}_x$)
Assuming sufficient regularity of $u\in V$, the following estimates hold on tensor product type meshes:

$$\|\partial_y^l(u-\mathcal{N}_x u)\|_K\leqslant h_x^{r+1}\|\partial_x^{r+1}\partial_y^l u\|_K,\quad l\in\mathbb{N}_0$$

$$\|\partial_x\partial_y^l(u-\mathcal{N}_x u)\|_K\leqslant h_x^{r}\|\partial_x^{r+1}\partial_y^l u\|_K,\quad l\in\mathbb{N}_0$$

Proof
The interpolation error $\Phi(x,y):=(u-\mathcal{N}_x u)(x,y)$ has $r+1$ zeros for every y. Owing to the special anisotropic character, this is also true for all y-derivatives of the interpolation error:

$$\frac{\partial^l}{\partial^l y}\Phi(\chi^i,y)=0,\quad i=0,\ldots,r,\ \ l\in\mathbb{N}_0$$

The estimates now follow as direct application of the one-dimensional L^2 and H^1 interpolation estimates applied to Φ and $\partial_y\Phi$. □

Combining the stability result from Lemma 2 with Lemma 3 we get interpolation estimates for the operator i_x by standard arguments.

Lemma 4 (*Interpolation estimates for the operator* i_x)
Assuming sufficient regularity of a function $u\in V$, the following interpolation estimates hold on tensor product-type meshes:

$$\|u-i_x u\|_K\leqslant ch_x^{r+1}\|\partial_x u\|_{\hat{K}}$$

$$\|\partial_x(u-i_x u)\|_K\leqslant ch_x^{r}\|\partial_x^{r+1}u\|_{\hat{K}}+c\kappa^{1/2}h_x^r h_y\|\partial_x^{r+1}\partial_y u\|_{\hat{K}}$$

$$\|\partial_y^l(u-i_x u)\|_K\leqslant ch_x^{r+1}\|\partial_x^{r+1}\partial_y^l u\|_{\hat{K}},\quad l\in\mathbb{N}$$

These estimates exhibit the properties of the partially discretized anisotropic function spaces. Since the estimate in the cross direction (e.g. the y-derivative of the x-interpolation error) yields a higher order, a separation of the error into the coordinate directions is possible. We note that for functions $v_x\in V_x$, it holds that $v_x=i_x v_x=\mathcal{N}_x v_x$. Further, the interpolation operators i_x and i_y commute for all functions $v\in V$:

$$i_h v:=i_x i_y v=i_y i_x v$$

Finally, we will state a special interpolation estimate of importance for anisotropic error estimation.

DOI: 10.1002/fld

Lemma 5 (*Product interpolation estimate*)
Assuming sufficient regularity of a function $u \in V$, the following estimates hold on tensor product-type meshes for h_x and h_y small enough:

$$\|u + i_h u - i_x u - i_y u\| = \|(\mathrm{id} - i_x)(\mathrm{id} - i_y)u\| \leqslant c h_x^{r+1} h_y^{r+1} \|\partial_x^{r+1} \partial_y^{r+1} u\|_{\hat{K}}$$

$$\|\nabla(u + i_h u - i_x u - i_y u)\| = \|\nabla(\mathrm{id} - i_x)(\mathrm{id} - i_y)u\| \leqslant c\kappa h_x^r h_y^r (h_x + h_y) \|\partial_x^{r+1} \partial_y^{r+1} u\|_{\hat{K}}$$

Proof
We will prove the second estimate. Introducing $\Phi := (\mathrm{id} - i_y)u$, Lemma 4 yields

$$\|\nabla(\Phi - i_x \Phi)\|_K \leqslant c h_x^r (\|\partial_x^{r+1} \Phi\|_{\hat{K}} + (h_x + \kappa^{1/2} h_y) \|\partial_x^{r+1} \partial_y \Phi\|_{\hat{K}})$$

Lemma 4 together with the stability results for higher derivatives in the cross-direction in Lemma 2 applied to the right-hand side of this estimate gives

$$\|\nabla(\mathrm{id} - i_x)(\mathrm{id} - i_y)u\|_K \leqslant c h_x^r h_y^r (h_x + h_y + \kappa^{1/2} h_y + \kappa^{1/2} h_x + h_x h_y \kappa) \|\partial_x^{r+1} \partial_y^{r+1} u\|_{\hat{K}}$$

Neglecting the higher-order term, the estimate follows for h_x and h_y small enough. The first estimate holds by similar arguments. □

4.2. Anisotropic DWR

In this section we present a splitting of the functional error into directional errors

$$J(u) - J(u_h) \sim J(u) - J(u_x) + J(u) - J(u_y)$$

Then, a modification of the DWR framework allows for the estimation of these two directional errors. We state the central result.

Theorem 2 (*Anisotropic a posteriori error estimation*)
For the Galerkin approximations $x \in X$ and $x_h \in X_h$ of (3), the following *a posteriori* error estimate holds:

$$\begin{aligned} J(u) - J(u_h) &= \{J(u) - J(u_x)\} + \{J(u) - J(u_y)\} + R \\ &= \frac{1}{2} \left\{ \min_{y_x \in X_x} L'(x_h)(x - y_x) + \min_{y_y \in X_y} L'(x_h)(x - y_y) \right\} + R \end{aligned} \tag{15}$$

where the dominant part of the remainder term R is given by $R \sim R_1 + R_x + R_y$ with

$$R_1 := L'(x_h)((\mathrm{id} - i_x)(\mathrm{id} - i_y)x)$$

$$R_x := L''(x_h)(x - i_x x, x_x - x_h)$$

$$R_y := L''(x_h)(x - i_y x, x_y - x_h)$$

For the proof of Theorem 2 we give two technical lemmas. For the ease of presentation we will from now on denote by $R^{(3)}$ terms, which consist of the third derivatives of $L(\cdot)$ and which are thus of third order in the arguments. If only one argument is given this appears thrice (like in (4)); otherwise, we indicate all arguments. First, we give an estimate concerning the splitting of the isotropic error in the functional $J(u) - J(u_h)$ into directional errors.

DOI: 10.1002/fld

Lemma 6 (*Splitting into directional errors*)
For the Galerkin approximations $x \in X$, $x_h \in X_h$, $x_x \in X_x$ and $x_y \in X_y$ of (3), we get the following splitting of the error:

$$J(u) - J(u_h) = (J(u) - J(u_x)) + (J(u) - J(u_y)) + R$$

where the dominant part of the remainder R is given by

$$R \sim R_1 + R_x + R_y$$

with

$$R_1 := L'(x_h)((\mathrm{id} - i_x)(\mathrm{id} - i_y)x), \quad R_x := L''(x_h)(x - i_x x, x_x - x_h)$$
$$R_y := L''(x_h)(x - i_y x, x_y - x_h)$$

Proof
The remainder term R will be written as

$$R = (J(u) - J(u_h)) - (J(u) - J(u_x)) - (J(u) - J(u_y))$$

Applying Theorem 1 three times we get with arbitrary $y_h \in X_h$, $y_y \in X_y$ and $y_x \in X_x$:

$$R = \tfrac{1}{2}L'(x_h)(x - y_h) - \tfrac{1}{2}L'(x_x)(x - y_x) - \tfrac{1}{2}L'(x_y)(x - y_y) + R_1^{(3)} \tag{16}$$

with the third-order remainders from Theorem 1

$$R_1^{(3)} := R^{(3)}(x - x_h) + R^{(3)}(x - x_x) + R^{(3)}(x - x_y)$$

With the choice $y_h := i_h x$, $y_x := i_x x$ and $y_y := i_y x$ in (16) and by introducing $\pm\frac{1}{2}L'(x_h)(x - i_x x)$ and $\pm\frac{1}{2}L'(x_h)(x - i_y x)$ we have

$$R = \frac{1}{2}\underbrace{L'(x_h)(i_x x + i_y x - x - i_h x)}_{=R_1} + \frac{1}{2}\int_0^1 L''(x_x + s(x_h - x_x))(x - i_x x, x_h - x_x)\,\mathrm{d}s$$
$$+ \frac{1}{2}\int_0^1 L''(x_y + s(x_h - x_y))(x - i_y x, x_h - x_y)\,\mathrm{d}s + R_1^{(3)}$$

Approximating the integrals with the box-rule at $s = 1$, we conclude with

$$R = \tfrac{1}{2}(R_1 - R_x - R_y) + R^{(3)}(x - i_x x, x_h - x_x, x_h - x_x) + R^{(3)}(x - i_y x, x_h - x_y, x_h - x_y) + R_1^{(3)} \quad \square$$

The role of the remainder terms R_1, R_x, R_y has to be analyzed separately for every equation and will be discussed in Section 4.3. This lemma allows for a splitting of the functional error $J(u) - J(u_h)$ into the directional errors $J(u) - J(u_x)$ and $J(u) - J(u_y)$ in the coordinate directions. Instead of simply applying Theorem 1 to estimate these directional errors, we introduce an additional approximation to avoid the non-computable values x_x and x_y that would appear in the error estimate.

DOI: 10.1002/fld

Lemma 7 (*A posteriori estimate for directional error*)
For the Galerkin approximations $x \in X$ and $x_x \in X_x$ of (3), the following *a posteriori* estimate holds:

$$J(u) - J(u_x) = \frac{1}{2} \min_{y_x \in X_x} L'(x_h)(x - y_x) + R$$

where the dominant part of the remainder term R is given by $R \sim R_x$, with R_x as described in Lemma 6.

Proof
Theorem 1 yields

$$J(u) - J(u_x) = \frac{1}{2} \min_{y_x \in X_x} L'(x_x)(x - y_x) + R^{(3)}(x - x_x)$$

Similar to the proof of Lemma 6

$$\begin{aligned} L'(x_x)(x - y_x) &= L'(x_h)(x - y_x) + \int_0^1 L''(x_h + s(x_x - x_h))(x - y_x, x_x - x_h)\, \mathrm{d}s \\ &= L'(x_h)(x - y_x) + L''(x_h)(x - y_x, x_x - x_h) + R^{(3)}(x - y_x, x_h - x_x, x_h - x_x) \end{aligned}$$

With the choice $y_x = i_x x$ the middle term turns to be R_x^x, which concludes the proof. □

Combining Lemmas 6 and 7 for both directional error $J(u) - J(u_x)$ and $J(u) - J(u_y)$, the central result Theorem 2 is proven. The remainder terms appearing are all third-order terms $R^{(3)}$ or given by R_x, R_y or R_1.

4.3. Discussion of the remainder terms

Here we discuss the remainder terms R_1, R_x and R_y as well as the third-order remainders $R^{(3)}$ arising in the process of splitting and estimating the error in detail. This discussion has to be accomplished separately for every equation. Prototypically for elliptic PDE's we discuss Poisson's equation with linear finite elements and homogeneous Dirichlet boundary values:

$$(\nabla u, \nabla \phi) = (f, \phi) \quad \forall \phi \in V$$

We consider the functional

$$J(u) = \|u\|_2^2$$

Assume that f and the domain Ω are of sufficient regularity to apply all interpolation estimates from Section 4.1. Remarks on other (nonlinear) equations will be given where necessary. For showing higher order of the remainders, $O(h^3)$ is necessary. Starting with R_1, we introduce the notation $\Psi(x) := (\mathrm{id} - i_x)(\mathrm{id} - i_y)x$ and with integration by parts obtain

$$\begin{aligned} |R_1| &\leqslant |(\nabla u_h, \nabla \Psi(z))| + |(\nabla z_h, \nabla \Psi(u))| + |(f, \Psi(u))| + 2|(u_h, \Psi(u))| \\ &\leqslant \|\Delta u_h\| \|\Psi(z)\| + \|\Delta z_h\| \|\Psi(u)\| + (\|f\| + 2\|u_h\|) \|\Psi(u)\| \end{aligned}$$

DOI: 10.1002/fld

The first two terms arise from the main part of the differential equation, the third part is the remainder from the right-hand side and the final one arises from the functional evaluation. The boundary terms from integration by parts vanish, since $\Psi(x)=0$ on the edges. With Lemma 5 we conclude

$$|R_1| \leqslant c h_x^2 h_y^2 (\|\Delta u_h\| + \|\Delta z_h\|)(\|\partial_x^2 \partial_y^2 u\| + \|\partial_x^2 \partial_y^2 z\|) = O(h^4)$$

Other equations lead to similar remainders. The worst case includes estimates for $\|\nabla\Psi(x)\| = O(h^3)$ still of higher order. For example for the Navier–Stokes equations, we get remainders of the type

$$|(u_h \cdot \nabla\Psi(u), z_h)| \leqslant \|u_h\|_\infty \|z_h\| \|\nabla\Psi(u)\| = O(h^3)$$

Next, we will exemplarily discuss R_x for Poisson's equation

$$R_x = (\nabla(u - i_x u), \nabla(z_x - z_h)) + (\nabla(z - i_x z), \nabla(u_x - u_h)) + 2(u - i_x u, u - i_x u)$$

Here, the first two terms arise from differentiating the weak formulation of the Poisson equation, the last term is the second derivative of the functional. Owing to linearity, the right-hand side f has no influence in L''. A straightforward estimation of the functional term with Lemma 4 yields

$$2|(u - i_x u, u - i_x u)| = 2\|u - i_x u\|^2 \leqslant 2c h_x^4 \|\partial_x^2 u\|^2 = O(h^4)$$

The two other terms are more difficult to estimate. Simply applying Cauchy–Schwartz inequality does not give the correct order

$$|(\nabla(u - i_x u), \nabla(z_x - z_h))| \leqslant \|\nabla(u - i_x u)\| \|\nabla(z_x - z_h)\| = O(h^2) \tag{17}$$

since it is the product of an H^1 interpolation and an H^1 approximation error. By splitting

$$(\nabla(u - i_x u), \nabla(z_x - z_h)) = \underbrace{(\partial_x(u - i_x u), \partial_x(z_x - z_h))}_{=:R_x^x} + \underbrace{(\partial_y(u - i_x u), \partial_y(z_x - z_h))}_{=:R_x^y} \tag{18}$$

we can estimate the second part R_x^y using the higher-order interpolation result for the cross-derivative in Lemma 4 by

$$|R_x^y| = |(\partial_y(u - i_x u), \partial_y(z_x - z_h))| \leqslant \|\partial_y(u - i_x u)\| \|\partial_y(z_x - z_h)\| \leqslant c h_x^2 \|\partial_x^2 \partial_y y\| \|\nabla(z_x - z_h)\| = O(h^3)$$

For estimating R_x^x we consult results from [21] shown for linear finite elements:

$$(\nabla(u - i_h u), \nabla v_h) \leqslant h^2 \|u\|_3 \|\nabla v_h\| \quad \forall v_h \in V_h$$

A similar result can be shown for $v_x \in V_x$ and anisotropic interpolation in x-direction:

$$(\nabla(u - i_x u), \nabla v_x) \leqslant h_x^2 \|u\|_3 \|\partial_x v_x\| \leqslant h_x^2 \|u\|_3 \|\nabla v_x\| \quad \forall v_x \in V_x \tag{19}$$

DOI: 10.1002/fld

On parallelograms K this term is even zero. Let K be a rectangle with sides aligned to the coordinate system. Then, integration by parts yields

$$(\partial_x(u-i_x u), \partial_x v_x)_K = -(u-i_x u, \partial_{xx} v_x) + <u-i_x u, \partial_n v_x>_{\partial K^x} = 0$$

where ∂K^x are the left and right sides of the element. Since v_x is linear in x-direction, it holds that $v_{xx}=0$. Further, the interpolation error is zero on the left and right boundaries. This (stronger) result can be generalized to parallelograms, if we split the derivatives (18) not in x- and y-direction, but in the dominant directions given by the element K. Lemma 1 (ii) now demonstrates that $z_x - z_h \in V_x$

$$z_x - z_h = \mathscr{R}_x(z-z_h) =: v_x \in V_x$$

and with $z_x - z_h \in V_x$ we can apply (19) to obtain at least $|R_x^x| = O(h^3)$.

Terms not of second order, e.g. arising from $(u_h \cdot \nabla u_h, \phi_h)$ result in products of L^2 and H^1 (or L^2) interpolation and approximation errors. Here, $|R_x| = O(h^3)$ is gained with the standard estimate (17).

If a special character of the equation or the functional does not allow for a higher-order estimation of R_x and R_y, these terms can be included into the error estimator. The next section will show a way to numerically approximate the remainder terms.

For nonlinear equations the third-order remainder terms will be present, such as for the Navier–Stokes equations we have

$$R^{(3)}(x-x_x) = 2((u-u_x)\cdot\nabla(u-u_x), z-z_x)$$

This product can be estimated as the product of one H^1, a L^2 and a L^∞ approximation error. *A priori* estimates then yield higher order.

4.4. Localization of the error estimator and mesh adaptation

Similar to (5) in Section 3 we introduce a local interpolation operator to approximate the interpolation errors $x - i_x x$ and $x - i_y x$ by a higher-order reconstruction. We exemplarily demonstrate this easy strategy for the x-direction. By $V_x^{(2r,2h)}$, we define the space of piece-wise polynomials of degree r on a mesh of size h and degree $2r$ in direction x with mesh size $2h$. The spaces V_h and $V_x^{(2r,2h)}$ share the same DOF. Thus, we can define the nodal basis functions $\{\phi_h^i\}$ of V_h and $\{\phi_x^i\}$ of $V_x^{(2r,2h)}$ with $\phi_h^j(\chi_i) = \phi_x^j(\chi_i) = \delta_{ij}$ in the supporting points χ_i for $i, j = 1, \ldots, N$. This allows us to define the discrete interpolation operator $i_x^{(2r,2h)} : V_h \to V_x^{(2r,2h)}$ applied to $x_h \in V_h$ (see Figure 5

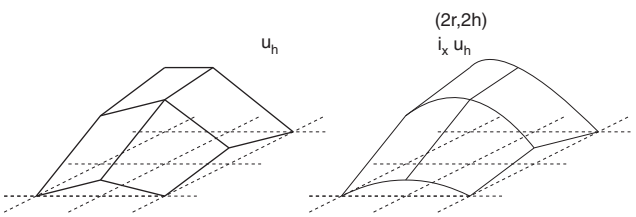

Figure 5. Interpolation to fine space $V_x^{(2r,2h)}$.

DOI: 10.1002/fld

for an example):

$$i_x^{(2r,2h)} x_h = i_x^{(2r,2h)} \sum x^i \phi_h^i = \sum x^i \phi_x^i \tag{20}$$

Now we treat $i_x^{(2r,2h)} x_h - x_h$ as an approximation to $x - i_x x$. The *a posteriori* error estimator from Theorem 2 is evaluated as

$$J(u) - J(u_h) \approx \underbrace{\tfrac{1}{2} L'(x_h)(i_x^{(2r,2h)} x_h - x_h)}_{=:\eta_x} + \underbrace{\tfrac{1}{2} L'(x_h)(i_y^{(2r,2h)} x_h - x_h)}_{=:\eta_y} \tag{21}$$

If we want to include the remainder terms R_x and R_y in error estimation, we approximate

$$R_x \approx L''(x_h)(i_x^{(2r,2h)} x_h - x_h, i_y^{(2r,2h)} x_h - x_h) \approx R_y$$

since we can treat $i_y^{(2r,2h)} x_h$ as a higher-order approximation to x_x.

For mesh adaptation, we localize the estimators η_x and η_y to values η_x^i, η_y^i in the DOF. Using (20) we can evaluate the directional error η_x in (21) as the product

$$\eta_x = \sum_i \underbrace{\overbrace{\frac{1}{2} L'(x_h)(\phi_x^i - \phi_h^i)}^{=:L_x^i} x_h^i}_{=:\eta_x^i} = \langle (L_x), (x_h) \rangle$$

It is well known that a simple restriction of residual terms to elements does not result in the correct order. This is also true for the node-wise local values η_x^i. In general we have $\sum_i |\eta_x^i| \gg |\sum_i \eta_x^i| = |\eta_x|$, and the local values η_x^i are not suitable for mesh adaptation since they would result in over refinement.

Let $V_x^{(2h)}$ be the space of piece-wise polynomials of degree r in all directions with mesh size $2h$ in direction x and h in the other directions. (This space is coarser in x-direction than V_h). We show an example of the interpolation $i_x^{(2h)} : V_x \to V_x^{(2h)}$ into this coarse space in Figure 6. By choosing for y_x in (15) not $y_x = i_x x$ but $y_x := i_x x - i_x^{(2h)}(x - i_x x)$ followed by the approximation of the interpolation error $(x - i_x x) \approx (i_x^{(2r,2h)} x_h - x_h)$, the error estimator η_x features the correct

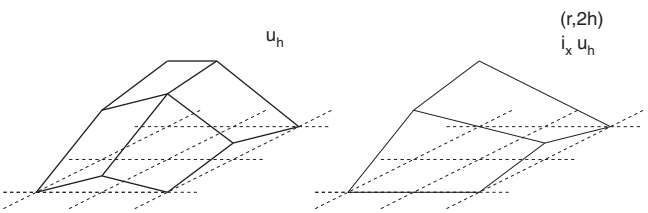

Figure 6. Interpolation to coarse space $V_x^{(2h)}$.

DOI: 10.1002/fld

local order. We then evaluate η_x as

$$\eta_x = \tfrac{1}{2} L'(x_h)((i_x^{(2r,2h)} - \mathrm{id})(x_h - i_x^{(2h)} x_h)) = \langle (L_x), (x_h - i_x^{(2h)} x_h) \rangle$$

With this localization of the error estimator, the mesh refinement algorithm is as follows: (replacing Step (2) in Algorithm 1)

Algorithm 2: Anisotropic mesh refinement

(1) Estimate directional error indicators (η_x^i) and (η_y^i)
(2) Calculate the mean of the absolute values $\bar{\eta} = \frac{1}{2N} \sum_{i=1}^{N} |\eta_x^i| + |\eta_y^i|$
(3a) If $|\eta_x^i| > \alpha\bar{\eta}$ refine elements adjacent to i in x-direction.
(3b) If $|\eta_y^i| > \alpha\bar{\eta}$ refine elements adjacent to i in y-direction.

The parameter α in the last step describes the locality of the refinement, not the threshold for anisotropy. There are no parameters controlling the amount of anisotropy. The decision to refine an element in one direction does not depend on the local error in the other directions. This strategy aims at balancing the error in all elements and in all directions. Usually, we choose $\alpha \in [\frac{1}{4}, 4]$. Choices of $\alpha < 1$ lead to global refinement if the error is uniformly distributed over the mesh. Larger values of α result in a sharper local refinement. If the problem and thus the error indicators are symmetric, the resulting meshes will also be symmetric.

5. NUMERICAL EXAMPLES

In this section we will analyze the qualities of the meshes generated by the anisotropic error estimator. First, we will illustrate the difference between the proposed adaptation method and standard methods based on the recovery of derivatives for the identification of the anisotropy using a specially constructed test case. Further we will analyze a simple three-dimensional convection–diffusion problem showing the potential of anisotropic mesh refinement.

5.1. A pathological test case

For a better understanding of the difficulties related with output-oriented anisotropic mesh adaptation, we consider the convection–diffusion equation

$$\begin{aligned} -\frac{1}{200}\Delta u + u_x &= 0 \quad \text{on } \Omega = (0,1)^2 \\ u &= \begin{cases} g(y), & (x,y) \in \Gamma_1 \\ 0, & (x,y) \in \partial\Omega / \Gamma_1 \end{cases} \end{aligned} \tag{22}$$

The boundary function $g(y)$ on $\Gamma_1 = \{(0,y) : 0 \leqslant y \leqslant 1\}$ is a smooth regularization of $g(y) = 1$ with $g(0) = g(1) = 0$:

$$g(y) = \tanh(25y(1-y)) = \frac{1 - \mathrm{e}^{-50y(1-y)}}{1 + \mathrm{e}^{-50y(1+y)}}$$

DOI: 10.1002/fld

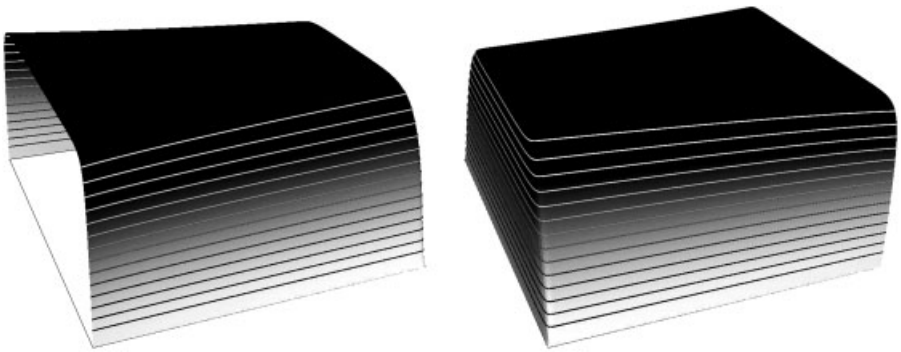

Figure 7. Sketch of primal (left) and adjoint (right) solution the first test case, Section 5.1.

As quantity of interest we evaluate the functional

$$J(u) = \int_{\Gamma_2} \partial_x u(0, y) \cdot g(y)\, \mathrm{d}y$$

where Γ_2 is the opposite side of Γ_1. The weak formulation of (22) and the corresponding adjoint equation are given by

$$u \in u^0 + V : \frac{1}{200} \int_\Omega \nabla u \cdot \nabla \phi \, \mathrm{d}x + \int_\Omega u_x \cdot \phi \, \mathrm{d}x = 0 \quad \forall \phi \in V, \quad u^0_{|\Gamma_1} = g(y), \quad u^0_{|\Gamma/\Gamma_1} = 0$$

$$z \in z^0 + V : \frac{1}{200} \int_\Omega \nabla z \cdot \nabla \phi \, \mathrm{d}x + \int_\Omega z \cdot \phi_x \, \mathrm{d}x = 0 \quad \forall \phi \in V, \quad z^0_{|\Gamma_2} = g(y), \quad z^0_{|\Gamma/\Gamma_2} = 0$$

The strong formulation of the adjoint equation reads $-0.005\Delta z - z_x = 0$, with $z = g$ on Γ_2. Thus we have $u(x, y) = z(1-x, y)$. In Figure 7 a sketch of the primal and adjoint solution is shown.

Neglecting the boundary perturbation due to the smoothing function, thus using $g = 1$, the primal solution would be given by

$$u_{g=1}(x, y) = \frac{1 - \mathrm{e}^{-200(1-x)}}{1 - \mathrm{e}^{-200}}$$

and features a strong boundary layer at Γ_2. Thus, the adjoint solution has this boundary layer at the opposite boundary Γ_1. Common sense would suggest anisotropic refinement along the y-axes at these two boundaries. Apart from the boundary layers, the primal and adjoint solutions are constant, equal to one; the residual of the primal equation will be close to zero where the adjoint solution has a boundary layer and vice versa. Including zero boundary values on the two remaining boundary parts, the solution gets disturbed in the lower and upper part of the domain with (less steep) boundary layers along the remaining two boundaries. These new layers are present in the primal and dual solution.

DOI: 10.1002/fld

In the following we will compare different methods for anisotropic adaption. The set of elements to be refined is always identified using the standard DWR method. How to refine every element will be decided using the following strategies:

[D] Dual Hessian: Analyze Hessian matrix of z. Try to minimize interpolation error $|z-i_h z|_K$ on every element K.

$$\begin{cases} h_x^2|z_{xx}|_K > \beta h_y^2|z_{yy}|_K & \text{refine in } x\text{-direction} \\ h_x^2|z_{xx}|_K < \beta^{-1} h_y^2|z_{yy}|_K & \text{refine in } y\text{-direction} \\ & \text{otherwise refine isotropically} \end{cases}$$

[M] Mixed Hessians: Analyze the product of the interpolation errors of the primal and dual solution in the energy norm $|\nabla(u-i_h u)\cdot\nabla(z-i_h z)|_K$ on every element K.

$$\begin{cases} h_x^2|u_{xx}|_K|z_{xx}|_K > \beta h_y^2|u_{yy}|_K|z_{yy}|_K & \text{refine in } x\text{-direction} \\ h_x^2|u_{xx}|_K|z_{xx}|_K < \beta^{-1} h_y^2|u_{yy}|_K|z_{yy}|_K & \text{refine in } y\text{-direction} \\ & \text{otherwise refine isotropically} \end{cases}$$

[A] Anisotropic DWR: Use Algorithm 2 in Section 4.4.

Remark 3

1. The first method is the standard approach for anisotropic refinement. Usually the Hessian of the primal or dual solution is assembled by recovery techniques and decomposed. The eigenvectors indicate the dominant directions, the eigenvalues the optimal stretching. Our simplification applies to meshes parallel to the coordinate axes if we only allow for refinement along these axes. This method can be generalized to higher polynomial degrees by using higher derivatives.
2. The second approach tries to extract the anisotropy information using the primal and adjoint solution by analyzing the energy errors of both problems.
3. The parameter β controls the balance of anisotropy. For linear finite elements, we choose the value $\beta=4$. This choice stems from a local optimization consideration: By predicting the error on the element K after refinement, what is the best balance between the number of newly generated elements and the predicted error. Higher values of β yield less anisotropy.
4. The third approach does not need a separate decision of which element to refine and how to refine the element. Using the directional indicators, we can analyze every direction independently. If both conditions are effective, the element will be refined isotropically.

In Figure 8 the error for these three refinement methods along with the error on a globally refined mesh is plotted. Only the anisotropic DWR method results in meshes with a better approximation property than structured meshes.

For comparison, three meshes for the different methods are given in Figure 9. We only show the lower left quart of the meshes $(0, 0.5)^2 \subset \Omega$. The meshes using adaption schemes [M] and [A] are symmetric in both direction. The mesh connected to scheme [D] is only symmetric with regard to the x-axis and with less refinement in the right half of the domain.

DOI: 10.1002/fld

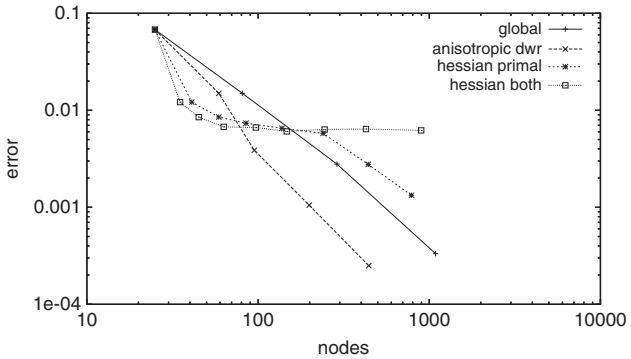

Figure 8. Comparison of different kinds of anisotropic mesh control for the convection diffusion example.

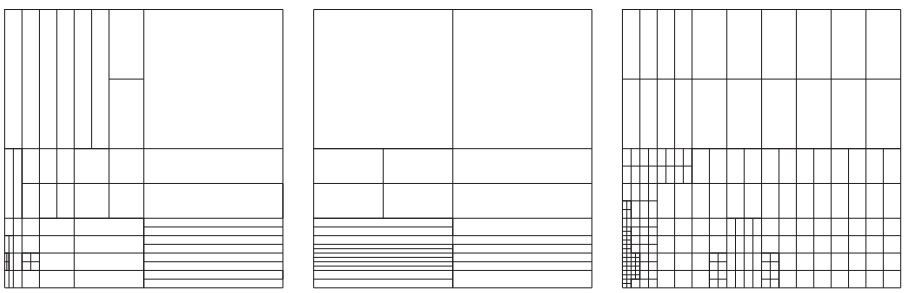

Figure 9. Cutout of the meshes for the three kinds of anisotropic mesh control: [D], [M] and [A].

The left mesh corresponds to the first adaptation strategy. We get anisotropic boundary layers where the dual solution has large derivatives. The plot in the middle shows the mesh using the mixed strategy. Both solutions have large derivatives along the lower and upper boundary. This method resolves only these layers. The (for the functional evaluation) more important left and right boundaries are not resolved, since on both sides either the primal or the adjoint solution is nearly constant and the product does not exhibit a considerably interpolation error. Finally, the mesh using the anisotropic DWR method does not present a substantial anisotropic character. This rather surprising result can only be achieved with the correct weighting of residuals and anisotropy information in the directional estimator.

5.2. Three-dimensional convection–diffusion problem

We consider the convection diffusion equation in a domain containing a three-dimensional wing. In Figure 10(a), we give a sketch of the configuration and the coarse mesh. This is a typical geometry for three-dimensional flow problems. Here, we try to identify the possible savings using anisotropic mesh refinement for this kind of geometry with a simplified equation. For stabilization of the convective term, we use a stream upwind Petrov–Galerkin formulation:

$$(\varepsilon\nabla u, \nabla\phi) + (u_x, \phi) + \sum_{K\in\Omega_h} (\delta_K(u_x - \varepsilon\Delta u), \phi_x) = 0, \quad \varepsilon = 10^{-3}$$

DOI: 10.1002/fld

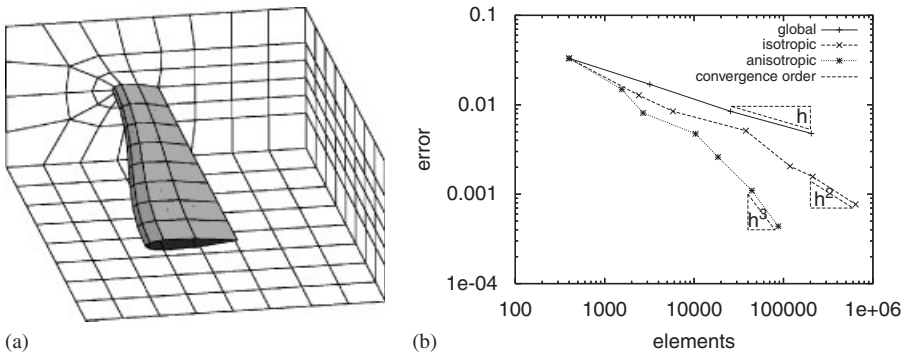

Figure 10. Convection—diffusion problem in a domain with a three-dimensional wing: (a) sketch of geometry and coarse mesh and (b) error for global, for local isotropic and local anisotropic mesh refinement.

Table I. Functional values for the convection–diffusion problem. The reference value is given as $J(u)=0.774\pm10^{-4}$.

Global		Isotropic		Anisotropic	
# elem	$J(u_h)$	# elem	$J(u_h)$	# elem	$J(u_h)$
400	0.8070	400	0.8070	400	0.8070
3 200	0.7909	2416	0.7868	1 557	0.7889
25 600	0.7815	5804	0.7824	2 688	0.7821
204 800	0.7788	38 004	0.7791	10 372	0.7787
		118 420	0.7761	18 524	0.7766
		211 996	0.7753	44 210	0.7751
		641 174	0.7747	87 466	0.7744

where the stabilization parameter δ_K depends on the local Peclet number. The boundary conditions are homogeneous Dirichlet $u=0$ on the wing, $u=1$ on the left boundary (the 'inflow') and homogeneous Neumann boundary on all the other sides. Mesh adaptation aims at minimizing the error in the functional

$$J(u)=\int_{\Gamma_w}\partial_n u\,\mathrm{do}$$

where Γ_w is the boundary of the wing. Extrapolation yields the reference value $J_{\mathrm{ref}}=0.7740\pm10^{-4}$. In Table I we compare the results for global mesh refinement with adaptive isotropic and adaptive anisotropic mesh refinements. Plots of the error are given in Figure 10(b). As should be expected, there is a large benefit of using anisotropic mesh refinement for this problem. The trailing and leading edges of the wing induce singularities (or at least sharp gradients) to the solution and call for local refinement. In z-direction, the solution is very smooth, a large element stretching is sufficient. This can only be achieved by anisotropic mesh refinement. Figure 11 gives a section of the locally refined anisotropic mesh used in the calculations. The grey shaded elements are the

DOI: 10.1002/fld

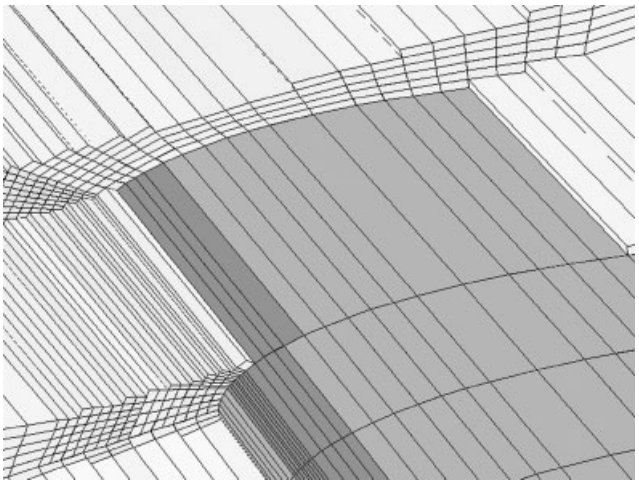

Figure 11. Anisotropic mesh for the convection—diffusion problem.

surface of the wing. The maximum aspect ratio (relation between longest and shortest element stretching) used in the calculation is 1 : 200.

6. ANISOTROPIC NAVIER–STOKES

Finally, we apply the anisotropic adaptation method to a three-dimensional Navier–Stokes benchmark problem. For treating the Navier–Stokes equations

$$A(U)(\Phi) := (\nu \nabla v, \nabla \phi) + (v \cdot \nabla v, \phi) - (p, \nabla \cdot \phi) + (\nabla \cdot v, \xi) \tag{23}$$

we have to deal with stability problems, raised by the inf–sup condition as well as instabilities due to the convective term. We address both issues with the local projection methods [9, 22] by adding a stabilization term $S(U)(\Phi)$ including fluctuations with regard to a coarse space V_H. Introducing the local interpolation operator $i_H : V_h \to V_H$ and $\pi_H := id - i_H$, the local projection method is given by

$$\tilde{A}(U)(\Phi) := A(U)(\Phi) + S(U)(\Phi), \quad S(U)(\Phi) = (\alpha \pi_H(\nabla p_h), \nabla \xi_h) + (\delta \pi_H(\nabla v_h), \nabla \phi_h) \tag{24}$$

with stabilization parameters, α, δ, element-wise depending on the local grid Peclet number

$$Pe_K = |v|_{K,\infty} \frac{h_K}{\nu}$$

In [23] the extension of this stabilization method to anisotropic meshes for the Stokes system has been introduced. The idea is to separately stabilize the partial derivatives using the dominant local mesh directions. In [24] the local projection method has been formulated for the Oseen system and optimal stability and approximation properties have been proven. For the ease of presentation,

DOI: 10.1002/fld

we describe the case of tensor–product meshes aligned with the coordinate axes. Further, let $h_{K,y}<h_{K,x}$. The anisotropic stabilization term is then

$$S(U)(\Phi)=(\alpha^x\pi_H(\partial_x p_h),\partial_x\xi_h)+(\alpha^y\pi_H(\partial_y p_h),\partial_y\xi_h)$$
$$+(\delta^x\pi_H(\partial_x v_h),\partial_x\phi_h)+(\delta^y\pi_H(\partial_y v_h),\partial_y\phi_h) \tag{25}$$

with stabilization parameters chosen as

$$\alpha^{x/y}{}_{|K}=\frac{h^2_{K,x/y}}{v}\min(1,Pe^{-1}_{\min}),\quad \delta^{x/y}{}_{|K}=|v|^2_{K,\infty}\alpha^{x/y}{}_{|K}$$

where $Pe_{\min}$ is the local Peclet number depending on minimum mesh size $h_{K,y}$. In [24] stability and error estimates have been shown in the triple norm

$$|||U|||^2=\nu\|\nabla v\|^2+S(U)(U)=\tilde{A}(U)(U) \tag{26}$$

For linear finite elements with $h_{K,y}<h_{K,x}$ on every element, the anisotropic *a priori* error estimate holds:

$$|||U-U_h|||^2\leqslant\sum_{K\in\Omega_h}\frac{h^4_{K,x}}{h_{K,y}}(C^p_K\|\partial_x\nabla p\|^2_K+C^v_K\|\partial_x\nabla v\|^2_K)$$
$$+h^3_{K,y}(C^p_K\|\partial_y\nabla p\|^2_K+C^v_K\|\partial_y\nabla v\|^2_K) \tag{27}$$

with

$$C^p_K=\min(Pe_{\min},1)\|v\|^{-1}_{K,\infty}\sim\frac{h_{K,y}}{\nu},\quad C^v_K=\nu h^{-1}_{K,y}+\|v\|_{K,\infty}\sim\frac{\nu}{h_{K,y}}$$

This estimate exhibits the factor $h_{\max}h^{-1}_{\min}$ that can be arbitrary large. However, the estimate shows the correct balancing of the terms on appropriate meshes, since this factor is scaled with the small derivatives (in x-direction). The next section gives a numerical test for the stability of this scheme on highly anisotropic meshes.

6.1. Stability of the local projection method on anisotropic meshes

Stability of the local projection stabilization (LPS) method on anisotropic meshes is demonstrated on a variation of the Driven Cavity problem, driven by a forcing term on parts of the domain. On a square of size $\Omega=(0,1)^2$, we prescribe homogeneous Dirichlet boundary values on all parts of the boundary $\partial\Omega$. The right-hand-side for the continuity equations is given as

$$f(x,y)=\begin{pmatrix}f_x(x,y)\\0\end{pmatrix},\quad f_x(x,y)=\begin{cases}1, & (x,y)\in\Omega_1:=\left[\frac{1}{4},\frac{3}{4}\right]\times\left[\frac{3}{4},\frac{15}{16}\right]\\0, & (x,y)\in\Omega/\Omega_1\end{cases}$$

This special driving force is chosen to avoid effects by non-matching Dirichlet boundary conditions. For Reynolds number $Re=100$, we estimate the value of

$$J(U)=\int_\Omega|p|^2+|\nabla v|^2\,\mathrm{d}x \tag{28}$$

DOI: 10.1002/fld

Table II. Effectivity of the anisotropic error estimator for the Driven Cavity test case on sequences of meshes.

# elements	e_h	η_h	η_x	η_y	e_h/η_h	$e_h/(\eta_x+\eta_y)$	a-r
(a) *Isotropic global mesh refinement*							
1024	2.20×10^{-2}	2.13×10^{-2}	1.81×10^{-2}	3.10×10^{-3}	1.032	1.037	1
4096	5.33×10^{-3}	5.34×10^{-3}	4.57×10^{-3}	7.60×10^{-4}	0.998	1.000	1
16 384	1.33×10^{-3}	1.33×10^{-3}	1.14×10^{-3}	1.89×10^{-4}	0.999	0.999	1
65 536	3.32×10^{-4}	3.32×10^{-4}	2.85×10^{-4}	4.70×10^{-5}	0.999	1.000	1
262 144	8.30×10^{-5}	8.30×10^{-5}	7.13×10^{-5}	1.17×10^{-5}	0.999	0.999	1
(b) *Anisotropic mesh refinement in x-direction*							
1024	2.20×10^{-2}	2.13×10^{-2}	1.81×10^{-2}	3.10×10^{-2}	1.032	1.037	4^0
4096	4.26×10^{-3}	4.23×10^{-3}	1.12×10^{-3}	3.08×10^{-2}	1.009	1.016	4^1
16 384	3.19×10^{-3}	3.15×10^{-3}	4.61×10^{-5}	3.08×10^{-2}	1.014	1.022	4^2
65 536	3.13×10^{-3}	3.08×10^{-3}	2.07×10^{-5}	3.08×10^{-2}	1.014	1.022	4^3
262 144	3.12×10^{-3}	3.08×10^{-3}	2.48×10^{-5}	3.08×10^{-2}	1.014	1.022	4^4

using piecewise linear finite elements. This problem does not feature a strong anisotropic character; however, we can study the stability of the scheme with respect to anisotropic meshes. Further, we get a hint of the robustness of the anisotropic DWR method and the splitting of the error into the directional errors with respect to the aspect ratio. In Table II we list the error $e_h = J(u) - J(u_h)$, the estimated errors $\eta_h \sim J(u) - J(u_h)$ as well as the directional estimates $\eta_x \sim J(u) - J(u_x)$ and $\eta_y \sim J(u) - J(u_y)$ for a sequence of refined meshes. We further indicate the effectivity of the isotropic *a posteriori* error estimator e_h/η_h and the effectivity of the splitting $e_h/(\eta_x+\eta_y)$. In Table II(b) we see that even for high anisotropies, the anisotropic error estimator yields excellent results. Further, the approximation quality does not degenerate on highly anisotropic meshes. Since the anisotropic meshes do not suit the solution, the error does not converge to zero; however, the approximation is stable.

6.2. *Three-dimensional Navier–Stokes Flow*

Finally, we apply the anisotropic adaptation method to the benchmark problem 'Laminar Flow Around a Cylinder' [25]. This problem has been extensively studied with different continuous FEMs. In this benchmark, from 1995 the accurate calculation of several functional values for different configurations of the flow around an obstacle was asked for. Here, we analyze a laminar flow with Reynolds number 20 around a square obstacle. Error estimation and mesh adaptation are based on estimating the drag value of this obstacle. In Figure 12 the configuration of this test-case is given.

Owing to the edges of the obstacle, this problem is very difficult. Singularities in the derivatives of the solution foil the advantages of higher order methods, if no optimal local mesh adaptation is used. In [3] the reference value $c_{\text{drag}} = 7.767 \pm 2\cdot10^{-3}$ could be identified using adaptive calculations involving $30\cdot10^6$ DOF on parallel computers. Table III and Figure 13 compare the results of the anisotropic calculation with the locally isotropic refined results taken from [3].

DOI: 10.1002/fld

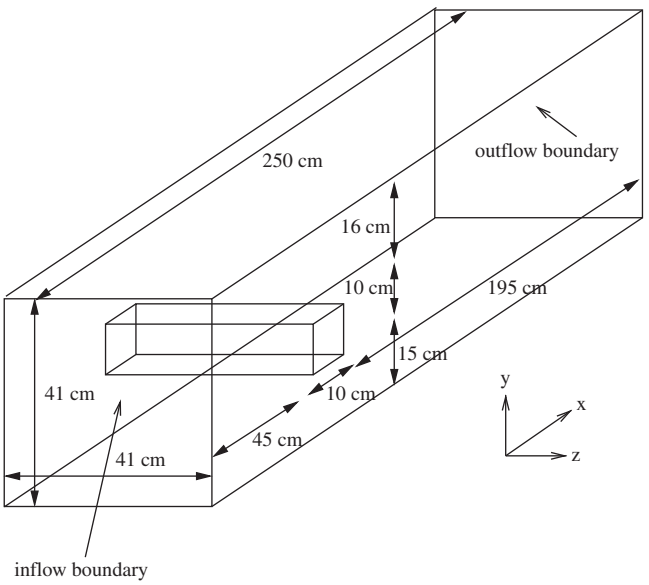

Figure 12. Configuration of the benchmark: laminar flow around a square obstacle.

Table III. Drag coefficient of the quadratic obstacle. Values for global and isotropic local mesh refinement are taken from [3].

Global refinement		Adaptive isotropic		Adaptive anisotropic	
# elements	c_{drag}	#elements	c_{drag}	# elements	c_{drag}
78	13.315	624	8.0445	78	10.674
624	8.0445	2427	7.9742	220	8.0141
4 992	7.9759	7120	7.7881	1344	7.8596
39 936	7.7878	16 808	7.7595	3124	7.7900
319 488	7.7579	54 880	7.7620	5316	7.7787
2 555 904	7.7612	113 128	7.7659	15 948	7.7655

The reference value is $7.767 \pm 2 \cdot 10^{-3}$.

With anisotropic local mesh refinement, a relative error of approximately $2 \cdot 10^{-4}$ is achieved on a mesh with seven times less elements than using isotropic local mesh refinement. The slight differences in the drag values on the coarse mesh are due to small variations in the anisotropic version of the local projection methods. All calculations start with the same initial mesh.

Although this problem does not feature a strong anisotropic behavior at this low Reynolds number, the savings due to anisotropic meshes are considerably compared with an adaptive method using isotropic meshes. Finally, in Figure 14 we show a cut-out of the locally refined mesh close to the obstacle. The maximum aspect ratio in the mesh is about 1 : 20.

DOI: 10.1002/fld

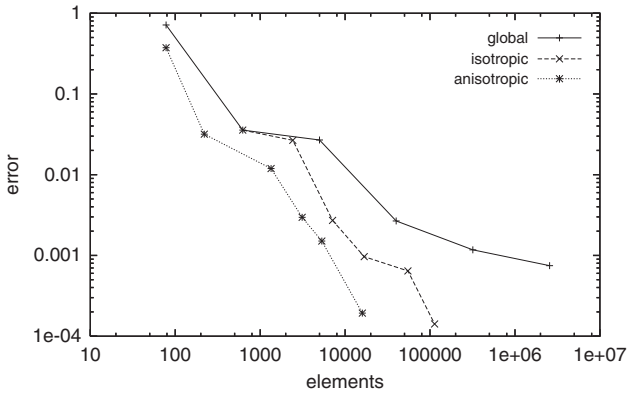

Figure 13. Relative error in the drag coefficient using different refinement methods. The values for global and isotropic local refinement are taken from [3].

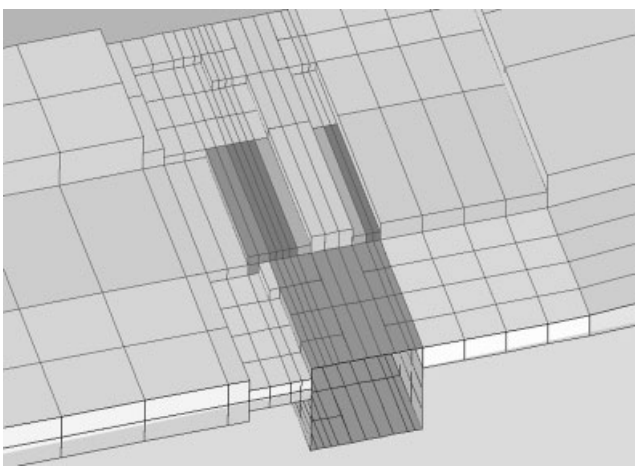

Figure 14. Anisotropic mesh at the obstacle. The maximum aspect ration used is 1 : 20 on the dark shaded elements at the edges.

7. SUMMARY

In this work we have demonstrated a method for automatic anisotropic mesh adaptation. This new approach allows for a unified treatment of mesh refinement and anisotropy detection based on the error in functional outputs. Numerical examples including a three-dimensional flow problem demonstrate the feasibility and efficiency of this method. The essential part of the implementation—here not described in detail—is a three-dimensional mesh structure making anisotropic bisection of hexahedra possible.

This anisotropic adaptation scheme is limited to capturing anisotropies aligned with the elements in the coarse mesh. Anisotropies going diagonally through the elements cannot be captured. The

DOI: 10.1002/fld

extension of this method to general anisotropies is not possible in this framework, since the anisotropic error estimation is based on the partially discretized function spaces. We do not detect where anisotropies are present and in which direction they proceed, we can just estimate directional errors.

General anisotropies cannot be realized with adaptive methods based on hierarchical splitting of quadrilateral or hexahedral elements. A remedy would be to remesh after some initial refinements. Based on higher derivatives of the primal and adjoint solution, a new coarse mesh aligned to all dominant anisotropies can be generated. The correct balancing is then established by restarting the anisotropic adaptation scheme on this adapted coarse mesh.

For all numerical computations, the software library *Gascoigne* [26] was used. The visualization of the meshes is done with *Visual 3* developed by Bob Haimes at MIT.

REFERENCES

1. Becker R, Rannacher R. An optimal control approach to a posteriori error estimation in finite element methods. In *Acta Numerica 2001*, Volume 37, Iserles A (ed.). Cambridge University Press: Cambridge, 2001; 1–225.
2. Becker R. Weighted a posteriori error control in FE methods. *Enumath 95 Proceedings*. World Science Publications: Singapore, 1997.
3. Braack M, Richter T. Solutions of 3D Navier–Stokes benchmark problems with adaptive finite elements. *Computers and Fluids* 2006; **35**(4):372–392.
4. Braack M, Becker R, Rannacher R. Adaptive finite elements for reactive flows. *ENUMATH-97*, *Second European Conference on Numerical Mathematics and Advanced Applications*. World Scientific Publications: Singapore, 1998; 206–213.
5. Becker R, Vexler B. A posteriori error estimation for finite element discretization of parameter identification problems. *Numerische Mathematik* 2004; **96**(3):435–459.
6. Apel T. *Anisotropic Finite Elements*: *Local Estimates and Applications*. Advances in Numerical Mathematics. Teubner: Stuttgart, 1999.
7. Ainsworth M, Coggins P. The stability of mixed hp-finite element methods for stokes flow on high aspect ratio elements. *SIAM Journal on Numerical Analysis* 2001; **38**(5):1721–1761.
8. Apel T, Nicaise S. The inf–sup condition for the Bernardi–Fortin–Raugel element on anisotropic meshes. *Calcolo* 2004; **41**:89–113.
9. Becker R, Braack M. A finite element pressure gradient stabilization for the Stokes equations based on local projections. *Calcolo* 2001; **38**(4):173–199.
10. Barnhill RE, Gregory JA. Interpolation remainder theory from Taylor expansions on triangle. *Numerische Mathematik* 1976; **25**:401–408.
11. D'Azevedo EF, Simpson RB. On optimal interpolation triangle incidences. *SIAM Journal on Scientific and Statistical Computing* 1989; **10**(6):1063–1075.
12. Peraire J, Vahdati M, Morgan K, Zienkiewicz OC. Adaptive remeshing for compressible flow computations. *Journal of Computational Physics* 1987; **72**:449–466.
13. Zienkiewicz OC, Wu J. Automatic directional refinement in adaptive analysis of compressible flows. *International Journal for Numerical Methods in Engineering* 1994; **37**:2189–2210.
14. Castro-Diaz MJ, Hecht F, Mohammadi B, Pironneau O. Anisotropic unstructured mesh adaptation for flow simulations. *International Journal for Numerical Methods in Fluids* 1997; **25**:475–491.
15. Venditti DA, Darmofal DL. Anisotropic grid adaptation for functional outputs: application to two-dimensional viscous flows. *Journal of Computational Physics* 2003; **187**:22–46.
16. Leicht T, Hartmann R. Anisotropic mesh refinement for discontinuous Galerkin methods in two-dimensional aerodynamic flow simulations. *International Journal for Numerical Methods in Fluids* 2007; **56**(11):2111–2138.
17. Formaggia L, Perotto S, Zunino P. An anisotropic a-posteriori error estimate for a convection–diffusion problem. *Computing and Visualization in Science* 2001; **4**:99–2001.
18. Micheletti S, Formaggia L, Perotto S. Anisotropic mesh adaptation in computational fluid dynamics: application to the advection–diffusion–reaction and the Stokes problems. *Applied Numerical Mathematics* 2004; **51**(4):511–533.
19. Richter Th. Parallel multigrid for adaptive finite elements and its application to 3D flow problem. *Ph.D. Thesis*, Universität Heidelberg, 2005.

DOI: 10.1002/fld

20. Scott LR, Zhang S. Finite element interpolation of nonsmooth functions satisfying boundary conditions. *Mathematics of Computation* 1990; **54**(190):483–493.
21. Blum H. Asymptotic error expansion and defect correction in the finite element method. SFB-123 Preprint 640, Habilitationsschrift, Institut für Angewandte Mathematik, Universität Heidelberg, 1991.
22. Braack M, Burman E. Local projection stabilization for the Oseen problem and its interpretation as a variational multiscale method. *SIAM Journal on Numerical Analysis* 2006; **43**(6):2544–2566.
23. Braack M, Richter T. Local projection stabilization for the stokes system on anisotropic quadrilateral meshes. In *Enumath*, de Castro B *et al.* (ed.). Springer: Berlin, 2005; 770–778.
24. Braack M. A stabilized finite element scheme for the Navier–Stokes equations on quadrilateral anisotropic meshes. *M2AN* 2008; **42**:903–924.
25. Schäfer M, Turek S. Benchmark computations of laminar flow around a cylinder (With support by Durst F, Krause E, Rannacher R). In *Flow Simulation with High-performance Computers II. DFG Priority Research Program Results 1993–1995*, Hirschel EH (ed.). Notes on Numerical Fluid Mechanics, vol. 52. Vieweg: Wiesbaden, 1996; 547–566.
26. The finite element toolkit GASCOIGNE. http://www.gascoigne.uni-hd.de.

DOI: 10.1002/fld