

# A primal-dual adaptive finite element method for total variation minimization

Martin Alkämper<sup>1</sup>, Stephan Hilb<sup>2</sup>, Andreas Langer<sup>3\*</sup>

<sup>1</sup>Ratingen, Germany.

<sup>2</sup>Stuttgart, Germany.

<sup>3</sup>Department for Mathematical Sciences, Lund University, Sölvegatan  
18A, Lund, 22100, Sweden.

\*Corresponding author(s). E-mail(s): [andreas.langer@math.lth.se](mailto:andreas.langer@math.lth.se);

## Abstract

Based on previous work we extend a primal-dual semi-smooth Newton method for minimizing a general  $L^1-L^2-TV$  functional over the space of functions of bounded variations by adaptivity in a finite element setting. For automatically generating an adaptive grid we introduce indicators based on a-posteriori error estimates. Further we discuss data interpolation methods on unstructured grids in the context of image processing and present a pixel-based interpolation method. The efficiency of our derived adaptive finite element scheme is demonstrated on image inpainting and the task of computing the optical flow in image sequences. In particular, for optical flow estimation we derive an adaptive finite element coarse-to-fine scheme which allows resolving large displacements and speeds-up the computing time significantly.

**Keywords:** A-posteriori error estimate, Adaptive finite element discretization, Non-smooth optimisation, Combined  $L^1/L^2$  data-fidelity, Total variation, Optical flow estimation, Inpainting

**MSC Classification:** 65N50 , 65N30 , 49M29 , 90C25 , 94A08

## 1 Introduction

Let  $\Omega \subset \mathbb{R}^d$ ,  $d \in \mathbb{N}$ , be an open, bounded and simply connected domain with Lipschitz boundary. We consider the problem of minimizing the  $L^1-L^2$ -TV functional [1], which has various applications in image processing and has been subject of study in [2, 3]

and in a finite element context in [4]. More precisely, let  $V \subset H^1(\Omega)^m$ ,  $m \in \mathbb{N}$ , be a continuously embedded Hilbert space which is weakly closed in  $H^1(\Omega)^m$  and  $L^2(\Omega)^m$ , then as in [5] we consider the  $L^1$ - $L^2$ -TV functional in the following form

$$\inf_{\vec{u} \in V} \left\{ \alpha_1 \int_{\Omega} \varphi_{\gamma_1}(|T\vec{u} - g|) \, d\vec{x} + \frac{\alpha_2}{2} \|T\vec{u} - g\|_{L^2}^2 + \frac{\beta}{2} \|S\vec{u}\|_{V_S}^2 + \lambda \int_{\Omega} \varphi_{\gamma_2}(|\nabla \vec{u}|_F) \, d\vec{x} =: E(\vec{u}) \right\}, \quad (1.1)$$

where  $S : V \rightarrow V_S$  is a bounded linear operator for some Hilbert space  $V_S$ ,  $T : L^2(\Omega)^m \rightarrow L^2(\Omega)$  is a bounded linear operator,  $g \in L^2(\Omega)$  is some given data,  $|\cdot|_F : \mathbb{R}^{d \times m} \mapsto \mathbb{R}$  denotes the Frobenius norm, and  $\alpha_1, \alpha_2, \lambda, \beta \geq 0$  are adjustable weighting parameters. Further  $\gamma_1, \gamma_2 \geq 0$  are Huber-regularization parameters associated to the *Huber-function*  $\varphi_\gamma : \mathbb{R} \rightarrow [0, \infty)$  for  $\gamma \geq 0$  defined by

$$\varphi_\gamma(x) := \begin{cases} \frac{1}{2\gamma}x^2 & \text{if } |x| \leq \gamma, \\ |x| - \frac{\gamma}{2} & \text{if } |x| > \gamma. \end{cases} \quad (1.2)$$

We remark that  $d$  denotes the spatial dimension, e.g. one sets  $d = 1$  for signals and  $d = 2$  for images, and  $m$  describes the number of channels, i.e. for grey-scale images  $m = 1$  and for motion fields we have  $m = d$ . In [5] a primal-dual framework around (1.1) has been developed and a finite element semi-smooth Newton scheme established. The present work aims to extend [5] to an adaptive finite element setting. Note that every finite dimensional subspace of a normed vector space is closed [6, Corollary 5.34] and convex and hence also weakly closed [7, Corollary 8.74]. Thus  $V$  in (1.1) includes any arbitrary finite element space. Moreover, we note that the finite element semi-smooth Newton scheme in [5] is also derived for this space setting.

## Related work

While mathematical image processing techniques are often presented or discretized in a finite difference setting due to simple handling of image array data, finite element methods in image processing have also been studied, see [8] for an overview. The ability to adaptively refine or coarsen unstructured finite element meshes seems promising in order to save computational resources during processing. We refer the reader to [9, 10] for an overview of adaptive finite element methods. We note that images usually have homogeneous regions as well as parts with a lot of details. It seems intuitively sufficient to discretize finely in regions of transitions and fine details, while a coarse discretization should do well in uniform parts. To this end error estimates are crucial in order to correctly guide the refinement process. First attempts for using adaptive finite element methods in image processing (restoration) have been made for non-linear diffusion models, which are mainly used for image denoising and edge detection. In particular, in [11–13] adaptive finite element methods are proposed for the modified Perona-Malik model [14]. Based on primal-dual gap error estimators adaptive finite element methods for the  $L^2$ -TV model, i.e. (1.1) with  $\alpha_1 = \beta = \gamma_2 = 0$ , are proposed in [15–17]. While in [15] the dual problem is discretized by continuous,

piecewise affine finite elements, in [17] the Brezzi-Douglas-Morini finite elements [18] are used. Additionally, [16] introduces an adaptive finite element framework based on Crouzeix–Raviart elements. Note that in [15–17] only clean images are considered and no image reconstruction or analysis problem is tackled. In a similar way relying on a semi-smooth Newton method for minimizing the  $L^2$ -TV model over the Hilbert space  $H_0^1(\Omega)$  in [19] an adaptive finite element method, again using a posteriori error estimates, is suggested for image denoising only. Adaptive finite element methods for optical flow computation have been presented in [20, 21]. However, these methods differ significantly from our approaches. Firstly, a different minimization problem is considered. Secondly, the mesh in the methods in [20, 21] is initialized at fine image resolution and iteratively coarsened only after a costly computation, while our approach allows to start with a coarse mesh and refines cells only if deemed necessary.

## On this work

In the present work we extend the finite element framework of [5] to an adaptive finite element setting and demonstrate its efficiency by considering the applications of inpainting and motion estimation. The proposed adaptive finite element method is based on a-posteriori error estimates. In particular, we derive two different a-posteriori error estimators, one using the residual and the other using a primal-dual approach. In order to obtain a primal-dual gap error estimate we use the same approach as in [17]. However, due to the presence of Huber regularization terms and potentially non-local operators  $T$  in (1.1) we need to adjust arguments of [17]. For deriving the residual-based error estimates we do not consider a variation inequality setting as in [19] but utilize the strong convexity of the functional in (1.1). Further, while the approach in [19] only considers the image denoising case, i.e.  $T$  being the identity, our estimate allows for general linear and bounded operators  $T$ . Utilizing our derived a-posteriori estimators allows us to propose an adaptive finite element scheme for solving (1.1) for general bounded and linear operators  $T$ . Note that implementing global operators on an unstructured grid is generally considered impractical due to the inefficiency caused by the non-local access pattern. Therefore, we restrict our numerical experiments in this paper to local operators  $T$ .

We demonstrate the usability of the adaptive scheme by solving the task of image inpainting and for estimating motion in image sequences. For both applications an adaptive coarse-to-fine scheme which makes use of the developed adaptive finite element framework is used. That is, the algorithm starts with a coarse mesh, typically coarser than the observed image, and adaptively refines the mesh at parts where deemed necessary. In the setting of optical flow computation this means that we incorporate into [5, Algorithm 3] an adaptive coarse-to-fine warping scheme which is able to resolve large displacements. In particular, due to the adaptive coarse-to-fine warping scheme the proposed algorithm improves [5, Algorithm 3] in two directions. Firstly, it produces more accurate approximations of the optical flow and secondly, it significantly reduces the computing time, see Section 4 below.

Note that for finite elements in general an interpolation method needs to be employed to transfer given data onto a finite element function. The choice of this method is especially relevant when the underlying grid is coarser than the original

image resolution since in that case e.g. nodal sampling for standard linear Lagrange finite elements can lead to aliasing artifacts. In this vein, we propose a new pixel-adapted method to interpolate image data onto finite element meshes and compare it to other alternatives. This new method is designed to minimize the discrete  $\ell^2$ -distance to the original image on the original image grid. That is, it actually optimizes the peak signal-to-noise ratio (PSNR) and therefore might be especially suited for images.

The rest of the paper is organized as follows: In Section 2 we fix the basic notation and terminology used and describe the primal-dual setting of (1.1). Based on this primal-dual formulation we introduce in Section 3 our finite element setting. The main part of this section is devoted to the derivative of a-posteriori error estimates to establish an adaptive finite element algorithm for (1.1). Moreover, we describe several different image interpolation methods and compare them with a newly proposed pixel-adapted interpolation method. Based on the developed adaptive finite element method in Section 4 adaptive coarse-to-fine schemes for image inpainting and motion estimation are proposed. Numerical experiments demonstrate the applicability of the proposed adaptive discretization method.

## 2 Preliminaries

### 2.1 Notation and Terminology

For a Hilbert space  $H$  we denote by  $\langle \cdot, \cdot \rangle_H$  its inner product and by  $\|\cdot\|_H$  its induced norm. The duality pairing between  $H$  and its continuous dual space  $H^*$  is written as  $\langle \cdot, \cdot \rangle_{H,H^*}$ . In the case  $H = L^2(\Omega)^n$ ,  $n \in \mathbb{N}$ , we define the associated inner product by  $\langle \cdot, \cdot \rangle_H : ((u_k)_{k=1}^n, (v_k)_{k=1}^n) \mapsto \sum_{k=1}^n \langle u_k, v_k \rangle_{L^2(\Omega)}$ , where  $\langle \cdot, \cdot \rangle_{L^2(\Omega)}$  is the standard  $L^2$  inner product. Apart from notational convenience, we treat a matrix-valued space  $L^2(\Omega)^{d \times m}$  as equivalent to  $L^2(\Omega)^{dm}$  using the numbering  $(i, j) \mapsto (j-1)d + i$ ,  $i \in \{1, \dots, d\}$ ,  $j \in \{1, \dots, m\}$  of the respective components. Moreover, for any  $L^2$  function space we may use the inner product shorthand notations  $\langle \cdot, \cdot \rangle_{L^2} := \langle \cdot, \cdot \rangle_H$  and similarly  $\|\cdot\|_{L^2} := \|\cdot\|_H$  for the norm. In the sequel, the dual space  $H^*$  of a Hilbert space  $H$  will often be identified with itself.

For a bounded linear operator  $\Lambda \in \mathcal{L}(X, Y)$ , where  $X, Y$  are Hilbert spaces, we define by  $\|\Lambda\| := \|\Lambda\|_{\mathcal{L}(X, Y)}$  its operator norm and denote by  $\Lambda^* : Y^* \rightarrow X^*$  its adjoint operator. In the sequel by single strokes, i.e.  $|\cdot|$ , we describe the Euclidean norm on  $\mathbb{R}^n$ ,  $n \in \mathbb{N}$ . Often operations are applied in a pointwise sense, such that for a vector-valued function  $\vec{u} : \Omega \rightarrow \mathbb{R}^m$  the expression  $|\vec{u}|$  denotes the function  $|\vec{u}| : \Omega \rightarrow \mathbb{R}$ ,  $\vec{x} \mapsto |\vec{u}(\vec{x})|$ .

A function  $f : H \rightarrow \overline{\mathbb{R}} := \mathbb{R} \cup \{+\infty\}$  is called *coercive*, if for any sequence  $(\vec{v}_n)_{n \in \mathbb{N}} \subset H$  we have

$$\|\vec{v}_n\|_H \rightarrow \infty \implies f(\vec{v}_n) \rightarrow \infty.$$

A bilinear form  $a : H \times H \rightarrow \mathbb{R}$  is called *coercive*, if there exists a constant  $c > 0$  such that  $a(\vec{v}, \vec{v}) \geq c\|\vec{v}\|_H^2$  for all  $\vec{v} \in H$ . The subdifferential for a convex function

$f : X \rightarrow \overline{\mathbb{R}} := \mathbb{R} \cup \{+\infty\}$  at  $\vec{v} \in X$  is defined as the set valued function

$$\partial f(\vec{v}) = \begin{cases} \emptyset & \text{if } f(\vec{v}) = \infty, \\ \{\vec{v}^* \in X^* : \langle \vec{v}^*, \vec{u} - \vec{v} \rangle_{X^*, X} + f(\vec{v}) \leq f(\vec{u}) \forall \vec{u} \in X\} & \text{otherwise.} \end{cases}$$

Further for a predicate  $w : \Omega \rightarrow \{\text{true}, \text{false}\}$  we define the indicator  $\chi_w \in \overline{\mathbb{R}}$  as

$$\chi_w := \begin{cases} 0 & \text{if } w(\vec{x}) \text{ is true for almost every } \vec{x} \in \Omega, \\ \infty & \text{else.} \end{cases}$$

For example, let  $u \in L^2(\Omega)$ , then  $\chi_{|u| \leq 1}$  would evaluate to  $\infty$  if and only if  $|u|$  is greater than 1 on a set of non-zero measure.

## 2.2 Primal-Dual Formulation

Let us recall the functional setting and the primal-dual formulation of [5], on which our derived a-posteriori estimates and hence our proposed adaptive finite element scheme, presented in Section 3 below, are based. We restrict ourselves for the operator  $S$  in (1.1) and its related spaces to the following two settings:

- (S.i)  $S := I : V \rightarrow V_S$  with  $\|\cdot\|_V := \|\cdot\|_{L^2}$  and  $V_S = L^2(\Omega)^m$ ,  $\|\cdot\|_{V_S} := \|\cdot\|_{L^2}$   
or  
(S.ii)  $S := \nabla : V \rightarrow V_S$  with  $\|\cdot\|_V := \|\cdot\|_{H^1}$  and  $V_S = L^2(\Omega)^{d \times m}$  (the boundedness of  $S$  follows due to  $\|\nabla \vec{v}\|_{L^2} \leq \|\vec{v}\|_{H^1}$ ),  $\|\cdot\|_{V_S} := \|\cdot\|_{L^2}$ .

We define the symmetric bilinear form  $a_B : V \times V \rightarrow \mathbb{R}$  by

$$a_B(\vec{u}, \vec{w}) := \alpha_2 \langle T\vec{u}, T\vec{w} \rangle_{L^2} + \beta \langle S\vec{u}, S\vec{w} \rangle_{L^2} = \langle B\vec{u}, \vec{w} \rangle_{V^*, V}, \quad (2.1)$$

whereby  $B := \alpha_2 T^* T + \beta S^* S : V \rightarrow V^*$ . The bilinear form  $a_B$  induces a respective energy norm defined by

$$\|\vec{u}\|_B^2 := a_B(\vec{u}, \vec{u}) \quad \text{for } \vec{u} \in V.$$

To guarantee the existence of a (unique) solution of (1.1) in the sequel we assume that  $a_B$  is coercive [5, Proposition 3.1]:

- (As1) The bilinear form  $a_B : V \times V \rightarrow \mathbb{R}$  is coercive.

Here the coercivity is to be understood in the following ways according to Setting (S.i) and Setting (S.ii): In Setting (S.i) the coercivity is with respect to the  $L^2$ -norm, i.e.  $a_B$  is  $L^2$ -coercive. For Setting (S.ii) the coercivity is understood with respect to  $\|\cdot\|_V$  (i.e. the  $H^1$ -norm). Since  $V$  is weakly closed in  $H^1(\Omega)^m$  as well as  $L^2(\Omega)^m$ , the existence of a solution in  $V$  is guaranteed in both settings [5, Proposition 2.2]. If  $V$  is finite dimensional, then Assumption (As1) guarantees a solution in the respective finite dimensional space.

Further, the coercivity of  $a_B$  implies the invertibility of  $B$  [5, Remark 2.1]. Accordingly, we define the norm  $\|\vec{u}\|_{B^{-1}}^2 := \langle \vec{u}, B^{-1} \vec{u} \rangle_{V^*, V}$  for  $\vec{u} \in V^*$ . Additionally, we introduce the linear and bounded operator  $\Lambda := (T, \nabla) : V \rightarrow W := L^2(\Omega) \times L^2(\Omega)^{d \times m}$ .

In these settings we recall the following duality result.

**Proposition 2.1** ([5, Theorem 3.1 & Proposition 3.2]). *The pre-dual problem to (1.1) reads*

$$\inf_{\vec{p}=(p_1, \vec{p}_2) \in W^*} \left\{ \frac{1}{2} \left\| \Lambda^* \vec{p} - \alpha_2 T^* g \right\|_{B^{-1}}^2 - \frac{\alpha_2}{2} \|g\|_{L^2}^2 + \langle g, p_1 \rangle_{L^2} + \chi_{|p_1| \leq \alpha_1} + \frac{\gamma_1}{2\alpha_1} \|p_1\|_{L^2}^2 + \chi_{|\vec{p}_2|_F \leq \lambda} + \frac{\gamma_2}{2\lambda} \|\vec{p}_2\|_{L^2}^2 =: D(\vec{p}) \right\}, \quad (2.2)$$

where for  $\alpha_1 = 0$  or  $\lambda = 0$  we use the convention that the terms  $\frac{\gamma_1}{2\alpha_1} \|p_1\|_{L^2}^2$  and  $\frac{\gamma_2}{2\lambda} \|\vec{p}_2\|_{L^2}^2$  vanish respectively.

Furthermore, solutions  $\vec{p} = (p_1, \vec{p}_2) \in W^*$  and  $\vec{u} \in V$  of (2.2) and (1.1) respectively are characterized by

$$\begin{aligned} 0 &= \Lambda^* \vec{p} - \alpha_2 T^* g + B \vec{u}, \\ 0 &= p_1 \max\{\gamma_1, |T \vec{u} - g|\} - \alpha_1 (T \vec{u} - g), \quad |p_1| \leq \alpha_1, \\ 0 &= \vec{p}_2 \max\{\gamma_2, |\nabla \vec{u}|_F\} - \lambda \nabla \vec{u}, \quad |\vec{p}_2|_F \leq \lambda, \end{aligned} \quad (2.3)$$

where  $\max$  denotes the pointwise maximum.

The primal functional  $E$  from (1.1) satisfies the following strong convexity property.

**Lemma 2.2.** *If  $\vec{u} \in V$  is a minimizer of  $E$ , then we have*

$$\frac{1}{2} \|\vec{v} - \vec{u}\|_B^2 \leq E(\vec{v}) - E(\vec{u})$$

for all  $\vec{v} \in V$ .

*Proof.* We apply the same method as in [22, Lemma 10.2]. For the energy  $E$  from (1.1) we write  $E(\vec{v}) = F(\vec{v}) + G(\vec{v})$  with

$$G(\vec{v}) := \alpha_1 \int_{\Omega} \varphi_{\gamma_1}(|T \vec{v} - g|) \, d\vec{x} + \lambda \int_{\Omega} \varphi_{\gamma_2}(|\nabla \vec{v}|_F) \, d\vec{x}$$

and

$$F(\vec{v}) := \frac{\alpha_2}{2} \|T \vec{v} - g\|_{L^2}^2 + \frac{\beta}{2} \|S \vec{v}\|_{L^2}^2.$$

Note that  $F$  is Frechet-differentiable with

$$\langle F'(\vec{v}), \vec{w} \rangle_{V^*, V} = \alpha_2 \langle T \vec{v} - g, T \vec{w} \rangle_{L^2} + \beta \langle S \vec{v}, S \vec{w} \rangle_{L^2}$$

for all  $\vec{w} \in V$ . Expanding  $F(\vec{v})$  at  $\vec{u}$  then yields

$$F(\vec{v}) = F(\vec{u}) + \langle F'(\vec{u}), \vec{v} - \vec{u} \rangle_{V^*, V} + \frac{1}{2} \|\vec{v} - \vec{u}\|_B^2.$$

Since  $\vec{u} \in V$  is a minimizer, we have  $0 \in \partial E(\vec{u}) = F'(\vec{u}) + \partial G(\vec{u})$ . Hence  $-F'(\vec{u}) \in \partial G(\vec{u})$ , i.e.

$$\langle -F'(\vec{u}), \vec{v} - \vec{u} \rangle_{V^*, V} \leq G(\vec{v}) - G(\vec{u})$$

and thus

$$\frac{1}{2} \|\vec{v} - \vec{u}\|_B^2 \leq F(\vec{v}) - F(\vec{u}) + G(\vec{v}) - G(\vec{u}),$$

which proves the statement.  $\square$

Putting Lemma 2.2 and coercivity of  $a_B$  from Assumption (As1) together we obtain that for a minimizer  $\vec{u}$  of  $E$  there is a constant  $c > 0$  such that

$$E(\vec{v}) - E(\vec{u}) \geq \frac{1}{2} \|\vec{u} - \vec{v}\|_B^2 = \begin{cases} \frac{c}{2} \|\vec{u} - \vec{v}\|_{L^2}^2 & \text{for Setting (S.i)} \\ \frac{c}{2} \|\vec{u} - \vec{v}\|_{H^1}^2 & \text{for Setting (S.ii)} \end{cases}$$

for all  $\vec{v} \in V$ .

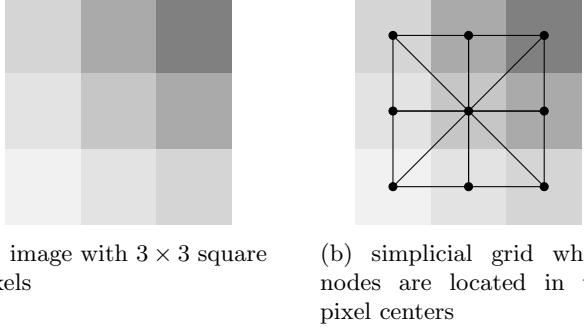
### 3 Discretization: Finite Elements

Central to the idea of finite element discretization is the idea to find a solution within a finite dimensional subspace of the original space. This discrete space usually consists of cellwise polynomial functions defined on a mesh of cells, e.g. simplices. In contrast to finite differences the discretization may be adaptively refined to accommodate certain local error indicators, thereby reducing the amount of degrees of freedom needed to represent a solution up to a certain accuracy. For more details on adaptive finite element schemes in general, we refer the reader to the survey [9].

We are interested in discretizing two-dimensional computer images of size  $n_1 \times n_2$  pixels, where  $n_1, n_2 \in \mathbb{N}$ , via finite elements. That is here and in the sequel we have  $d = 2$ . For such an image the domain  $\Omega$  is defined as  $\Omega := [1, n_1] \times [1, n_2]$ . We triangulate  $\Omega$  by simplices with nodes at integer coordinates  $(x_1, x_2) \in \mathbb{Z}^2$ ,  $1 \leq x_1 \leq n_1$ ,  $1 \leq x_2 \leq n_2$  corresponding to pixel centers as depicted in Figure 1.

Let  $\mathcal{T}$  denote the set of cells and  $\Gamma$  the set of oriented facets (i.e. edges as  $d = 2$ ) of the simplicial triangulation. For any cell  $K \in \mathcal{T}$  let  $P_k(K)$  be the space of polynomial functions on  $K$  with total degree  $k \in \mathbb{N}$ . We define the finite subspaces  $V_h \subseteq V \subseteq H^1(\Omega)^m$ ,  $W_h^* \subset L^2(\Omega) \times L^2(\Omega)^{d \times m} \simeq W^*$ , i.e.  $W^*$  can be identified with  $W = L^2(\Omega) \times L^2(\Omega)^{d \times m}$ , and  $Z_h \subset L^2(\Omega)$  as

$$\begin{aligned} V_h &:= \{\vec{u} \in C(\Omega)^m : \vec{u}|_K \in P_1(K)^m, K \in \mathcal{T}\}, \\ W_h^* &:= \{(p_1, \vec{p}_2) \in C(\Omega) \times L^2(\Omega)^{d \times m} : p_1|_K \in P_1(K), \vec{p}_2|_K \in P_0(K)^{d \times m}, \\ &\quad K \in \mathcal{T}\}, \\ Z_h &:= \{g \in C(\Omega) : g|_K \in P_1(K), K \in \mathcal{T}\}. \end{aligned} \tag{3.1}$$



**Fig. 1:** Image aligned simplicial grid construction as in [5].

As in [5] we do not discretize (1.1) directly but the optimality conditions (2.3) instead. In our discrete finite element setting this means that we search for solutions  $\vec{p}_h = (p_{h,1}, \vec{p}_{h,2}) \in W_h^*$ ,  $\vec{u}_h \in V_h$  which pointwise satisfy

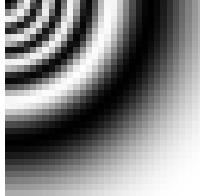
$$\begin{aligned} 0 &= \Lambda^* \vec{p}_h - \alpha_2 T^* g_h + B \vec{u}_h, \\ 0 &= p_{h,1} \max\{\gamma_1, |T \vec{u}_h - g_h|\} - \alpha_1 (T \vec{u}_h - g_h), \quad |p_{h,1}| \leq \alpha_1, \\ 0 &= \vec{p}_{h,2} \max\{\gamma_2, |\nabla \vec{u}_h|_F\} - \lambda \nabla \vec{u}_h, \quad |\vec{p}_{h,2}|_F \leq \lambda, \end{aligned} \quad (3.2)$$

where  $g_h \in Z_h$  is a discrete version of the data  $g$ . We elaborate further in Section 3.1 below on possibilities how to map image data onto the given grid. Note that the last two equations in (3.2) are enforced on vertices only. This is due to the fact, that the expression  $|T \vec{u}_h - g_h|$  is not necessarily cell-wise linear, even though  $T \vec{u}_h - g_h$  is.

Further we define: Let  $F \in \Gamma$  be an oriented inner facet with adjacent cells  $K_1, K_2 \in \mathcal{T}$  and  $\vec{\varphi} \in L^2(K_1 \cup K_2)^{d \times m}$  with  $\vec{\varphi}|_{K_i} \in C(K_i)^{d \times m}$ ,  $i \in \{1, 2\}$ , a square integrable function which allows continuous representations  $\vec{\varphi}|_{K_i}$  on each cell  $K_i$  individually. We then define the jump term  $[\vec{\varphi}]_F \in C(F)^{d \times m}$  by  $[\vec{\varphi}]_F(\vec{x}) := \vec{\varphi}|_{K_1}(\vec{x}) - \vec{\varphi}|_{K_2}(\vec{x})$  and omit the index as in  $[\vec{\varphi}]$  when the facet in question is clear. For outer facets only the term corresponding to the existing adjacent cell is considered. Note that  $[\vec{\varphi}]_F$  is in general dependent on the orientation of  $F$ , while e.g.  $[\vec{n}^\top \vec{\varphi}]_F$  for the oriented facet-normal  $\vec{n}$  is not.

### 3.1 On Image Interpolation Methods

For aligned grids as in Figure 1 image data  $A \in [0, 1]^{n_1 \times n_2}$  is interpolated to a grid function  $g_h \in Z_h$  by nodal sampling. This way, when resampling at pixel centers one receives the original image. For non-aligned grids, i.e. when pixel centers do not correspond to vertices of the grid, a mapping of the image  $A$  to some grid function  $g_h \in Z_h$  has to be computed. Choosing nodal sampling (e.g. of bilinearly interpolated image data) for this map can result in aliasing and is a well known undesired effect in image processing [23]. The Shannon-Whittaker sampling theorem provides ample conditions for correct equidistant sampling of bandwidth-limited functions [23, 24], namely to remove high-frequency contributions (e.g. by a linear smoothing filter) before sampling



**Fig. 2:** Discrete input image for mesh interpolation comparison,  $32 \times 32$  pixels.

**Table 1:** PSNR values (higher is better) of interpolated mesh functions from Figure 3, sampled at image coordinates. Here `nr_vertices` denotes the number of vertices per dimension.

<code>nr_vertices</code>	<code>nodal</code>	<code>12_lagrange</code>	<code>qi_lagrange</code>	<code>12_pixel</code>
32	$\infty$	55.826	22.606	327.597
16	21.141	22.020	21.483	23.106
13	18.861	19.650	19.053	19.981

on a coarser grid. It is however unclear to us how to apply this to general unstructured sampling points.

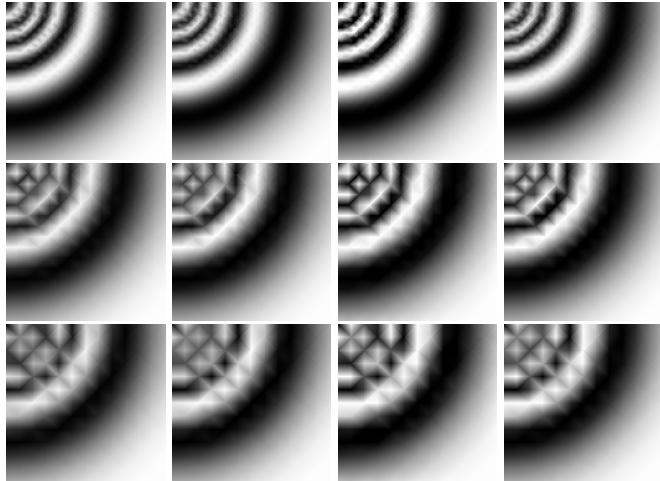
We evaluate three projection schemes and then explore one alternative. Let  $g \in C(\Omega)$  be the bilinear interpolation of the nodal image data  $A$ . For methods involving quadratures the quadrature over a cell  $E \in \mathcal{T}$  is computed using a simple averaging quadrature on a Lagrange lattice of degree  $[\text{diam}(E)]$ , i.e. we properly scale the number of quadrature points depending on the size of the cell (and thus the number of pixels it covers) to avoid aliasing effects.

The projection schemes are defined as follows:

- (i) **nodal**: Nodal interpolation, i.e.  $g_h \in Z_h$  such that  $g_h(\vec{x}) := g(\vec{x})$  at every mesh vertex  $\vec{x}$ .
- (ii) **12\_lagrange**: Standard  $L^2$ -projection, i.e.  $g_h \in Z_h$  such that  $\int_{\Omega} g_h \cdot \varphi \, d\vec{x} = \int_{\Omega} g \cdot \varphi \, d\vec{x}$  for all  $\varphi \in Z_h$ .
- (iii) **qi\_lagrange**: The general  $L^1$ -stable quasi-interpolation operator as proposed in [25], i.e. the continuous  $g_h \in Z_h$  is given by setting its nodal degrees of freedom at each vertex to the arithmetic mean of the corresponding local degrees of freedom of a discontinuous interpolant  $\tilde{g}_h \in \tilde{Z}_h := \{f \in L^\infty(\Omega) : f|_K \in P_1(K), K \in \mathcal{T}\}$ , which is defined on each cell  $K \in \mathcal{T}$  by its local nodal degrees of freedom  $\sigma_{K,i} = \frac{1}{|K|} \int_K g(\vec{x}) \cdot \rho_{K,i}(\vec{x}) \, d\vec{x}$ ,  $i = 1, \dots, 3$  where the test function  $\rho_{K,i}$  in our case is given in barycentric coordinates  $\lambda_K$  by  $\rho_{K,i}(\lambda_K) = 12\lambda_{K,i} - 3$ . The reader may refer to [25] for details on the general construction.
- (iv) **12\_pixel**: Our proposed method, i.e. minimizing the sum of pointwise squared errors over all pixel coordinates:  $\inf_{g_h \in Z_h} \sum_{\vec{x} \in \Omega \cap \mathbb{Z}^2} \frac{1}{2} |g_h(\vec{x}) - g(\vec{x})|^2$ .

Note that **12\_pixel** may be interpreted as an  $L^2$ -projection with a cell-dependent averaging quadrature rule, adapted to the original image pixel locations.

We evaluate these schemes by interpolating the discrete image given in Figure 2 onto  $V_h$  for regular meshes of different sizes. In Figure 3 we see the interpolated results



**Fig. 3:** Interpolated finite element functions for varying mesh sizes and interpolation methods. Methods from left to right: `nodal`, `12_lagrange`, `qi_lagrange`, `12_pixel`. Number of mesh vertices in each dimension from top to bottom: 32, 16, 13.

**Table 2:** SSIM values (higher is better) of interpolated mesh functions from Figure 3, sampled at image coordinates. Here `nr_vertices` denotes the number of vertices per dimension.

<code>nr_vertices</code>	<code>nodal</code>	<code>12_lagrange</code>	<code>qi_lagrange</code>	<code>12_pixel</code>
32	1.00000	0.99998	0.97846	1.00000
16	0.93868	0.95133	0.94757	0.96693
13	0.89307	0.90888	0.90053	0.92080

as cellwise linear functions. We sample the interpolated functions on the original image grid  $\Omega_h$  and quantify the error to the original image. In particular, Tables 1 and 2 list the PSNR, given for two scalar images  $u, v : \Omega_h \rightarrow [0, 1]$  by  $\text{PSNR}(u, v) := -10 \log_{10} \left( \frac{1}{|\Omega_h|} \sum_{\vec{x} \in \Omega_h} (u(\vec{x}) - v(\vec{x}))^2 \right)$ , and the structural similarity index SSIM as defined in [26]. Perhaps unsurprisingly, due to construction, `12_pixel` produces results closest to the original image for both metrics. We note that, theoretically, `12_pixel` in the finest mesh setting should provide exact results (i.e. infinite PSNR) and the discrepancy is due to numerical imprecision.

It should be noted that while `12.pixel` seems to provide visually superior results, it does not necessarily preserve other quantities of the image, such as the total mass, which may be relevant for the image processing task in question. Further, it is not necessarily well-defined for meshes finer than the original image and in that case a regularization needs to be applied.

We conclude that, since digital images are usually given as an array of brightness values and the output of image processing algorithms is expected to be so too, working in unstructured finite element spaces comes with the drawback of information

loss due to mesh interpolation. Apart from increased complexity, this may be considered a problem for applying unstructured adaptive finite element methods to image processing tasks in practice.

### 3.2 Residual A-Posteriori Error Estimator

In [19] an a-posteriori error estimate for a smooth functional, composed of an  $L^2$ -data term and a total variation term, was derived using residual-based methods for variational inequalities. We try to use a similar technique for our general setting to establish a suitable guiding criterion for adaptive spatial discretization. Our presentation avoids the variational inequality setting in [19] by making use of strong convexity, which seems to lead to a simpler presentation. Additionally, we consider both choices of Settings (S.i) and (S.ii), which requires different interpolation estimates depending on the corresponding coercivity of  $a_B$  stated in Assumption (A<sub>s1</sub>).

We recall  $E : V \rightarrow \overline{\mathbb{R}}$  from (1.1) in the split form  $E(\vec{u}) = F(\vec{u}) + G(\Lambda\vec{u})$ , where  $\Lambda = (T, \nabla) : V \rightarrow W$  as in Section 2.2 and  $F : V \rightarrow \overline{\mathbb{R}}$ ,  $G : W \rightarrow \overline{\mathbb{R}}$  are given by

$$\begin{aligned} F(\vec{u}) &:= \frac{\alpha_2}{2} \|T\vec{u} - g\|_{L^2}^2 + \frac{\beta}{2} \|S\vec{u}\|_{L^2}^2, \\ G(\Lambda\vec{u}) &:= \alpha_1 \int_{\Omega} \varphi_{\gamma_1}(|T\vec{u} - g|) \, d\vec{x} + \lambda \int_{\Omega} \varphi_{\gamma_2}(|\nabla\vec{u}|_F) \, d\vec{x}. \end{aligned}$$

The optimality conditions in this setting, see [27, Remark III.4.2], are given by

$$\begin{aligned} -\Lambda^* \vec{p} &\in \partial F(\vec{u}), \\ \vec{p} &\in \partial G(\Lambda\vec{u}), \end{aligned}$$

where we accounted for the sign change in the dual variable  $\vec{p}$  to be consistent with the formulation in (2.3). We assume for the moment that  $\alpha_1 = 0$  and the discrete solution pair  $\vec{u}_h \in V_h$ ,  $\vec{p}_h \in W_h^*$  satisfies the corresponding discrete optimality conditions

$$\begin{aligned} -\Lambda^* \vec{p}_h &\in \partial F_h(\vec{u}_h), \\ \vec{p}_h &\in \partial G_h(\Lambda\vec{u}_h), \end{aligned} \tag{3.3}$$

where for clarity we explicitly denoted with  $F_h : V_h \rightarrow \overline{\mathbb{R}}$ ,  $G_h : W_h \rightarrow \overline{\mathbb{R}}$  the restrictions of  $F$ ,  $G$  on the respective discrete spaces, since their subdifferentials differ from those of  $F$  and  $G$  respectively. Note that  $W_h$  denotes the (pre-)dual space of  $W_h^*$ . One may obtain this system analogously to [5] by application of the Fenchel duality from [27, Remark III.4.2] on the corresponding discrete energy functional. Indeed, Proposition 2.1 in general allows for  $V$  and  $W$  to be appropriate subspaces. This way existence of a discrete solution pair  $\vec{u}_h \in V_h$ ,  $\vec{p}_h \in W_h^*$  is ensured. Since  $F$  is Fréchet-differentiable we see that

$$\begin{aligned} \langle \partial F(\vec{u}), \vec{v} \rangle_{V^*, V} &= \alpha_2 \langle T\vec{u} - g, T\vec{v} \rangle_{L^2} + \beta \langle S\vec{u}, S\vec{v} \rangle_{L^2} \\ &= a_B(\vec{u}, \vec{v}) - l(\vec{v}) \end{aligned}$$

with  $l(\vec{v}) := \alpha_2 \langle T\vec{v}, g \rangle_{L^2}$ . Analogously  $\langle \partial F_h(\vec{u}_h), \vec{v}_h \rangle_{V_h^*, V_h} = a_B(\vec{u}_h, \vec{v}_h) - l(\vec{v}_h)$  and from (3.3) we infer for all  $\vec{v}_h \in V_h$ :

$$-\langle \vec{p}_h, \Lambda \vec{v}_h \rangle_{L^2} = a_B(\vec{u}_h, \vec{v}_h) - l(\vec{v}_h). \quad (3.4)$$

For  $\partial G$  on the other hand we have the following non-obvious observation.

**Lemma 3.1.** *Let  $\vec{p}_h = (p_{h,1}, \vec{p}_{h,2}) \in W_h^*$  satisfy (3.3). If  $\alpha_1 = 0$  then*

$$\vec{p}_h \in \partial G(\Lambda \vec{u}_h),$$

where  $\vec{p}_h \in W_h^* \subset W^*$  is identified with an element of the dual space  $(W^*)^* = W$ .

*Proof.* Let  $\vec{q} = (q_1, \vec{q}_2) \in \partial G(\Lambda \vec{u}_h)$ . Due to (2.3) and (3.3) we deduce  $q_1 = p_{h,1} = 0$  since  $\alpha_1 = 0$ . From (2.3) we infer for  $\vec{q}_2$ , and analogously for  $\vec{p}_{h,2}$ , due to (3.3) that pointwise, whenever  $\nabla \vec{u}_h \neq 0$ , we have

$$\vec{q}_2 = \frac{\lambda \nabla \vec{u}_h}{\max\{\gamma_2, |\nabla \vec{u}_h|_F\}} = \vec{p}_{h,2}.$$

Therefore  $\vec{q}_2$  is piecewise constant whenever  $\nabla \vec{u}_h \neq 0$ . Further both  $\vec{q}_2$  and  $\vec{p}_{h,2}$  are bounded pointwise by  $|\vec{q}_2|_F \leq \lambda$  and  $|\vec{p}_{h,2}|_F \leq \lambda$ . On cells where  $\nabla \vec{u}_h = 0$  holds,  $\vec{q}_2$  may assume any value bounded by  $|\vec{q}_2|_F \leq \lambda$ . Since this holds true for  $\vec{p}_{h,2}$  in particular, one may choose  $\vec{q} := (0, \vec{p}_{h,2}) \in \partial G(\Lambda \vec{u}_h)$ .  $\square$

We note that for  $\alpha_1 > 0$  the optimality conditions (3.3) might not be valid for  $\vec{p}_h \in W_h^*$  due to the presence of the operator  $T$ . A possibility to circumvent this shortcoming might be to enlarge the space  $W_h^*$  to  $\{(p_1, \vec{p}_2) \in L^2(\Omega) \times L^2(\Omega)^{d \times m} : p_1|_K \in P_1(K), \vec{p}_2|_K \in P_0(K)^{d \times m}, K \in \mathcal{T}\}$ . However, even in such a new setting Lemma 3.1 would not necessarily hold true for  $\alpha_1 > 0$ , as  $\partial G(\Lambda \vec{u}_h) = \alpha_1 \frac{T\vec{u}_h - g}{\max\{\gamma_1, |T\vec{u}_h - g|\}}$  does not need to be piecewise linear, even if  $\vec{u}_h$  is. We will continue the derivation nevertheless, noting that theoretical justification is lacking for  $\alpha_1 > 0$ . That is in the sequel we assume that

$$\vec{p}_h \in \partial G(\Lambda \vec{u}_h). \quad (3.5)$$

With these observations we are ready to estimate the error  $\|\vec{u}_h - \vec{u}\|_V$ .

**Theorem 3.2.** *Let  $\vec{u} \in V$  be a solution of (1.1), and let  $\vec{u}_h \in V_h$  and  $\vec{p}_h \in W_h^*$  satisfy (3.3). If Assumption (As1) and (3.5) hold, then we have*

(i) *for Setting (S.ii)*

$$\begin{aligned} \|\vec{u}_h - \vec{u}\|_{H^1(\Omega)^m}^2 &\leq C \left( \sum_{K \in \mathcal{T}} h_K^2 \|\alpha_2 T^*(T\vec{u}_h - g) + T^* p_{h,1} - \beta \Delta \vec{u}_h - \operatorname{div} \vec{p}_{h,2}\|_{L^2(K)^m}^2 \right. \\ &\quad \left. + \sum_{F \in \Gamma} h_F \|[\vec{n}^\top (\beta \nabla \vec{u}_h + \vec{p}_{h,2})]\|_{L^2(F)^m}^2 \right) \end{aligned}$$

with constant  $C > 0$ , and

(ii) for Setting (S.i)

$$\begin{aligned} \|\vec{u}_h - \vec{u}\|_{L^2(\Omega)^m}^2 &\leq C \left( \sum_{K \in \mathcal{T}} \|\alpha_2 T^*(T\vec{u}_h - g) + T^* p_{h,1} + \beta \vec{u}_h - \operatorname{div} \vec{p}_{h,2}\|_{L^2(K)^m}^2 \right. \\ &\quad \left. + \sum_{F \in \Gamma} h_F^{-1} \|[\vec{n}^\top \vec{p}_{h,2}]\|_{L^2(F)^m}^2 \right) \end{aligned}$$

for some constant  $C > 0$ .

The proof of this theorem is postponed to Appendix A.

## Error Indicators

Theorem 3.2 gives raise to a local error indicator. In particular, for a cell  $K \in \mathcal{T}$  we define our local error indicators as follows

$$\eta_{\text{res},K}^2 := \tilde{\eta}_{\text{res},K}^2 + \sum_{\substack{F \in \Gamma \\ F \cap K \neq \emptyset}} \tilde{\eta}_{\text{res},F}^2, \quad (3.6)$$

where  $\tilde{\eta}_{\text{res},K}$  and  $\tilde{\eta}_{\text{res},F}$  are chosen as follows. For Setting (S.ii) and  $S = \nabla$  we set

$$\begin{aligned} \tilde{\eta}_{\text{res},K}^2 &:= h_K^2 \|\alpha_2 T^*(T\vec{u}_h - g) + T^* p_{h,1} - \beta \Delta \vec{u}_h - \operatorname{div} \vec{p}_{h,2}\|_{L^2(K)^m}^2, \\ \tilde{\eta}_{\text{res},F}^2 &:= h_F \|[\vec{n}^\top (\beta \nabla \vec{u}_h + \vec{p}_{h,2})]\|_{L^2(F)^m}^2, \end{aligned} \quad (3.7)$$

where for piecewise linear  $\vec{u}_h$  and piecewise constant  $\vec{p}_{h,2}$  the terms  $\Delta \vec{u}_h$  and  $\operatorname{div} \vec{p}_{h,2}$  in (3.7) vanish. For Setting (S.i) and  $S = I$  we set

$$\begin{aligned} \tilde{\eta}_{\text{res},K}^2 &:= \|\alpha_2 T^*(T\vec{u}_h - g) + T^* p_{h,1} + \beta \vec{u}_h - \operatorname{div} \vec{p}_{h,2}\|_{L^2(K)^m}^2, \\ \tilde{\eta}_{\text{res},F}^2 &:= h_F^{-1} \|[\vec{n}^\top \vec{p}_{h,2}]\|_{L^2(F)^m}^2. \end{aligned}$$

For Setting (S.i) with  $S = I$  the facet indicator  $\tilde{\eta}_{\text{res},F}$  scales inversely with the diameter and is therefore not very useful in the context of adaptive refinement. This is due to  $a_B$  only being  $L^2$ -coercive in that case, which limits our choice of interpolation estimates and it is unclear to us whether this result can be improved. Showing efficiency of these estimators may be considered in future work and is expected to work out in a similar way as in [19].

In the sequel we call the indicator (3.6) *residual based indicator*.

### 3.3 Primal-Dual A-Posteriori Error Estimator

In the following we use the same approach as in [17, 28] to derive a-posteriori error estimates from the primal-dual energy gap. We slightly adjust the arguments to account for the Huber regularization and potentially non-local operators  $T$  and  $B$ .

Utilizing `l2_lagrange`, we let  $g_h \in Z_h$  be the  $L^2$ -projection of  $g \in L^2(\Omega)$  onto  $Z_h$ , i.e.  $\|g_h - g\|_{L^2} = \inf_{v \in Z_h} \|v - g\|_{L^2}$ . Recalling (2.2) and (1.1) we define discretized functionals of  $D$  and  $E$  by  $D_h : W_h^* \rightarrow \overline{\mathbb{R}}$  and  $E_h : V_h \rightarrow \overline{\mathbb{R}}$  using the discretized data  $g_h$  as

$$\begin{aligned} D_h(\vec{p}_h) &:= \frac{1}{2} \|T^* p_{h,1} + \nabla^* \vec{p}_{h,2} - \alpha_2 T^* g_h\|_{B^{-1}}^2 - \frac{\alpha_2}{2} \|g_h\|_{L^2}^2 + \langle g_h, p_{h,1} \rangle_{L^2} \\ &\quad + \chi_{|p_{h,1}| \leq \alpha_1} + \chi_{|\vec{p}_{h,2}|_F \leq \lambda} + \frac{\gamma_1}{2\alpha_1} \|p_{h,1}\|_{L^2}^2 + \frac{\gamma_2}{2\lambda} \|\vec{p}_{h,2}\|_{L^2}^2, \\ E_h(\vec{u}_h) &:= \alpha_1 \int_{\Omega} \varphi_{\gamma_1}(|T\vec{u}_h - g_h|) d\vec{x} + \frac{\alpha_2}{2} \|T\vec{u}_h - g_h\|_{L^2}^2 + \frac{\beta}{2} \|S\vec{u}_h\|_{L^2}^2 \\ &\quad + \lambda \int_{\Omega} \varphi_{\gamma_2}(|\nabla \vec{u}_h|_F) d\vec{x}. \end{aligned}$$

Note that  $\nabla^*$  may be understood as  $\nabla^* : L^2(\Omega)^{d \times m} \rightarrow V^*, \vec{p} \mapsto (\vec{w} \mapsto \langle \vec{p}, \nabla \vec{w} \rangle_{L^2})$ .

**Theorem 3.3.** *Let  $u \in V$  be the solution to (1.1) and  $\vec{u}_h \in V_h$  be the discrete minimizer of  $E_h$ . Then one has for any  $v_h \in V_h$ ,  $\vec{q}_h \in W_h^*$  the a-posteriori error estimate*

$$\frac{1}{2} \|\vec{u}_h - \vec{u}\|_B^2 \leq \eta_h^2(v_h, \vec{q}_h) + c \|g_h - g\|_{L^2},$$

where the estimator  $\eta_h : V_h \times W_h^* \rightarrow \overline{\mathbb{R}}$  is given as

$$\begin{aligned} \eta_h^2(v_h, \vec{q}_h) &:= E_h(v_h) + D_h(\vec{q}_h) \\ &= \alpha_1 \int_{\Omega} \varphi_{\gamma_1}(|Tv_h - g_h|) d\vec{x} - \langle Tv_h - g_h, q_{h,1} \rangle_{L^2} + \frac{\gamma_1}{2\alpha_1} \|q_{h,1}\|_{L^2}^2 \\ &\quad + \lambda \int_{\Omega} \varphi_{\gamma_2}(|\nabla v_h|_F) d\vec{x} - \langle \nabla v_h, \vec{q}_{h,2} \rangle_{L^2} + \frac{\gamma_2}{2\lambda} \|\vec{q}_{h,2}\|_{L^2}^2 \\ &\quad + \frac{1}{2} \left\| B^{-1}(\Lambda^* \vec{q}_h - \alpha_2 T^* g_h) + v_h \right\|_B^2 \end{aligned} \tag{3.8}$$

with  $|q_{h,1}| \leq \alpha_1$ ,  $|\vec{q}_{h,2}|_F \leq \lambda$ , and  $c > 0$  is a constant depending only on the model parameters, the model operators  $T$  and  $S$ , the domain size  $|\Omega|$ , and data  $g$ .

### Error Indicators

Theorem 3.3 motivates a corresponding cell-wise error indicator as follows: Let  $\vec{u}_h \in V_h$ ,  $\vec{p}_h \in W_h^*$  be feasible discrete approximations to the solution of (2.3). The primal reconstruction from the dual variable is given weakly by

$$w := -B^{-1}(\Lambda^* \vec{p}_h - \alpha_2 T^* g_h).$$

For every cell  $K \in \mathcal{T}$  we then define the cell indicator as

$$\begin{aligned} \eta_{\text{pd},K}^2(\vec{u}_h, \vec{p}_h) := & \int_K \alpha_1 \varphi_{\gamma_1}(|T\vec{u}_h - g_h|) - (T\vec{u}_h - g_h) \cdot p_{h,1} + \frac{\gamma_1}{2\alpha_1} |p_{h,1}|^2 \, d\vec{x} \\ & + \int_K \lambda \varphi_{\gamma_2}(|\nabla \vec{u}_h|_F) - \nabla \vec{u}_h \cdot \vec{p}_{h,2} + \frac{\gamma_2}{2\lambda} |\vec{p}_{h,2}|_F^2 \, d\vec{x} \\ & + \frac{1}{2} \int_K |w - \vec{u}_h|^2 \, d\vec{x} \end{aligned} \quad (3.9)$$

and denote it as *primal-dual based indicator*.

### 3.4 Adaptive Refinement Strategy

For refinement we bisect triangles using the newest-vertex strategy [9], i.e. the bisection edge is chosen to be opposite of the vertex which was inserted last. In an adaptive refinement setting we mark cells for refinement using the greedy Dörfler marking strategy [9] with  $\theta_{\text{mark}} \in [0, 1]$ , e.g.  $\theta_{\text{mark}} = 0.5$ , i.e. given error indicators in descending order  $(\eta_{K_n})_{1 \leq n \leq |\mathcal{T}|}$  for triangles  $K_n \in \mathcal{T}$  we refine the first  $n_{\text{mark}} \in \mathbb{N}$  triangles that satisfy

$$\sum_{n=1}^{n_{\text{mark}}} \eta_{K_n} \geq \theta_{\text{mark}} \sum_{n=1}^{|\mathcal{T}|} \eta_{K_n}.$$

Our generic adaptive finite element algorithm for solving (1.1) is presented in Algorithm 1.

---

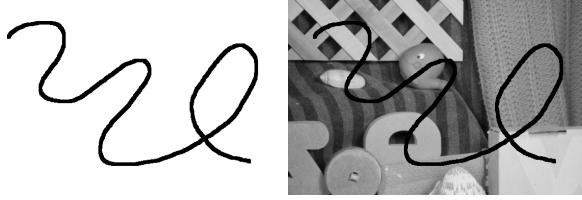
**Algorithm 1** AFEM for (1.1)

---

- 1.) Choose an initial grid  $\mathcal{T}$ .
  - 2.) Find a finite element solution  $\vec{u}_h$  of (1.1) on  $\mathcal{T}$ , e.g. by [5, Algorithm 1].
  - 3.) Compute on each cell a local indicator, e.g. given by (3.6) or (3.9).
  - 4.) Stop if a certain termination criterion holds (e.g. a certain number of refinements is reached), or continue with next step.
  - 5.) Determine  $\mathcal{R}_h$ , the set of cells to be refined, by using the greedy Dörfler strategy as described in Section 3.4.
  - 6.) Refine the cells  $K \in \mathcal{R}_h$  by utilizing newest-vertex bisection and set  $\mathcal{T}$  to be the new grid.
  - 7.) Project the image data onto  $\mathcal{T}$  (e.g. by `12_lagrange` or `12_pixel`) and continue with step 2.).
- 

## 4 Applications

In the following, we demonstrate that our model (1.1), in conjunction with Algorithm 1, can effectively be applied in practice to solve image processing tasks such as



**Fig. 4:** Inpainting domain (left), corrupted image (right).

image inpainting and optical flow computation in image sequences. Thereby we apply Algorithm 1 in the following way: As initial grid in step 1.) a coarse grid is chosen which is coarser than the observed, original image grid. Based on the local indicators and the greedy Dörfler strategy in step 6.) only a certain amount of, namely  $n_{\text{mark}}$ , cells are refined. Utilizing Algorithm 1 in this way generates a grid which is partly coarse and partly fine, see e.g. Figure 5. In this vein we call this strategy an adaptive coarse-to-fine scheme. Moreover, in all our experiments to find a finite element solution  $\bar{u}_h$  in step 2.) of Algorithm 1 we utilize the primal-dual semi-smooth Newton method of [5].

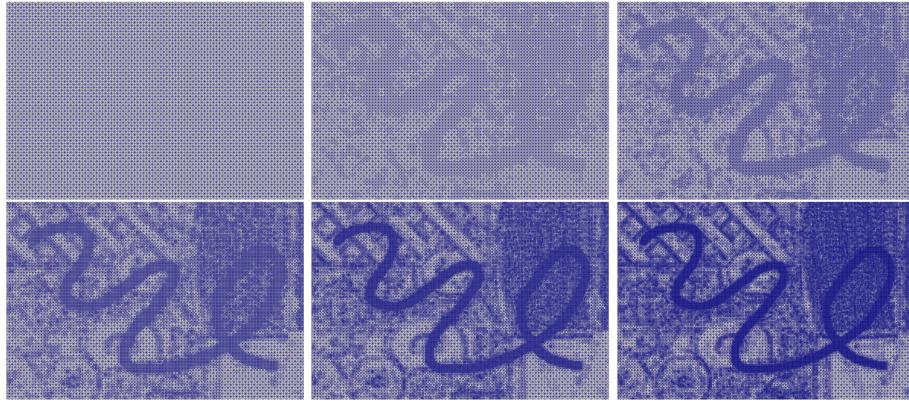
The implementation of the performed algorithms is done in the programming language Julia [29] and can be found at [30]. All evaluations were done on a notebook with a Intel Core i7-10875H CPU @ 2.30GHz and 32GB RAM.

#### 4.1 Inpainting

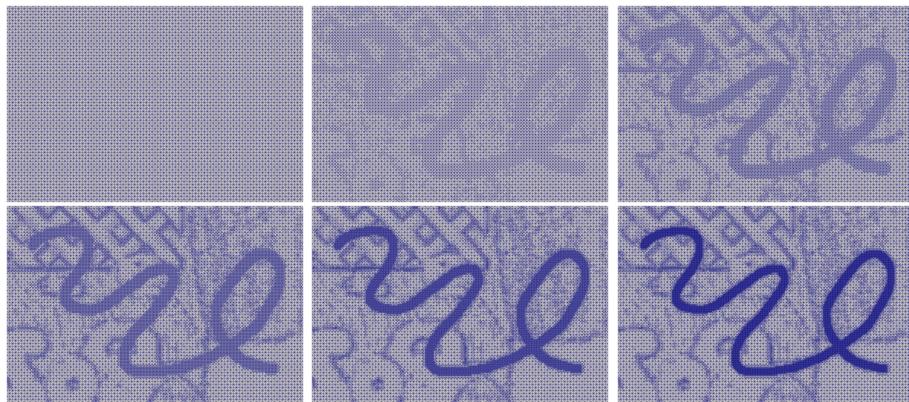
Image inpainting is the problem of reconstructing a given defected image  $g \in L^2(\Omega \setminus D)$ , whereby any image information in  $D \subset \Omega$  is missing. The domain  $D$  is therefore called inpainting domain. Figure 4 depicts an example of an inpainting domain  $D$  and a respective corrupted observation  $g$ . We tackle this inpainting task by using model (1.1) and utilizing Algorithm 1. Thereby we set  $T := \text{Id}_{\Omega \setminus D}$ , where

$$(\text{Id}_{\Omega \setminus D} u)(\vec{x}) := \begin{cases} u(\vec{x}) & \text{if } \vec{x} \in \Omega \setminus D, \\ 0 & \text{if } \vec{x} \in D, \end{cases}$$

for any  $u \in L^2(\Omega)$ ,  $S = I$  and we manually select the parameters  $\alpha_1 = 0$ ,  $\alpha_2 = 50$ ,  $\lambda = 1$ ,  $\beta = 1 \cdot 10^{-5}$ ,  $\gamma_1 = 1 \cdot 10^{-4}$ ,  $\gamma_2 = 1 \cdot 10^{-4}$ . The finite element solution in each iteration is computed by [5, Algorithm 1] with  $\varepsilon_{\text{newton}} = 1 \cdot 10^{-4}$ . The initial grid is chosen to be uniform and  $\frac{1}{2^{n_{\text{coarsen}}/2}}$  of the image resolution, rounded down to integer values, where  $n_{\text{coarsen}} \in \mathbb{N}_0$ . The algorithm performs then  $n_{\text{refine}} = \lfloor \log_2 \frac{\text{Number of cells in the original image}}{\text{Number of cells in the initial grid}} \rfloor$  refinements, see Figure 5 for  $n_{\text{coarsen}} = 5$  which leads to  $n_{\text{refine}} = 5$ . This allows that the smallest element is approximately of the size of the elements of the original image, but not smaller. Note that our indicators, presented in Section 3, may not be insightful in the inpainting domain. For example in the case of the primal-dual based indicators if  $n_{\text{coarsen}}$  is so large that the initial grid consists only of the inpainting domain, then the local indicators calculate to 0 and no element will be marked for refinement. To circumvent such situations, after



**Fig. 5:** Adapted mesh for inpainting during the iteration for  $n_{\text{coarsen}} = 5 = n_{\text{refine}}$  using the residual based indicators and  $\theta_{\text{mark}} = 0.5$ .



**Fig. 6:** Adapted mesh for inpainting during the iteration for  $n_{\text{coarsen}} = 5 = n_{\text{refine}}$  with the primal-dual based indicators and  $\theta_{\text{mark}} = 0.99$ .

applying the greedy Dörfler marking strategy we additionally mark all elements in the inpainting area for refinement for both types of indicators. We use `qi_lagrange` for projecting onto the finite element space except when  $n_{\text{refine}} = 0$ , in which case we use nodal interpolation of the image data. The choice of `qi_lagrange` over `12_pixel` and `12_lagrange` is based on the following factors. Firstly, `12_pixel` is not well-suited for inpainting tasks because it is a global operator, which can cause data leakage from the inpainting domain. Additionally, `qi_lagrange` seems to perform slightly better than `12_lagrange` in our example, offering faster results and improved restoration quality.

We present in Figure 5 for the residual based indicators with  $\theta_{\text{mark}} = 0.5$  and in Figure 6 for the primal-dual based indicators with  $\theta_{\text{mark}} = 0.99$  (see below for the reason of this choice) the adaptive mesh generated by Algorithm 1 in each iteration for  $n_{\text{refine}} = 5$ . Since the edges of the elements are depicted in blue in Figures 5 and 6,



**Fig. 7:** Reconstruction obtained with  $n_{\text{coarsen}} = 0$  (left),  $n_{\text{coarsen}} = 5$  (middle),  $n_{\text{coarsen}} = 10$  (right) using the residual based indicators.



**Fig. 8:** Reconstruction obtained with  $n_{\text{coarsen}} = 1$  (left),  $n_{\text{coarsen}} = 5$  (middle),  $n_{\text{coarsen}} = 10$  (right) using the primal-dual based indicators.

dark areas indicate fine refinement, while in brighter areas the mesh is coarser. Recall that all elements in the inpainting domain are always marked for refinement. Moreover, we observe that in uniform areas as in the right lower corner of our image, the mesh stays coarse as this seems sufficient to keep the approximation error at a reasonable level and still obtain a visually appealing reconstruction, see Figures 7 and 8.

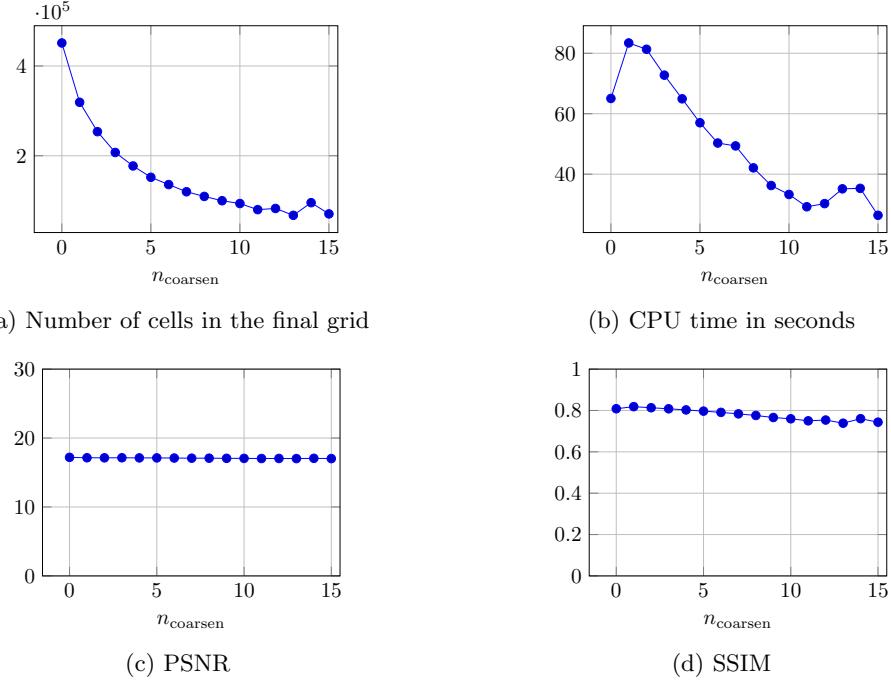
Note that using the primal-dual gap as an error estimator by computing it on a discrete mesh requires solving the model in the primal-dual gap equation on a finer mesh than what the candidate solution is using in order to provide meaningful results. Otherwise on the same mesh it will evaluate to zero. Moreover, we observed that for this application and with  $\theta_{\text{mark}} = 0.5$  the primal-dual based indicators allow to select for refinement only cells in the inpainting domain leading to undesirable results. However, setting  $\theta_{\text{mark}} = 0.99$  leads to similar results as with the residual based indicators and  $\theta_{\text{mark}} = 0.5$ , see Figures 7 and 8. From the mesh depicted in Figure 6 one easily sees that even with  $\theta_{\text{mark}} = 0.99$  the algorithm mainly refines in the inpainting domain. Hence the mesh is much coarser (76082 cells in the final grid with  $n_{\text{coarsen}} = 5$ ) than the one generated with the help of the residual based indicators (152081 cells in the final grid with  $n_{\text{coarsen}} = 5$ ). Nevertheless, the computation with the primal-dual based indicators is much more expensive, as the indicators need to be computed on a globally refined mesh in each iteration. In particular, for  $n_{\text{coarsen}} = 5$  it takes 95.35 seconds generating a restoration with PSNR = 17.07 and SSIM = 0.7459. On the contrary, using the residual based indicators for  $n_{\text{coarsen}} = 5$  a result is generated within 57.01 seconds with PSNR = 17.12 and SSIM = 0.7968. Quantitative similar results are obtained with different  $n_{\text{coarsen}}$ , suggesting that for inpainting our residual based indicators perform better in Algorithm 1 than our primal-dual based indicators.

**Table 3:** Comparison of different values  $n_{\text{coarsen}}$  for solving the inpainting problem in Figure 4 using the residual based indicators.

$n_{\text{coarsen}}$	$n_{\text{refine}}$	# cells in final grid	PSNR	SSIM	Time (s)
0	0	451 242	17.20	0.8083	65.05
1	1	319 120	17.14	0.8184	83.41
2	2	253 678	17.13	0.8132	81.33
3	3	207 341	17.14	0.8081	72.76
4	4	177 321	17.12	0.8027	64.93
5	5	152 081	17.12	0.7968	57.01
6	6	135 825	17.10	0.7908	50.30
7	7	119 846	17.08	0.7836	49.35
8	8	109 463	17.08	0.7758	42.09
9	9	99 839	17.06	0.7659	36.25
10	10	93 638	17.05	0.7598	33.29
11	11	80 006	17.03	0.7499	29.22
12	12	82 627	17.04	0.7537	30.26
13	13	67 377	17.02	0.7385	35.15
14	15	95 524	17.06	0.7603	35.31
15	16	70 524	17.02	0.7434	26.39

### Influence on the initial grid

Here we restrict ourselves to the residual based indicators, since in inpainting they are superior to the primal-dual based indicators. We investigate the behavior of our algorithm for different choices of  $n_{\text{coarsen}}$ , namely  $n_{\text{coarsen}} \in \{0, 1, 2, \dots, 15\}$ , when applied to the inpainting problem in Figure 4. Note that with  $n_{\text{coarsen}} = 0$  no coarsening is performed and hence also no refinement, i.e. with this choice we indicate the performance of the algorithm on a uniform mesh as presented in [5]. Since the size of the image in Figure 4 is  $584 \times 388$  pixels,  $n_{\text{coarsen}} = 15$  is the maximal possible number for coarsening the image yielding an image with  $3 \times 2$  vertices, i.e. 4 cells. Our findings, namely the number of cells in the finest grid, the PSNR and SSIM values of the restoration as well as the CPU time needed to obtain the reconstruction, are summarized in Table 3 and visualized in Figure 9. As expected, we observe that increasing  $n_{\text{coarsen}}$  leads to a significant reduction in the number of cells in the finest, final grid. At the same time, the PSNR and SSIM values of the reconstruction remain relatively stable, resulting in a similarly appealing outcome, see Figure 7. Nevertheless, from Table 3 we observe that the quality measures slightly decrease with increasing  $n_{\text{coarsen}}$ , indicating a minor reduction in restoration capability. This might be attributed to the coarser grid, which may not allow the restored image to be represented with the same level of precision. Considering the CPU times in Table 3 we would like to emphasize that for  $n_{\text{coarsen}} \geq 1$  a significant amount of time is spent on the refinement process. In particular, the more cells are refined the slower the process. This is the reason why there is a jump between  $n_{\text{coarsen}} = 0$  and  $n_{\text{coarsen}} = 1$  in Figure 9(b). Hence the reduction in CPU time with increasing  $n_{\text{coarsen}}$  is then two-sided. First, due to the reduction of cells the used semi-smooth Newton method converges faster and second, the refinement process needs to refine less cells and hence is also faster. This results



**Fig. 9:** Number of cells, CPU time, PSNR and SSIM for different initial image resolutions coarsened by the factor  $\frac{1}{2^{n_{\text{coarsen}}/2}}$ ,  $n_{\text{coarsen}} \in \{0, 1, 2, \dots, 15\}$ .

in the proposed adaptive algorithm terminating faster as  $n_{\text{coarsen}}$  increases, compared to when  $n_{\text{coarsen}} = 0$ , albeit with a minor reduction in restoration ability.

## 4.2 Motion Estimation

Motion estimation is the problem of numerically computing the apparent motion field in a sequence of images. Following the discussion in [5] the motion field  $\vec{u}$  of two (smooth) images  $f_0, f_1$  may be computed solving model (1.1) with

$$T\vec{u} := \nabla f_w \cdot \vec{u}, \quad g := \nabla f_w \cdot \vec{u}_0 - (f_w - f_0),$$

where  $f_w(\vec{x}) := f_1(\vec{x} + \vec{u}_0(\vec{x}))$  is a warped version of  $f_1$  with  $\vec{u}_0$  being some smooth initial guess and  $\vec{x} \in \Omega$ . To account for large displacements an optical flow warping algorithm is introduced [5, Algorithm 3]. We combine the warping technique from [5, Algorithm 3] with adaptive refinement leading to Algorithm 2.

To approximately solve for  $\vec{u}_k$  in Algorithm 2, we use our model (1.1) by choosing  $T\vec{u} := \nabla f_{w,k-1} \cdot \vec{u}$  and  $g := \nabla f_{w,k-1} \cdot \vec{u}_{k-1} - (f_{w,k-1} - f_0)$  and utilize the primal-dual semi-smooth Newton method of [5]. Thereby we use the discrete space  $Z_h$  for the images  $f_0, f_1, f_{w,k}$ , except for the warping step, in which  $f_{w,k}(\vec{x}) = f_1(\vec{x} + \vec{u}_k(\vec{x}))$  is carried out at original image resolution by evaluating  $f_1$  using bicubic interpolation

---

**Algorithm 2** Optical flow algorithm with adaptive warping

---

*Parameters:* warping threshold  $\varepsilon_{\text{warp}}$ , parameters for [5, Algorithm 1]

*Input:* images  $f_0, f_1$ , initial guess  $\vec{u}_0$

*Output:* motion fields  $(\vec{u}_k)_k$

```
 $f_{w,0}(\vec{x}) = f_1(\vec{x} + \vec{u}_0(\vec{x}))$ 
for  $k = 1, 2, \dots$  do
    find approximate discrete solution  $\vec{u}_k$  to (1.1) using [5, Algorithm 1]
     $f_{w,k}(\vec{x}) = f_1(\vec{x} + \vec{u}_k(\vec{x}))$ 
    if  $\frac{\|f_{w,k-1} - f_0\|_{L^2} - \|f_{w,k} - f_0\|_{L^2}}{\|f_{w,k-1} - f_0\|_{L^2}} < \varepsilon_{\text{warp}}$  then
        refine mesh and reproject image data
    end if
end for
```

---

and in a second step projected (e.g. using `12.lagrange` or `12.pixel`) onto the current finite element space in order to capture more detailed displacement information. For  $g$  we instead use a cellwise linear discontinuous space to capture the discontinuous component  $\nabla f_{w,k-1}$ .

Loosely speaking, Algorithm 2 solves the linearized optical flow equation for  $u_k$  and warps the input data by the computed flow field until it no longer improves on the data difference  $f_{w,k} - f_0$ , which indicates displacement. In that case, the mesh is refined using the indicators from (3.6) or (3.9) together with  $\theta_{\text{mark}} = 0.5$  in the greedy Dörfler marking strategy and the process repeats, now including more detailed image data.

We note, that this approach to adaptivity allows us to start off with a coarse mesh and refine cells only if deemed necessary by the error indicator. In that respect it is different from the only other adaptive finite element methods for optical flow we are aware of, see [20, 21], where the mesh is initialized at fine image resolution first and iteratively coarsened only after a costly computation of the flow field and a suitable metric for adaptivity on this fine mesh has been established.

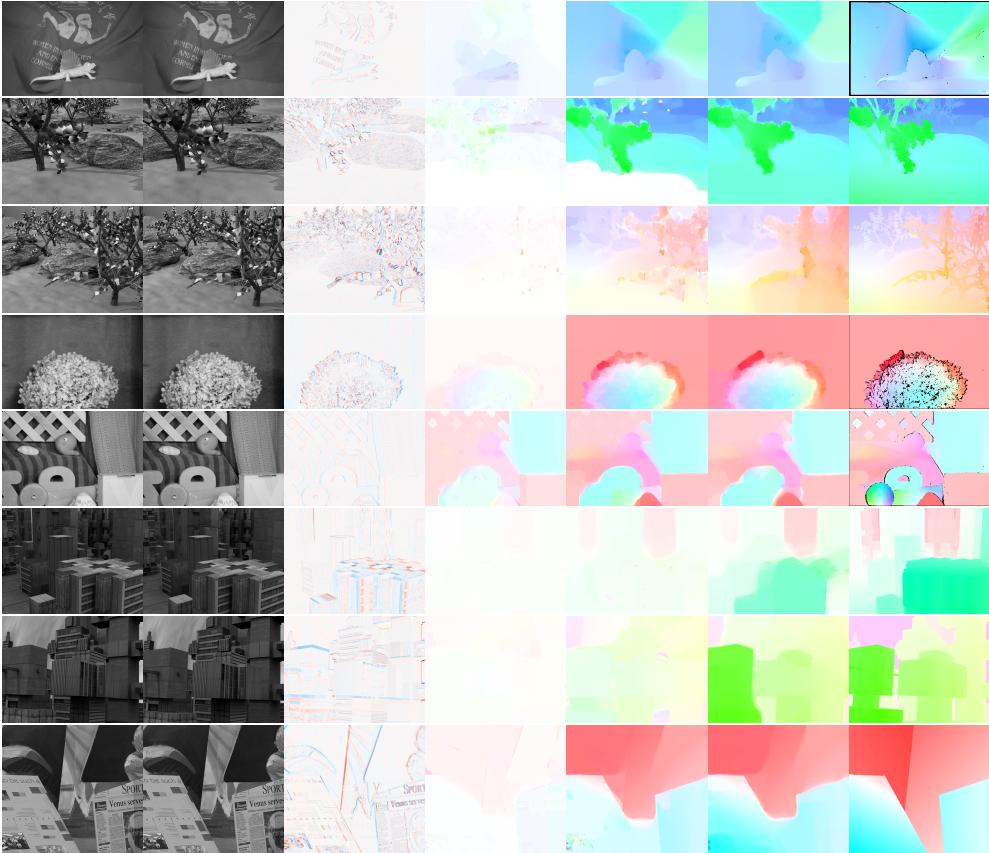
## Experiments

For all benchmarks the same manually chosen model parameters were used. We choose  $\alpha_1 = 10$ ,  $\alpha_2 = 0$ ,  $\lambda = 1$  to obtain visually pleasing results, cf. superiority of L1-TV in [32],  $\beta = 1 \cdot 10^{-5}$ ,  $\gamma_1 = 1 \cdot 10^{-4}$ ,  $\gamma_2 = 1 \cdot 10^{-4}$  to balance between speed and quality of the reconstruction and  $\vec{u}_0 := 0$ . For the interior method  $\varepsilon_{\text{newton}} = 1 \cdot 10^{-3}$  was chosen. The mesh is initialized to  $\frac{1}{8}$  of the image resolution, rounded down to integer values,  $\varepsilon_{\text{warp}} = 5 \cdot 10^{-2}$  and a constant number of 6 total refinements are carried out before stopping the algorithm.

In Figure 10 we evaluate Algorithm 2 visually against the middlebury optical flow benchmark [31] for  $S = \nabla$ , using the residual based indicators in the marking strategy and using `12.lagrange` for projecting onto the finite element space. The color-coded images representing optical flow fields are normalized by the maximum motion of the ground truth flow data and black areas of the ground truth data represent unknown flow information, e.g. due to occlusion. A good resemblance of the computed

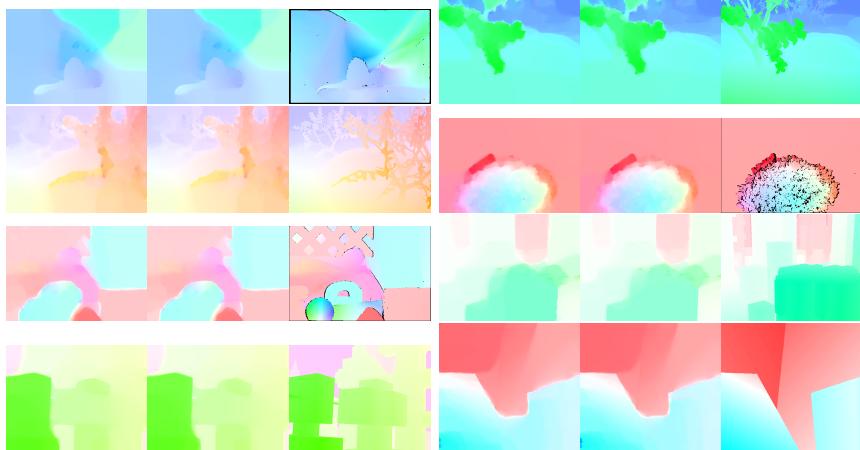
**Table 4:** Quantitative results for  $S = \nabla$  using the residual based indicators: Endpoint errors (EE) and angular errors (AE) as in [31], each given by mean and standard deviation, and computational times in seconds.

Benchmark	Algorithm	EE-mean	EE-stddev	AE-mean	AE-stddev	Time (s)
Dimetrodon	[5, Algorithm 3] without warping	1.80	0.75	0.84	0.22	211.00
	[5, Algorithm 3] with warping	0.25	0.23	0.08	0.07	2,147.03
	Algorithm 2 (with 12-lagrange)	0.41	0.37	0.13	0.12	83.96
	Algorithm 2 (with 12-pixel)	0.42	0.38	0.14	0.12	74.62
Grove2	[5, Algorithm 3] without warping	2.93	0.46	1.10	0.14	285.74
	[5, Algorithm 3] with warping	1.13	1.30	0.42	0.54	1,727.35
	Algorithm 2 (with 12 lagrange)	0.40	0.56	0.11	0.16	152.76
	Algorithm 2 (with 12-pixel)	0.39	0.56	0.11	0.16	124.45
Grove3	[5, Algorithm 3] without warping	3.69	2.33	1.01	0.32	312.74
	[5, Algorithm 3] with warping	1.93	2.11	0.26	0.29	1,628.13
	Algorithm 2 (with 12 lagrange)	1.12	1.50	0.16	0.29	168.02
	Algorithm 2 (with 12-pixel)	1.10	1.49	0.16	0.29	176.56
Hydrangea	[5, Algorithm 3] without warping	3.33	1.22	0.91	0.23	222.65
	[5, Algorithm 3] with warping	0.22	0.52	0.04	0.10	1,396.16
	Algorithm 2 (with 12 lagrange)	0.58	0.47	0.08	0.11	106.36
	Algorithm 2 (with 12-pixel)	0.56	0.44	0.07	0.11	83.33
RubberWhale	[5, Algorithm 3] without warping	0.67	0.58	0.36	0.25	263.60
	[5, Algorithm 3] with warping	0.26	0.52	0.14	0.26	633.41
	Algorithm 2 (with 12 lagrange)	0.37	0.59	0.20	0.33	80.61
	Algorithm 2 (with 12-pixel)	0.37	0.59	0.20	0.33	71.32
Urban2	[5, Algorithm 3] without warping	8.16	8.12	0.99	0.40	238.07
	[5, Algorithm 3] with warping	7.62	8.15	0.64	0.39	691.12
	Algorithm 2 (with 12 lagrange)	4.53	5.32	0.27	0.30	105.81
	Algorithm 2 (with 12-pixel)	5.01	5.79	0.30	0.32	88.97
Urban3	[5, Algorithm 3] without warping	7.04	4.45	1.13	0.24	237.47
	[5, Algorithm 3] with warping	5.56	4.45	0.50	0.40	1,173.47
	Algorithm 2 (with 12 lagrange)	1.72	2.35	0.25	0.50	137.35
	Algorithm 2 (with 12-pixel)	1.78	2.35	0.25	0.50	119.87
Venus	[5, Algorithm 3] without warping	3.55	1.75	1.00	0.24	125.81
	[5, Algorithm 3] with warping	1.25	1.53	0.23	0.35	1,056.74
	Algorithm 2 (with 12 lagrange)	0.72	1.13	0.15	0.34	81.04
	Algorithm 2 (with 12-pixel)	0.72	1.13	0.15	0.34	85.11

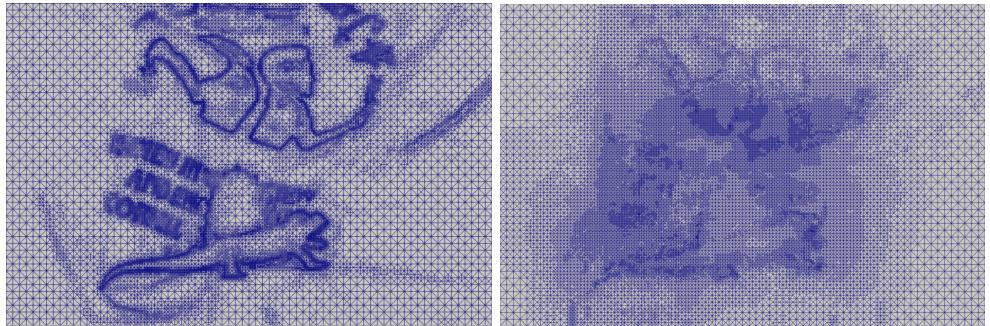


**Fig. 10:** Middlebury Optical Flow Benchmark for  $S = \nabla$  using the residual based indicators: columns from left to right:  $f_0$ ,  $f_1$ , image difference  $f_1 - f_0$ , computed optical flow  $\bar{u}$  using [5, Algorithm 3] without warping and with warping, computed optical flow  $\bar{u}$  using Algorithm 2 (adaptive warping), ground truth optical flow . Benchmarks from top to bottom: Dimetrodon, Grove2, Grove3, Hydrangea, RubberWhale, Urban2, Urban3, Venus.

optical flow to the ground truth and the effect of total variation regularization, i.e. sharp edges separating homogeneous regions, can be seen clearly. Large displacements are resolved quite well, e.g. the fast moving triangle-shaped object at the bottom of the RubberWhale benchmark, thanks to the warping algorithm employed. Using the same example, smaller slow-moving parts adjacent to the larger moving objects tend to get somewhat distorted however. For quantitatively evaluating our experiments we use the endpoint error and angular error defined as in [31] and summarize our findings in Table 4. Additionally we also compare in Table 4 the influence of the projections 12\_lagrange and 12\_pixel on Algorithm 2. In particular, it turns out that independently whether 12\_lagrange or 12\_pixel is used, the obtained result are nearly the same and visually hardly distinguishable, see Figure 11.



**Fig. 11:** Middlebury Optical Flow Benchmark (Dimetrodon, Grove2, Grove3, Hydrangea, RubberWhale, Urban2, Urban3, Venus): Algorithm 2 with 12\_lagrange (1<sup>st</sup> and 4<sup>th</sup> column), Algorithm 2 with 12\_pixel (2<sup>nd</sup> and 5<sup>th</sup> column) and respective ground truth (3<sup>rd</sup> and 6<sup>th</sup> column)



**Fig. 12:** Exemplary adapted mesh for Algorithm 2 on the Dimetrodon Middlebury optical flow benchmark, using the residual based indicators (left) and the primal-dual based indicators (right).

Remarkable, but maybe not surprisingly, is the speed-up of our adaptive warping technique, which can be clearly seen in Table 4. Hence Algorithm 2 not only improves qualitatively the computation of the optical flow of [5, Algorithm 3] but also dramatically shortens the computational time. In particular we see from Table 4 that Algorithm 2 is even faster than [5, Algorithm 3] without warping and additionally generates a qualitatively much better optical flow.

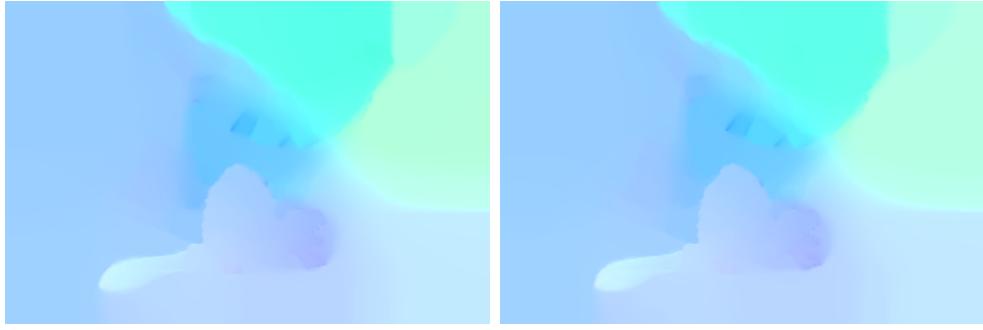
It is unclear how much visual improvement more careful or adaptive parameter selection may give and further study remains to be done. Exemplary, the adapted mesh for the Dimetrodon benchmark can be seen in Figure 12, where refinement seems to take place largely around image edges.

**Table 5:** Results for  $S = I$  using the residual based indicators: Endpoint errors (EE) and angular errors (AE) as in [31], each given by mean and standard deviation, and computational time in seconds.

Benchmark	Algorithm	EE-mean	EE-stddev	AE-mean	AE-stddev	Time (s)
Dimetrodon	[5, Algorithm 3] without warping	1.81	0.75	0.84	0.22	151.10
	[5, Algorithm 3] with warping	0.25	0.23	0.08	0.07	1,834.52
	Algorithm 2 (with 12-lagrange)	0.43	0.40	0.14	0.13	79.52
	Algorithm 2 (with 12-pixel)	0.43	0.39	0.14	0.12	60.89
Grove2	[5, Algorithm 3] without warping	2.93	0.46	1.10	0.14	195.36
	[5, Algorithm 3] with warping	1.13	1.30	0.42	0.54	1,689.12
	Algorithm 2 (with 12 lagrange)	0.40	0.57	0.11	0.16	134.80
	Algorithm 2 (with 12-pixel)	0.39	0.56	0.11	0.16	109.99
Grove3	[5, Algorithm 3] without warping	3.69	2.33	1.01	0.32	211.27
	[5, Algorithm 3] with warping	1.93	2.10	0.27	0.29	1,579.88
	Algorithm 2 (with 12 lagrange)	1.14	1.51	0.17	0.29	152.71
	Algorithm 2 (with 12-pixel)	1.12	1.49	0.16	0.29	151.63
Hydrangea	[5, Algorithm 3] without warping	3.33	1.22	0.91	0.23	151.16
	[5, Algorithm 3] with warping	0.22	0.52	0.04	0.10	1,303.31
	Algorithm 2 (with 12 lagrange)	0.60	0.50	0.08	0.12	86.75
	Algorithm 2 (with 12-pixel)	0.57	0.47	0.08	0.11	72.85
RubberWhale	[5, Algorithm 3] without warping	0.67	0.58	0.36	0.25	187.59
	[5, Algorithm 3] with warping	0.26	0.52	0.14	0.26	624.25
	Algorithm 2 (with 12 lagrange)	0.39	0.59	0.21	0.33	65.65
	Algorithm 2 (with 12-pixel)	0.39	0.60	0.21	0.33	57.27
Urban2	[5, Algorithm 3] without warping	8.16	8.12	0.99	0.40	212.86
	[5, Algorithm 3] with warping	7.62	8.15	0.64	0.39	637.57
	Algorithm 2 (with 12 lagrange)	4.52	5.26	0.28	0.32	83.84
	Algorithm 2 (with 12-pixel)	5.03	5.76	0.32	0.34	69.16
Urban3	[5, Algorithm 3] without warping	7.04	4.45	1.13	0.24	242.47
	[5, Algorithm 3] with warping	5.57	4.45	0.50	0.40	1,062.80
	Algorithm 2 (with 12 lagrange)	1.76	2.34	0.25	0.50	127.76
	Algorithm 2 (with 12-pixel)	1.82	2.35	0.26	0.50	100.39
Venus	[5, Algorithm 3] without warping	3.55	1.75	1.00	0.24	121.23
	[5, Algorithm 3] with warping	1.25	1.52	0.23	0.35	887.40
	Algorithm 2 (with 12 lagrange)	0.73	1.13	0.15	0.34	79.46
	Algorithm 2 (with 12-pixel)	0.75	1.14	0.15	0.34	80.27

**Table 6:** Results for  $S = \nabla$  obtained by Algorithm 2 (with 12.lagrange) using the primal-dual based indicators: Endpoint errors (EE) and angular errors (AE) as in [31], each given by mean and standard deviation, and computational time in seconds.

Benchmark	EE-mean	EE-stddev	AE-mean	AE-stddev	Time (s)
Dimetrodon	0.42	0.38	0.14	0.12	139.94
Grove2	0.39	0.55	0.11	0.16	249.05
Grove3	1.15	1.50	0.17	0.29	237.76
Hydrangea	0.52	0.44	0.07	0.10	151.24
RubberWhale	0.39	0.60	0.21	0.33	140.42
Urban2	4.38	5.16	0.29	0.38	175.19
Urban3	1.76	2.35	0.25	0.50	248.80
Venus	0.70	1.11	0.14	0.34	154.45



**Fig. 13:** Algorithm 2 with 12.lagrange applied on Dimetrodon: residual based indicators (left), primal-dual based indicators (right)

We note that for  $S = I$  very similar qualitative and quantitative results are obtained, see Table 5. A reason for this might be that we choose  $\beta = 1 \cdot 10^{-5}$ , which is rather small and hence the choice of  $S$  does not play a crucial role for numerically computing a solution. Indeed, as the associated term  $\frac{\beta}{2} \|S\vec{u}\|_{V_S}^2$  is only added for theoretical reasons, one typically wants to choose  $\beta$  very small [5].

Further, very similar qualitatively results are obtained when the primal-dual based indicators is used in the marking strategy instead of the residual based indicators, see Table 6. However, since the indicator needs to be computed on a globally refined mesh in each iteration, the computation times are larger than the ones when the residual based indicators is used. However, it is still in nearly all the experiments faster than [5, Algorithm 3] without warping. Interestingly, Algorithm 1 using the primal-dual based indicators generates a different mesh than when the residual based indicators are used, see Figure 12. However, the generated optical flows are visually hardly distinguishable, as we observe in the case of Dimetrodon, see Figure 13.

## Declarations

**Funding.** This work was partly supported by the Ministerium für Wissenschaft, Forschung und Kunst of Baden-Württemberg (Az: 7533.-30-10/56/1) through the RISC-project “Automatische Erkennung von bewegten Objekten in hochauflösenden Bildsequenzen mittels neuer Gebietszerlegungsverfahren”, by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany’s Excellence Strategy – EXC 2075 – 390740016, and by the Crafoord Foundation through the project “Locally Adaptive Methods for Free Discontinuity Problems”.

**Conflict of interest.** The authors have no competing interests to declare that are relevant to the content of this article.

**Code availability.** The code is available as a Julia package and can be found at <https://gitlab.mathematik.uni-stuttgart.de/stephan.hilb/SemiSmoothNewton.jl>.

## Appendix A Proof of Theorem 3.2

Denote  $\vec{e}_h := \vec{u}_h - \vec{u} \in V$  and bound using Lemma 2.2:

$$\begin{aligned} \frac{1}{2} \|\vec{u}_h - \vec{u}\|_B^2 &\leq E(\vec{u}_h) - E(\vec{u}) \\ &= F(\vec{u}_h) - F(\vec{u}) + G(\Lambda \vec{u}_h) - G(\Lambda \vec{u}) \\ &\leq \langle \partial F(\vec{u}_h), \vec{e}_h \rangle_{V^*, V} + \langle \partial G(\Lambda \vec{u}_h), \Lambda \vec{e}_h \rangle_{L^2} \\ &= a_B(\vec{u}_h, \vec{e}_h) - l(\vec{e}_h) + \langle \partial G(\Lambda \vec{u}_h), \Lambda \vec{e}_h \rangle_{L^2}. \end{aligned}$$

Depending on  $S$ , we will now further bound the error  $\frac{1}{2} \|\vec{u}_h - \vec{u}\|_B^2$ .

### Setting (S.ii): the Case $S = \nabla$

We will make use of the fact that  $a_B$  is coercive on  $V$ , see Assumption (As1), to derive an estimate on the error  $\|\vec{e}_h\|_{H^1(\Omega)^m}$ .

Due to (3.5) and the discrete optimality condition (3.4) we have for any bounded linear operator  $I_h : V \rightarrow V_h$ :

$$\begin{aligned} \frac{1}{2} \|\vec{u}_h - \vec{u}\|_B^2 &\leq a_B(\vec{u}_h, \vec{e}_h) - l(\vec{e}_h) + \langle \vec{p}_h, \Lambda \vec{e}_h \rangle_{L^2} \\ &= a_B(\vec{u}_h, \vec{e}_h - I_h \vec{e}_h) - l(\vec{e}_h - I_h \vec{e}_h) + \langle \vec{p}_h, \Lambda(\vec{e}_h - I_h \vec{e}_h) \rangle_{L^2} \\ &= \langle \alpha_2(T \vec{u}_h - g) + p_{h,1}, T(\vec{e}_h - I_h \vec{e}_h) \rangle_{L^2} \\ &\quad + \langle \beta S \vec{u}_h, S(\vec{e}_h - I_h \vec{e}_h) \rangle_{L^2} + \langle \vec{p}_{h,2}, \nabla(\vec{e}_h - I_h \vec{e}_h) \rangle_{L^2}. \end{aligned} \tag{A.1}$$

Actually we choose  $I_h : V \rightarrow V_h$  to be a quasi-interpolation operator which satisfies the interpolation estimates [10, Proposition 1.3] [33, Theorem 1.7]:

$$\|\vec{v} - I_h \vec{v}\|_{L^2(K)} \leq c_1 h_K \|\nabla \vec{v}\|_{L^2(\omega_K)}, \tag{A.2}$$

$$\|\vec{v} - I_h \vec{v}\|_{L^2(F)} \leq c_2 h_F^{\frac{1}{2}} \|\nabla \vec{v}\|_{L^2(\omega_F)}, \tag{A.3}$$

where  $c_1, c_2 > 0$ ,  $\omega_K, \omega_F$  denote the union of triangles which share a common vertex with the cell  $K$  or the facet  $F$  respectively and  $h_K, h_F$  denote the diameter of  $K$  and  $F$  respectively.

Utilizing Green's formula, see e.g. [34, Proposition 2.4], in (A.1) yields

$$\begin{aligned} \frac{1}{2}\|\vec{u}_h - \vec{u}\|_B^2 &\leq \langle \alpha_2(T\vec{u}_h - g) + p_{h,1}, T(\vec{e}_h - I_h\vec{e}_h) \rangle_{L^2} \\ &\quad + \langle \beta\nabla\vec{u}_h + \vec{p}_{h,2}, \nabla(\vec{e}_h - I_h\vec{e}_h) \rangle_{L^2} \\ &= \sum_{K \in \mathcal{T}} \left( \left\langle \alpha_2 T^*(T\vec{u}_h - g) + T^*p_{h,1} - \beta\Delta\vec{u}_h - \operatorname{div}\vec{p}_{h,2}, \vec{e}_h - I_h\vec{e}_h \right\rangle_{L^2(K)^m} \right. \\ &\quad \left. + \left\langle \vec{n}^\top(\beta\nabla\vec{u}_h + \vec{p}_{h,2}), \vec{e}_h - I_h\vec{e}_h \right\rangle_{L^2(\partial K)^m} \right). \end{aligned}$$

Denoting  $\vec{\xi} := \alpha_2 T^*(T\vec{u}_h - g) + T^*p_{h,1} - \beta\Delta\vec{u}_h - \operatorname{div}\vec{p}_{h,2} \in L^2(\Omega)^m$  and  $\vec{\zeta} := \beta\nabla\vec{u}_h + \vec{p}_{h,2} \in L^2(\Omega)^{d \times m}$  we obtain using interpolation estimates (A.2) and (A.3):

$$\begin{aligned} \frac{1}{2}\|\vec{u}_h - \vec{u}\|_B^2 &\leq \sum_{K \in \mathcal{T}} \left( \langle \vec{\xi}, \vec{e}_h - I\vec{e}_h \rangle_{L^2(K)^m} + \langle \vec{n}^\top \vec{\zeta}, \vec{e}_h - I\vec{e}_h \rangle_{L^2(\partial K)^m} \right) \\ &= \sum_{K \in \mathcal{T}} \langle \vec{\xi}, \vec{e}_h - I\vec{e}_h \rangle_{L^2(K)^m} + \sum_{F \in \Gamma} \langle [\vec{n}^\top \vec{\zeta}], \vec{e}_h - I\vec{e}_h \rangle_{L^2(F)^m} \\ &\leq \max\{c_1, c_2\} \left( \sum_{K \in \mathcal{T}} \|\nabla\vec{e}_h\|_{L^2(\omega_K)^{d \times m}} h_K \|\vec{\xi}\|_{L^2(K)^m} \right. \\ &\quad \left. + \sum_{F \in \Gamma} \|\nabla\vec{e}_h\|_{L^2(\omega_F)^{d \times m}} h_F^{\frac{1}{2}} \|[\vec{n}^\top \vec{\zeta}]\|_{L^2(F)^m} \right) \\ &\leq \max\{c_1, c_2\} \left( \sum_{K \in \mathcal{T}} \|\nabla\vec{e}_h\|_{L^2(\omega_K)^{d \times m}}^2 h_K^2 + \sum_{F \in \Gamma} \|\nabla\vec{e}_h\|_{L^2(\omega_F)^{d \times m}}^2 h_F^2 \right)^{\frac{1}{2}} \\ &\quad \left( \sum_{K \in \mathcal{T}} h_K^2 \|\vec{\xi}\|_{L^2(K)^m}^2 + \sum_{F \in \Gamma} h_F \|[\vec{n}^\top \vec{\zeta}]\|_{L^2(F)^m}^2 \right)^{\frac{1}{2}}. \end{aligned}$$

Let  $B(K)$  be the set of facets of  $K$  belonging to  $\partial\Omega$  and  $I(K)$  all other facets of  $K$ . Realizing that  $\omega_F \subset \omega_K$  for all to  $F$  adjacent cells  $K$  and having at most one boundary

facet at each cell we may bound

$$\begin{aligned}
& \sum_{F \in \Gamma} \|\nabla \vec{e}_h\|_{L^2(\omega_F)^{d \times m}}^2 \\
&= \sum_{K \in \mathcal{T}} \left( \sum_{F \in B(K)} \|\nabla \vec{e}_h\|_{L^2(\omega_F)^{d \times m}}^2 + \frac{1}{2} \sum_{F \in I(K)} \|\nabla \vec{e}_h\|_{L^2(\omega_F)^{d \times m}}^2 \right) \\
&\leq \sum_{K \in \mathcal{T}} \left( \sum_{F \in B(K)} \|\nabla \vec{e}_h\|_{L^2(\omega_K)^{d \times m}}^2 + \frac{1}{2} \sum_{F \in I(K)} \|\nabla \vec{e}_h\|_{L^2(\omega_K)^{d \times m}}^2 \right) \\
&\leq 2 \sum_{K \in \mathcal{T}} \|\nabla \vec{e}_h\|_{L^2(\omega_K)^{d \times m}}^2 \leq 2c_{\mathcal{T}}^2 \|\nabla \vec{e}_h\|_{L^2(\Omega)^{d \times m}}^2,
\end{aligned} \tag{A.4}$$

where  $c_{\mathcal{T}}$  denotes the shape regularity constant of the mesh independent of the mesh size. We thus arrive at

$$\begin{aligned}
& \frac{1}{2} \|\vec{u}_h - \vec{u}\|_B^2 \\
&\leq \max\{c_1, c_2\} \left( 3 \sum_{K \in \mathcal{T}} \|\nabla \vec{e}_h\|_{L^2(\omega_K)^{d \times m}}^2 \right)^{\frac{1}{2}} \left( \sum_{K \in \mathcal{T}} h_K^2 \|\vec{\xi}\|_{L^2(K)^m}^2 \right. \\
&\quad \left. + \sum_{F \in \Gamma} h_F \|[\vec{n}^\top \vec{\zeta}]\|_{L^2(F)^m}^2 \right)^{\frac{1}{2}} \\
&\leq \sqrt{3} \max\{c_1, c_2\} c_{\mathcal{T}} \|\nabla \vec{e}_h\|_{L^2(\Omega)^{d \times m}} \left( \sum_{K \in \mathcal{T}} h_K^2 \|\vec{\xi}\|_{L^2(K)^m}^2 \right. \\
&\quad \left. + \sum_{F \in \Gamma} h_F \|[\vec{n}^\top \vec{\zeta}]\|_{L^2(F)^m}^2 \right)^{\frac{1}{2}} \\
&\leq \sqrt{3} \max\{c_1, c_2\} c_{\mathcal{T}} \|\vec{e}_h\|_{H^1(\Omega)^m} \left( \sum_{K \in \mathcal{T}} h_K^2 \|\vec{\xi}\|_{L^2(K)^m}^2 \right. \\
&\quad \left. + \sum_{F \in \Gamma} h_F \|[\vec{n}^\top \vec{\zeta}]\|_{L^2(F)^m}^2 \right)^{\frac{1}{2}}.
\end{aligned}$$

Together with coercivity  $\|\vec{v}\|_B^2 = a_B(\vec{v}, \vec{v}) \geq c_B \|\vec{v}\|_V^2 = c_B \|\vec{v}\|_{H^1(\Omega)^m}^2$  from Assumption (As1) we arrive at an a-posteriori error bound

$$\begin{aligned}
\|\vec{u}_h - \vec{u}\|_{H^1(\Omega)^m}^2 &\leq C \left( \sum_{K \in \mathcal{T}} h_K^2 \|\alpha_2 T^*(T \vec{u}_h - g) + T^* p_{h,1} - \beta \Delta \vec{u}_h - \operatorname{div} \vec{p}_{h,2}\|_{L^2(K)^m}^2 \right. \\
&\quad \left. + \sum_{F \in \Gamma} h_F \|[\vec{n}^\top (\beta \nabla \vec{u}_h + \vec{p}_{h,2})]\|_{L^2(F)^m}^2 \right)
\end{aligned}$$

with constant  $C := \frac{12}{c_B} \max\{c_1^2, c_2^2\} c_{\mathcal{T}}^2$ .

### Setting (S.i): the case $S = I$

Using the fact that  $a_B$  is coercive in this setting with respect to  $L^2$ , see Assumption (As1), we derive an estimate for the error  $\|\vec{e}_h\|_{L^2(\Omega)^m}$ .

Let  $I_h : L^2(\Omega)^m \rightarrow V_h$  be a bounded and linear quasi-interpolation operator satisfying the interpolation estimate [33, Theorem 1.7]:

$$\|\vec{v} - I_h \vec{v}\|_{L^2(F)} \leq c_3 h_F^{-\frac{1}{2}} \|\vec{v}\|_{L^2(\omega_F)}, \quad (\text{A.5})$$

where  $c_3 > 0$ . Then, as above in Setting (S.ii), cf. (A.1), we obtain from (3.5) and the discrete optimality condition (3.4) that we have

$$\begin{aligned} \frac{1}{2} \|\vec{u}_h - \vec{u}\|_B^2 &= \langle \alpha_2(T\vec{u}_h - g) + p_{h,1}, T(\vec{e}_h - I_h \vec{e}_h) \rangle_{L^2} \\ &\quad + \langle \beta S \vec{u}_h, S(\vec{e}_h - I_h \vec{e}_h) \rangle_{L^2} + \langle \vec{p}_{h,2}, \nabla(\vec{e}_h - I_h \vec{e}_h) \rangle_{L^2}. \end{aligned} \quad (\text{A.6})$$

Applying Green's formula to (A.6) yields the estimate

$$\begin{aligned} \frac{1}{2} a_B(\vec{e}_h, \vec{e}_h) &\leq \langle \alpha_2 T^*(T\vec{u}_h - g) + T^* p_{h,1} + \beta \vec{u}_h, \vec{e}_h - I_h \vec{e}_h \rangle_{L^2} \\ &\quad + \langle \vec{p}_{h,2}, \nabla(\vec{e}_h - I_h \vec{e}_h) \rangle_{L^2} \\ &\leq \sum_{K \in \mathcal{T}} \left( \left\langle \alpha_2 T^*(T\vec{u}_h - g) + T^* p_{h,1} + \beta \vec{u}_h - \operatorname{div} \vec{p}_{h,2}, \vec{e}_h - I_h \vec{e}_h \right\rangle_{L^2(K)^m} \right. \\ &\quad \left. + \left\langle \vec{n}^\top \vec{p}_{h,2}, \vec{e}_h - I_h \vec{e}_h \right\rangle_{L^2(\partial K)^m} \right). \end{aligned}$$

Denoting  $\vec{\xi} := \alpha_2 T^*(T\vec{u}_h - g) + T^* p_{h,1} + \beta \vec{u}_h - \operatorname{div} \vec{p}_{h,2}$  and  $\vec{\zeta} := \vec{n}^\top \vec{p}_{h,2}$  for brevity one continues to derive using (A.5), Cauchy-Schwarz inequality and (A.4):

$$\begin{aligned} \frac{1}{2} a_B(\vec{e}_h, \vec{e}_h) &\leq \sum_{K \in \mathcal{T}} \|\vec{\xi}\|_{L^2(K)^m} \|\vec{e}_h - I_h \vec{e}_h\|_{L^2(K)^m} + \sum_{F \in \Gamma} \|[\vec{\zeta}]\|_{L^2(F)^m} \|\vec{e}_h - I_h \vec{e}_h\|_{L^2(F)^m} \\ &\stackrel{(\text{A.5})}{\leq} \sum_{K \in \mathcal{T}} \|\vec{\xi}\|_{L^2(K)^m} \|\vec{e}_h - I_h \vec{e}_h\|_{L^2(K)^m} + \sum_{F \in \Gamma} \|[\vec{\zeta}]\|_{L^2(F)^m} h_F^{-1/2} c_3 \|\vec{e}_h\|_{L^2(\omega_F)^m} \\ &\stackrel{\text{C.-S. ineq.}}{\leq} \left( \sum_{K \in \mathcal{T}} \|\vec{e}_h - I_h \vec{e}_h\|_{L^2(K)^m}^2 + \sum_{F \in \Gamma} c_3^2 \|\vec{e}_h\|_{L^2(\omega_F)^m}^2 \right)^{\frac{1}{2}} \left( \sum_{K \in \mathcal{T}} \|\vec{\xi}\|_{L^2(K)^m}^2 \right. \\ &\quad \left. + \sum_{F \in \Gamma} h_F^{-1} \|[\vec{\zeta}]\|_{L^2(F)^m}^2 \right)^{\frac{1}{2}} \\ &\leq \left( \sum_{K \in \mathcal{T}} \tilde{c} \|\vec{e}_h\|_{L^2(K)^m}^2 + \sum_{F \in \Gamma} c_3^2 \|\vec{e}_h\|_{L^2(\omega_F)^m}^2 \right)^{\frac{1}{2}} \left( \sum_{K \in \mathcal{T}} \|\vec{\xi}\|_{L^2(K)^m}^2 \right. \\ &\quad \left. + \sum_{F \in \Gamma} h_F^{-1} \|[\vec{\zeta}]\|_{L^2(F)^m}^2 \right)^{\frac{1}{2}} \end{aligned}$$

$$\begin{aligned}
& + \sum_{F \in \Gamma} h_F^{-1} \|[\vec{\zeta}]^2\|_{L^2(F)^m}^2 \Bigg)^{\frac{1}{2}} \\
& \stackrel{(A.4)}{\leq} \left( \sum_{K \in \mathcal{T}} \tilde{c} \|\vec{e}_h\|_{L^2(\omega_K)^m}^2 + 2c_3^2 \|\vec{e}_h\|_{L^2(\omega_K)^m}^2 \right)^{\frac{1}{2}} \left( \sum_{K \in \mathcal{T}} \|\vec{\xi}\|_{L^2(K)^m}^2 \right. \\
& \quad \left. + \sum_{F \in \Gamma} h_F^{-1} \|[\vec{\zeta}]^2\|_{L^2(F)^m}^2 \right)^{\frac{1}{2}} \\
& \stackrel{(A.4)}{\leq} (\tilde{c} + 2c_3^2)^{\frac{1}{2}} c_{\mathcal{T}} \|\vec{e}_h\|_{L^2(\Omega)^m} \left( \sum_{K \in \mathcal{T}} \|\vec{\xi}\|_{L^2(K)^m}^2 + \sum_{F \in \Gamma} h_F^{-1} \|[\vec{\zeta}]^2\|_{L^2(F)^m}^2 \right)^{\frac{1}{2}},
\end{aligned}$$

where  $\tilde{c} > 0$ . Together with the  $L^2$ -coercivity of  $a_B$  from Assumption (As1) we arrive at

$$\begin{aligned}
\|\vec{u}_h - \vec{u}\|_{L^2(\Omega)^m}^2 & \leq C \left( \sum_{K \in \mathcal{T}} \|\alpha_2 T^*(T\vec{u}_h - g) + T^* p_{h,1} + \beta \vec{u}_h - \operatorname{div} \vec{p}_{h,2}\|_{L^2(K)^m}^2 \right. \\
& \quad \left. + \sum_{F \in \Gamma} h_F^{-1} \|[\vec{n}^\top \vec{p}_{h,2}]\|_{L^2(F)^m}^2 \right)
\end{aligned}$$

for some constant  $C > 0$ .

## Appendix B Proof of Theorem 3.3

### B.1 Auxiliary Result

To proof Theorem 3.3 we need the following auxiliary result:

**Lemma B.1.** *Let  $H$  be a real Hilbert space with inner product  $\langle \cdot, \cdot \rangle_H$  and associated norm  $\|\cdot\|_H$ . Then for every  $v, w_h, w \in H$  one has*

$$\begin{aligned}
\|v - w\|_H^2 - \|v - w_h\|_H^2 & = \langle 2v - w_h - w, w_h - w \rangle_H \\
& \leq (\|v - w\|_H + \|v - w_h\|_H) \|w - w_h\|_H.
\end{aligned}$$

*Proof.* We obtain

$$\begin{aligned}
\|v - w\|_H^2 - \|v - w_h\|_H^2 & = \langle v - w, v - w \rangle_H - \langle v - w_h, v - w + w - w_h \rangle_H \\
& = \langle w_h - w, v - w \rangle_H - \langle v - w_h, w - w_h \rangle_H \\
& = \langle v - w, w_h - w \rangle_H + \langle v - w_h, w_h - w \rangle_H \\
& = \langle 2v - w - w_h, w_h - w \rangle_H,
\end{aligned}$$

which shows the equality in Lemma B.1. For the inequality one has

$$\langle 2v - w_h - w, w_h - w \rangle_H = \langle v - w_h, w_h - w \rangle_H + \langle v - w, w_h - w \rangle_H$$

$$\leq (\|v - w\|_H + \|v - w_h\|_H) \|w - w_h\|_H.$$

□

We may apply Lemma B.1 in particular to the weighted inner product  $\langle \cdot, B^{-1} \cdot \rangle_{V^*, V}$  on  $V^*$  with associated norm  $\|\cdot\|_{B^{-1}}$ .

Now we are ready to state the proof of Theorem 3.3:

## B.2 Proof

Let  $\vec{p} \in W^*$  be a solution to (2.2) and  $\vec{p}_h \in W_h^*$  be a discrete minimizer of  $D_h$ . Let  $v_h \in V_h$  and  $\vec{q}_h \in W_h^*$  be arbitrary. Then we have due to Lemma 2.2, strong duality from the Fenchel-duality theorem [27, Remark III.4.2], optimality of  $p$  in  $D$  and optimality of  $\vec{u}_h$  and  $\vec{p}_h$  in  $E_h$  and  $D_h$  respectively:

$$\begin{aligned} \frac{1}{2} \|\vec{u}_h - \vec{u}\|_B^2 &\leq E(\vec{u}_h) - E(u) = E(\vec{u}_h) + D(p) \leq E(\vec{u}_h) + D(\vec{p}_h) \\ &= \eta_h^2(\vec{u}_h, \vec{p}_h) + E(\vec{u}_h) - E_h(\vec{u}_h) + D(\vec{p}_h) - D_h(\vec{p}_h) \\ &\leq \eta_h^2(v_h, \vec{q}_h) + E(\vec{u}_h) - E_h(\vec{u}_h) + D(\vec{p}_h) - D_h(\vec{p}_h). \end{aligned}$$

We break up the terms of

$$\begin{aligned} \eta_h^2(v_h, \vec{q}_h) &= E_h(v_h) + D_h(\vec{q}_h) \\ &= \alpha_1 \int_{\Omega} \varphi_1(|Tv_h - g_h|) \, d\vec{x} + \frac{\alpha_2}{2} \|Tv_h - g_h\|_{L^2}^2 + \frac{\beta}{2} \|Sv_h\|_{L^2}^2 \\ &\quad + \lambda \int_{\Omega} \varphi_{\gamma_2}(|\nabla v_h|_F) \, d\vec{x} + \frac{1}{2} \|\Lambda^* \vec{q}_h - \alpha_2 T^* g_h\|_{B^{-1}}^2 - \frac{\alpha_2}{2} \|g_h\|_{L^2}^2 \\ &\quad + \langle g_h, q_{h,1} \rangle_{L^2} + \frac{\gamma_1}{2\alpha_1} \|q_{h,1}\|_{L^2}^2 + \frac{\gamma_2}{2\lambda} \|\vec{q}_{h,2}\|_{L^2}^2. \end{aligned}$$

Expanding

$$\begin{aligned} \alpha_1 \int_{\Omega} \varphi_1(|Tv_h - g_h|) \, d\vec{x} + \frac{\gamma_1}{2\alpha_1} \|q_{h,1}\|_{L^2}^2 \\ &= \alpha_1 \int_{\Omega} \varphi_1(|Tv_h - g_h|) \, d\vec{x} - \langle Tv_h - g_h, q_{h,1} \rangle_{L^2} + \frac{\gamma_1}{2\alpha_1} \|q_{h,1}\|_{L^2}^2 + \langle Tv_h - g_h, q_{h,1} \rangle_{L^2} \\ &\quad \lambda \int_{\Omega} \varphi_{\gamma_2}(|\nabla v_h|_F) \, d\vec{x} + \frac{\gamma_2}{2\lambda} \|\vec{q}_{h,2}\|_{L^2}^2 \\ &= \lambda \int_{\Omega} \varphi_{\gamma_2}(|\nabla v_h|_F) \, d\vec{x} - \langle \nabla v_h, q_{h,2} \rangle_{L^2} + \frac{\gamma_2}{2\lambda} \|\vec{q}_{h,2}\|_{L^2}^2 + \langle \nabla v_h, \vec{q}_{h,2} \rangle_{L^2} \end{aligned}$$

takes care of the first summands in (3.8). The remaining terms of the above two equations add up to

$$\begin{aligned} \frac{1}{2} \|\Lambda^* \vec{q}_h - \alpha_2 T^* g_h\|_{B^{-1}}^2 + \frac{\alpha_2}{2} \|Tv_h - g_h\|_{L^2}^2 - \frac{\alpha_2}{2} \|g_h\|_{L^2}^2 + \frac{\beta}{2} \|Sv_h\|_{L^2}^2 \\ + \langle g_h, q_{h,1} \rangle_{L^2} + \langle Tv_h - g_h, q_{h,1} \rangle_{L^2} + \langle \nabla v_h, \vec{q}_{h,2} \rangle_{L^2} \end{aligned}$$

$$\begin{aligned}
&= \frac{1}{2} \|B^{-1}(\Lambda^* \vec{q}_h - \alpha_2 T^* g_h)\|_B^2 + \frac{\alpha_2}{2} \|Tv_h\|_{L^2}^2 - \alpha_2 \langle Tv_h, g_h \rangle_{L^2} + \frac{\beta}{2} \|Sv_h\|_{L^2}^2 \\
&\quad + \langle v_h, T^* q_{h,1} \rangle_{L^2} + \langle v_h, \nabla^* \vec{q}_{h,2} \rangle_{V,V^*} \\
&= \frac{1}{2} \|B^{-1}(\Lambda^* \vec{q}_h - \alpha_2 T^* g_h)\|_B^2 + \frac{1}{2} \|v_h\|_B^2 + \langle v_h, BB^{-1}(\Lambda^* \vec{q}_h - \alpha_2 T^* g_h) \rangle_{V,V^*} \\
&= \frac{1}{2} \|B^{-1}(\Lambda^* \vec{q}_h - \alpha_2 T^* g_h) + v_h\|_B^2
\end{aligned}$$

which constitutes the last summand in (3.8).

It remains to bound the data approximation errors  $E(\vec{u}_h) - E_h(\vec{u}_h)$  and  $D(\vec{p}_h) - D_h(\vec{p}_h)$ .

According to [5, Proposition 3.3 (iii)] the Huber function satisfies  $|\varphi'_{\gamma_1}(x)| \leq 1$  for all  $x \in \mathbb{R}$ , so  $\varphi_{\gamma_1}$  is in particular Lipschitz-continuous with Lipschitz-constant 1 and we may use the inverse triangle inequality to obtain

$$\varphi_{\gamma_1}(|T\vec{u}_h - g|) - \varphi_{\gamma_1}(|T\vec{u}_h - g_h|) \leq | |T\vec{u}_h - g| - |T\vec{u}_h - g_h| | \leq |g - g_h|. \quad (\text{B.1})$$

Further, since  $g_h$  is the projection of  $g$  onto the closed and convex set  $Z_h$ , i.e.  $\|g_h - g\|_{L^2} \leq \|v - g\|_{L^2}$  for all  $v \in Z_h$ , we have that

$$\|g_h\|_{L^2} \leq \|g\|_{L^2} \quad ([\text{35}, 6.54 \text{ Lemma d.}]) \quad (\text{B.2})$$

$$\|g_h - g\|_{L^2} \leq \|g\|_{L^2}, \quad (\text{B.3})$$

$$\|T\vec{u}_h - g\|_{L^2} \leq \|T\vec{u}_h - g_h\|_{L^2} + \|g_h - g\|_{L^2}. \quad (\text{B.4})$$

Using (B.1)-(B.4) and Lemma B.1 we estimate

$$\begin{aligned}
E(\vec{u}_h) - E_h(\vec{u}_h) &= \alpha_1 \int_{\Omega} \varphi_{\gamma_1}(|T\vec{u}_h - g|) - \varphi_{\gamma_1}(|T\vec{u}_h - g_h|) \, d\vec{x} \\
&\quad + \frac{\alpha_2}{2} (\|T\vec{u}_h - g\|_{L^2}^2 - \|T\vec{u}_h - g_h\|_{L^2}^2) \\
&\leq \alpha_1 \int_{\Omega} |g - g_h| \, d\vec{x} + \frac{\alpha_2}{2} (\|T\vec{u}_h - g\|_{L^2} + \|T\vec{u}_h - g_h\|_{L^2}) \|g - g_h\|_{L^2} \\
&\leq \alpha_1 \int_{\Omega} |g - g_h| \, d\vec{x} + \frac{\alpha_2}{2} (2\|T\vec{u}_h - g_h\|_{L^2} + \|g - g_h\|_{L^2}) \|g - g_h\|_{L^2} \\
&\leq \alpha_1 |\Omega|^{\frac{1}{2}} \|g - g_h\|_{L^2} + \frac{\alpha_2}{2} (2E_h(0)^{\frac{1}{2}} + \|g\|_{L^2}) \|g - g_h\|_{L^2} \\
&= c_E \|g - g_h\|_{L^2}
\end{aligned}$$

with  $c_E := \alpha_1 |\Omega|^{\frac{1}{2}} + \alpha_2 E_h(0)^{\frac{1}{2}} + \frac{\alpha_2}{2} \|g\|_{L^2} > 0$  being a constant. Using [5, Proposition 3.3], (B.2), and Hölder's inequality we may further bound  $E_h(0)$  independent of  $g_h$ :

$$E_h(0) = \alpha_1 \int_{\Omega} \varphi_{\gamma_1}(|g_h|) + \frac{\alpha_2}{2} \|g_h\|_{L^2}^2 \leq \alpha_1 \|g_h\|_{L^1} + \frac{\alpha_2}{2} \|g\|_{L^2}^2 \leq \alpha_1 |\Omega|^{\frac{1}{2}} \|g\|_{L^2} + \frac{\alpha_2}{2} \|g\|_{L^2}^2.$$

By Lemma B.1 for the predual data approximation error one has

$$\begin{aligned}
D(\vec{p}_h) - D_h(\vec{p}_h) &= \frac{1}{2} \|\Lambda^* \vec{p}_h - \alpha_2 T^* g\|_{B^{-1}}^2 - \frac{1}{2} \|\Lambda^* \vec{p}_h - \alpha_2 T^* g_h\|_{B^{-1}}^2 \\
&\quad + \frac{\alpha_2}{2} (\|g_h\|_{L^2}^2 - \|g\|_{L^2}^2) + \langle g - g_h, p_{h,1} \rangle_{L^2}
\end{aligned}$$

$$\begin{aligned}
&\leq \frac{1}{2} \left( \|\Lambda^* \vec{p}_h - \alpha_2 T^* g\|_{B^{-1}} + \|\Lambda^* \vec{p}_h - \alpha_2 T^* g_h\|_{B^{-1}} \right) \|g - g_h\|_{B^{-1}} \\
&\quad + \frac{\alpha_2}{2} \langle g_h - g, g_h + g \rangle + \|p_{h,1}\|_{L^2} \|g - g_h\|_{L^2} \\
&\leq \frac{1}{2} \left( 2 \|\Lambda^* \vec{p}_h - \alpha_2 T^* g\|_{B^{-1}} + \|\alpha_2 T^* g - \alpha_2 T^* g_h\|_{B^{-1}} \right) \|g - g_h\|_{B^{-1}} \\
&\quad + \frac{\alpha_2}{2} \|g_h - g\|_{L^2} \|g_h + g\|_{L^2} + |\Omega|^{\frac{1}{2}} \|p_{h,1}\|_{L^\infty} \|g - g_h\|_{L^2} \\
&\leq \frac{1}{2} \left( 2 D_h(0)^{\frac{1}{2}} + \alpha_2 \|B^{-1}\|^{\frac{1}{2}} \|T\| \|g - g_h\|_{L^2} \right) \|B^{-1}\|^{\frac{1}{2}} \|g - g_h\|_{L^2} \\
&\quad + \alpha_2 \|g\|_{L^2} \|g_h - g\|_{L^2} + |\Omega|^{\frac{1}{2}} \alpha_1 \|g - g_h\|_{L^2} \\
&= c_D \|g - g_h\|_{L^2}
\end{aligned}$$

with constant  $c_D := \|B^{-1}\|^{\frac{1}{2}} D_h(0)^{\frac{1}{2}} + \frac{\alpha_2}{2} \|B^{-1}\| \|T\| \|g\|_{L^2} + \alpha_2 \|g\|_{L^2} + |\Omega|^{\frac{1}{2}} \alpha_1 > 0$ , where  $D_h(0)$  may be also bounded independent of  $g_h$ :

$$\begin{aligned}
D_h(0) &= \frac{1}{2} \|\alpha_2 T^* g_h\|_{B^{-1}}^2 - \frac{\alpha_2}{2} \|g_h\|_{L^2}^2 \\
&\leq \frac{\alpha_2^2}{2} \|T\|^2 \|B^{-1}\| \|g_h\|_{L^2}^2 - \frac{\alpha_2}{2} \|g_h\|_{L^2}^2 \\
&\leq \max\{0, \frac{\alpha_2}{2} (\alpha_2 \|T\|^2 \|B^{-1}\| - 1)\} \|g\|_{L^2}.
\end{aligned}$$

We finish the proof by combining both results:

$$\frac{1}{2} \|\vec{u}_h - \vec{u}\|_B^2 \leq \eta_h^2(v_h, \vec{q}_h) + (c_E + c_D) \|g_h - g\|_{L^2}$$

with  $c := c_E + c_D$  independent of  $g_h$ .

## References

- [1] Hintermüller, M., Langer, A.: Subspace correction methods for a class of nonsmooth and nonadditive convex variational problems with mixed  $L^1/L^2$  data-fidelity in image processing. SIAM Journal on Imaging Sciences **6**(4), 2134–2173 (2013) <https://doi.org/10.1137/120894130>
- [2] Langer, A.: Automated parameter selection in the  $L^1$ - $L^2$ -TV model for removing Gaussian plus impulse noise. Inverse Problems **33**(7), 074002 (2017)
- [3] Langer, A.: Locally adaptive total variation for removing mixed Gaussian–impulse noise. International Journal of Computer Mathematics **96**(2), 298–316 (2019)
- [4] Alkämper, M., Langer, A.: Using DUNE-ACFem for non-smooth minimization of bounded variation functions. Archive of Numerical Software **5**(1), 3–19 (2017) <https://doi.org/10.11588/ans.2017.1.27475>
- [5] Hilb, S., Langer, A., Alkämper, M.: A primal-dual finite element method for scalar and vectorial total variation minimization. Journal of Scientific Computing **96**(1), 24 (2023)

- [6] Hunter, J.K., Nachtergaelie, B.: Applied Analysis. World Scientific Publishing Company, Singapore (2001)
- [7] Einsiedler, M., Ward, T.: Functional Analysis, Spectral Theory, and Applications vol. 276. Springer, Cham (2017)
- [8] Chambolle, A., Pock, T.: Approximating the total variation with finite differences or finite elements. In: Handbook of Numerical Analysis vol. 22, pp. 383–417. Elsevier, Amsterdam (2021)
- [9] Nochetto, R.H., Siebert, K.G., Veeser, A.: Theory of adaptive finite element methods: An introduction. In: Multiscale, Nonlinear and Adaptive Approximation, pp. 409–542. Berlin: Springer, Berlin (2009). [https://doi.org/10.1007/978-3-642-03413-8\\_12](https://doi.org/10.1007/978-3-642-03413-8_12)
- [10] Verfürth, R.: A Posteriori Error Estimation Techniques for Finite Element Methods. OUP Oxford, Oxford (2013)
- [11] Bänsch, E., Mikula, K.: A coarsening finite element strategy in image selective smoothing. Computing and Visualization in Science **1**(1), 53–61 (1997)
- [12] Bazan, C., Blomgren, P.: Adaptive finite element method for image processing. In: Proceedings of COMSOL Multiphysics Conference, pp. 377–381 (2005)
- [13] Preußer, T., Rumpf, M.: An adaptive finite element method for large scale image processing. Journal of Visual Communication and Image Representation **11**(2), 183–195 (2000)
- [14] Catté, F., Lions, P.-L., Morel, J.-M., Coll, T.: Image selective smoothing and edge detection by nonlinear diffusion. SIAM Journal on Numerical Analysis **29**(1), 182–193 (1992) <https://doi.org/10.1137/0729012>
- [15] Bartels, S.: Error control and adaptivity for a variational model problem defined on functions of bounded variation. Mathematics of Computation **84**(293), 1217–1240 (2015)
- [16] Bartels, S., Kaltenbach, A.: Exact a posteriori error control for variational problems via convex duality and explicit flux reconstruction. arXiv preprint arXiv:2402.06429 (2024)
- [17] Bartels, S., Milicevic, M.: Primal-dual gap estimators for a posteriori error analysis of nonsmooth minimization problems. ESAIM: Mathematical Modelling and Numerical Analysis **54**(5), 1635–1660 (2020)
- [18] Brezzi, F., Douglas, J., Marini, L.D.: Two families of mixed finite elements for second order elliptic problems. Numerische Mathematik **47**, 217–235 (1985) <https://doi.org/10.1007/BF01389710>

- [19] Hintermüller, M., Rincon-Camacho, M.: An adaptive finite element method in  $L^2$ -tv-based image denoising. *Inverse Problems and Imaging* **8**(3), 685–711 (2014) <https://doi.org/10.3934/ipi.2014.8.685>
- [20] Belhachmi, Z., Hecht, F.: Control of the effects of regularization on variational optic flow computations. *Journal of Mathematical Imaging and Vision* **40**(1), 1–19 (2011) <https://doi.org/10.1007/s10851-010-0239-x>
- [21] Belhachmi, Z., Hecht, F.: An adaptive approach for the segmentation and the TV-filtering in the optic flow estimation. *Journal of Mathematical Imaging and Vision* **54**(3), 358–377 (2016) <https://doi.org/10.1007/s10851-015-0608-6>
- [22] Bartels, S.: Total variation minimization with finite elements: Convergence and iterative solution. *SIAM Journal on Numerical Analysis* **50**(3), 1162–1180 (2012) <https://doi.org/10.1137/11083277X>
- [23] Bredies, K., Lorenz, D.: Mathematical Image Processing, 1st edn. Applied and Numerical Harmonic Analysis. Birkhäuser, Cham (2018). <https://doi.org/10.1007/978-3-030-01458-2>
- [24] Shannon, C.E.: A mathematical theory of communication. *The Bell System Technical Journal* **27**(4), 623–656 (1948) <https://doi.org/10.1002/j.1538-7305.1948.tb00917.x>
- [25] Ern, A., Guermond, J.-L.: Finite element quasi-interpolation and best approximation. *ESAIM: Mathematical Modelling and Numerical Analysis* **51**(4), 1367–1385 (2017)
- [26] Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing* **13**(4), 600–612 (2004)
- [27] Ekeland, I., Témam, R.: Convex Analysis and Variational Problems, English edn. Classics in Applied Mathematics, vol. 28, p. 402. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA (1999). <https://doi.org/10.1137/1.9781611971088>. Translated from the French. <http://dx.doi.org/10.1137/1.9781611971088>
- [28] Bartels, S.: Numerical Methods for Nonlinear Partial Differential Equations vol. 14. Springer, Cham (2015)
- [29] Bezanson, J., Edelman, A., Karpinski, S., Shah, V.B.: Julia: A fresh approach to numerical computing. *SIAM review* **59**(1), 65–98 (2017)
- [30] Hilb, S.: <https://gitlab.mathematik.uni-stuttgart.de/stephan.hilb/SemiSmoothNewton.jl> <https://gitlab.mathematik.uni-stuttgart.de/stephan.hilb/SemiSmoothNewton.jl>

- [31] Baker, S., Scharstein, D., Lewis, J., Roth, S., Black, M.J., Szeliski, R.: A database and evaluation methodology for optical flow. *International Journal of Computer Vision* **92**(1), 1–31 (2011)
- [32] Dirks, H.: Variational methods for joint motion estimation and image reconstruction. PhD thesis, Universitäts-und Landesbibliothek Münster (2015)
- [33] Ainsworth, M., Oden, J.T.: *A Posteriori Error Estimation in Finite Element Analysis*. Wiley-Interscience, New York (2000)
- [34] Necas, J.: *Direct Methods in the Theory of Elliptic Equations*. Springer, Berlin (2011)
- [35] Aliprantis, C.D., Border, K.C.: *Infinite Dimensional Analysis*, 3rd edn. Springer, Berlin (2006). <https://doi.org/10.1007/3-540-29587-9>