Computer Vision, Graphics, and Pattern Recognition Group
Department of Mathematics and Computer Science
University of Mannheim
D-68131 Mannheim, Germany

Reihe Informatik
3/2005

**Domain Decomposition for Nonlinear Problems:
A Control-Theoretic Approach**

Timo Kohlberger, Christoph Schnörr, Andrés Bruhn,
Joachim Weickert

Technical Report 3/2005
Computer Science Series
April 2005

Domain Decomposition for Nonlinear Problems: A Control-Theoretic Approach

Timo Kohlberger[1], Christoph Schnörr[1], Andrés Bruhn[2], Joachim Weickert[2]

[1]Computer Vision, Graphics and Pattern Recognition Group
University of Mannheim
Dept. of Mathematics and Computer Science
D-68131 Mannheim, Germany
{tiko,schnoerr}@uni-mannheim.de

[2]Mathematical Image Analysis Group
Saarland University
Dept. of Mathematics and Computer Science
D-66041 Saarbrücken, Germany
{bruhn,weickert}@mia.uni-saarland.de

Abstract

We investigate the feasibility of a recent control-theoretic approach to domain decomposition for a class of nonlinear variational image processing problems. Like substructuring methods for solving in parallel linear (systems of) partial differential equations, the approach utilizes non-overlapping subdomains. Processor communication is therefore restricted to lower-dimensional interfaces. The approach is particularly suited for implementations on PC clusters, but also for on-chip parallelization on multi-core processors.

1 Introduction

Domain decomposition methods for the parallelized solving of nonlinear problems have been studied for the case of overlapping subdomains in, e.g., [11], [3], [5] and [24]. In these works, the well known Schwarz alternating methods for linear problems are applied to the nonlinear in a straightforward manner. In contrast, the direct application of non-overlapping domain decomposition methods, substructuring methods, has, to our knowledge, not been conducted successfully so far. However, recent approaches restating the decomposed problem by means of optimal control theory [22], [13], [16], [14], have shown to be feasible in application to the (nonlinear) Navier-Stokes equations. In order to study the practicability of those methods for nonlinear image processing problems, and thereby extending the studies made for the linear case at the example of motion estimation [19], [20], [21], the main focus of this article is the application to image denoising with Total Variation regularization, being a prominent representative for the class of nonlinear problems. In addition to the algorithmic details for the cases of two or more

subdomains, the mathematical and algorithmic structure of common solving techniques are explained.

The organization of the paper is as follows. After some preliminary definitions and the presentation of the model problem, the two-subdomains case is examined in Section 2. Besides the description of classical iterative solving techniques for optimal control problems, also numerical results for a significant example are reported. In Section 3, details for the application to multiple subdomains are elaborated, the followed by experimental results on four subdomains. The algorithmic and experimental results are then summarized in Section 4, and an outlook for future work is given.

1.1 Mathematic Preliminaries and Notations

Let $\Omega, \Omega_1, \Omega_2, \ldots \subset \mathbb{R}^2$ denote opened and bounded domains with "sufficiently smooth" (e.g., Lipschitz continuous) boundaries $\partial\Omega, \partial\Omega_1, \partial\Omega_2 \ldots$ and exterior unit normals $n, n^1, n^2, \ldots$. Furthermore, $\{\Omega_i | i = 1, \ldots, N\}$ is a non-overlapping partition of the image plane Ω, i.e. $\Omega = \bigcup_i \Omega_i$, $\Omega_i \cap \Omega_j = \emptyset$, $\forall j \neq i$—where Ω_i will be referred to as *subdomains* in the following—and the set of shared boundaries is defined by $\Gamma := \bigcup_i \partial\Omega^i \setminus \partial\Omega$. Additionally, we consider only those partitions for which a black-and-white coloring can be applied, i.e. adjoint subdomains always have differing colors there, see Figure 1 for two examples. Then, let I_B denote the indices of the black subdomains and I_W those of the white ones, i.e. $I_B := \{1 \leq i \leq N \mid \Omega_i \text{ is black}\}$, $I_W := \{1 \leq i \leq N \mid \Omega_i \text{ is white}\}$.

In cases with more than two subdomains, the common boundary set Γ is split further into parts Γ_{ij}, whose points are shared by boundaries of only two adjoint subdomains, i.e. $\Gamma_{ij} := \partial\Omega_i \cap \partial\Omega_j, \forall j \in N(i) \cap I_W, \forall i \in I_B$, where $N(i) := \{j | \partial\Omega_j \cap \partial\Omega_i \neq \emptyset, j \neq i\}$ denotes the indices of all subdomains being adjoint to Ω_i; and those points being the intersection of boundaries of more than two adjoint subdomains, which are collectively denoted as Γ_Π. Furthermore, all points on the boundary of a subdomain Ω_i which are also elements of adjoint subdomain's boundaries are denoted Γ_i. See Figure 1 for two examples.

In addition, we need the usual Sobolev space for second order elliptic boundary value problems

$$V = H^1(\Omega) = \{v \in L^2(\Omega) \ : \ \partial^\alpha v \in L^2(\Omega)\,,\ 0 \leq |\alpha| \leq 1\}\,,$$

with the scalar product of $L^2(\Omega)$ denoted by

$$(u, v) = \int_\Omega u(x)v(x)\ dx.$$

Moreover, we will make use of the notation ζ_i for the restriction of any function $\zeta \in V(\Omega)$ to Ω_i, i.e. $\zeta_i := \zeta_{|\Omega_i} \in V(\Omega_i)$. Analogically, we will abbreviate $V(\Omega_i)$ by V_i. Scalar products, being restricted to any subset $\Lambda \subset \Omega$ are denoted by

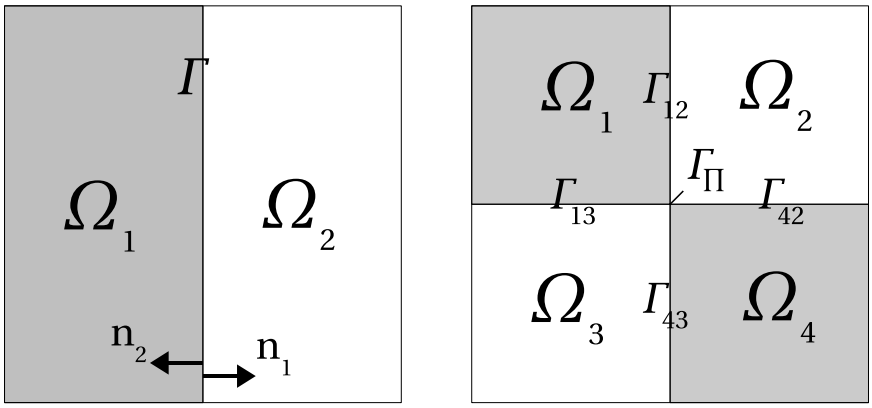
$$\big(u, v\big)_\Lambda := \int_\Lambda u(x)v(x)\ dx, \qquad \text{for } \Lambda \subset \Omega.$$

Finally, the following extension operators will be used: $P_i : \Gamma_i \to \Omega_i, P_{\Gamma_{ij}} : \Gamma_{ij} \to \Omega_i, P_{\Gamma\Pi} : \Gamma_\Pi \to \Gamma$, which all are the identity on their definition set and zero else.

Throughout this paper, all functions are discretized with standard conforming piecewise linear finite elements. To simplify notation, we use the *same symbols* for some function $v = v(x)$ and the coefficient vector $v \in \mathbb{R}^N$ representing the approximation of $v(x)$ in the subspace spanned by the piecewise linear basis functions $\{\phi(x)\}_{i=1,\dots,N}$:

$$v(x) \in H^1(\Omega) \quad \leftrightarrow \quad v \in \mathbb{R}^N \quad \leftrightarrow \quad \sum_{i=1}^{N} v_i \phi_i(x) \ .$$

Furthermore, we use the *same symbol* for the vector obtained by discretizing the action of some *linear* functional on some function $v(x)$. For example, we simply write f and w for the discretized versions of the linear functionals (f, v) and $a(w, v)$ (with $a(\cdot, \cdot)$ being a bilinear form and $w(x)$ fixed). We refer to standard textbooks like, e.g. [2], [9], [28], for the discretization of boundary value problems with finite elements.



(a) 1×2-partition (b) 2×2-partition

Figure 1: Exemplary partitions of the image plane Ω. (a) $\Gamma = \partial\Omega_1 \cap \partial\Omega_2$. (b) $\Gamma_\Pi = \bigcap_i \partial\Omega_i, \Gamma_{1i} = \partial\Omega_1 \cap \partial\Omega_i \setminus \Gamma_\Pi, i = 2, 3; \Gamma_{4i} = \partial\Omega_4 \cap \partial\Omega_i \setminus \Gamma_\Pi, i = 2, 3$.

1.2 The Model Problem

As a representative for the class of nonlinear elliptic boundary-value problems we consider the Total Variation-based denoising problem in the sequel [27], [26], which is defined by the optimization problem

$$\min_{u \in V} J(u) := \frac{1}{2} \int_\Omega (u - f)^2 \, dx \ + \ \alpha J_{TV}(u). \tag{1}$$

Here, $f \in V$ represents the degradated input image, α the regularization strength parameter and J_{TV} is the *Total Variation norm*

$$J_{TV}(u) := \sup_{w \in (C_0^\infty)^2} \left\{ \int_\Omega -u \nabla \cdot w \, dx \ : \|w\|_\infty \leq 1 \right\} \tag{2}$$

with $u \in V \subset BV$, see e.g. [31], denoting the image to be reconstructed, which plays the role of an edge-preserving regularization here. Finding a solution to this problem amounts to solving the nonlinear equation

$$\int_\Omega u\, v \; + \alpha \frac{\nabla u^\top \nabla v}{|\nabla u|_\beta}\, dx := a(u,v) = \int_\Omega f v \, dx := b(v), \qquad \forall v \in V(\Omega), \tag{3}$$

where the approximation

$$|z|_\beta := \sqrt{z \cdot z + \beta}$$

of the Euclidean norm is used[1], for a small perturbation parameter β, in order to prevent problems if $|\nabla u| \to 0$. Partial integration in (3) shows that u leads to the differential formulation:

$$u - \alpha \nabla \cdot \left(\frac{\nabla u}{|\nabla u|_\beta} \right) := A(u) = f \tag{4}$$

with the homogeneous Neumann boundary conditions

$$\frac{\partial u}{\partial n} = 0 \text{ on } \partial\Omega. \tag{5}$$

Due to nonlinearity of the TV-regularized problem, non-standard, iterative solving methods have emerged in the past 15 years. Starting with the slowly-converging steepest descent method proposed by Rudin and Osher [27], Vogel et al. [29], [30], [10], later on suggested a fixed point iteration with an improved convergence behavior, of which Heers et al. [17] noted to belong to the family of so-called *Kačanov methods* [18], [12], being already proposed at the end 1960s. Although quadratic convergence was then reached by employing Newton's method, [8], its domain of convergence has shown to be very small. In order to overcome this hindrance, Chan, Golub and Mulet proposed the primal-dual Newton's method [6], [7], having a large domain of convergence and a quadratic convergence rate. The latter method was used to solve Eq. (3), or its restrictions to subdomains, in the experiments whose results are presented below. As error measure the nonlinear relative residual error was chosen.

2 The Two-Subdomains Case

2.1 Problem Statement

Let us start with the spatial decomposition of the nonlinear problem (4) onto to two subdomains. Let us consider the two subproblems

$$\begin{aligned} A(u_1) &= f_1 \quad \text{and} \quad \frac{\partial u_1}{\partial n_1} = g \text{ on } \Gamma, \quad \frac{\partial u_1}{\partial n} = 0 \text{ on } \partial\Omega_1 \setminus \Gamma \\ A(u_2) &= f_2 \quad \text{and} \quad \frac{\partial u_2}{\partial n_2} = -g \text{ on } \Gamma, \quad \frac{\partial u_2}{\partial n} = 0 \text{ on } \partial\Omega_2 \setminus \Gamma \end{aligned} \tag{6}$$

[1] In fact, by utilizing this approximation, one can give an non-weak approximation of the TV norm by $J_{TV}(u) \approx \int_\Omega |u|_\beta \, dx$, for $u \in H^1$.

where g is a given function in $L^2(\Gamma)$, and

$$u = \begin{cases} u_1(x) & x \in \Omega_1 \\ u_2(x) & x \in \Omega_2 \end{cases}. \tag{7}$$

That is, by Eq. (6) a splitting of the original problem $A(u) = f$ into two subproblems is given, whose Neumann boundary conditions have been modified in such, that the normal derivatives of the local solutions u_1 and u_2 are equal to g and $-g$, respectively. Since g is given, we can thus write $u_1(g)$ and $u_2(g)$.

Furthermore, Eqs. (6) shall now serve as the constraints of the constrained optimization problem

$$\min_{u_1,u_2,g} \frac{1}{2} \int_\Gamma (u_1 - u_2)^2 \, d\chi + \frac{\gamma}{2} \int_\Gamma g^2 \, d\chi := J_\Gamma\big(u_1, u_2, g\big) \tag{8}$$
$$\text{subject to (6)},$$

cf. [22], whereas for the time being we assume $\gamma = 0$, i.e. neglect the second integral. Then, for a global minimum $(\hat{u}_1, \hat{u}_2, \hat{g})$ of J_Γ it obviously holds that $\hat{u}_1(g) = \hat{u}_2(g)$ on Γ, due to the definition of J_Γ; as well as $\partial_n \hat{u}_1 = g = -\partial_n u_2$, due to the construction of the constraint equations. That is, a solution to problem (8) does, in negligence of the second integral, also satisfy

$$\begin{cases} A_1(u_1) = f_1 \quad \text{and} \quad \dfrac{\partial u_1}{\partial n_1} = 0 \text{ on } \partial\Omega_1 \setminus \Gamma \\ \dfrac{\partial u_1}{\partial n_1} = g = -\dfrac{\partial u_2}{\partial n_2} \quad \text{on} \quad \Gamma \\ u_1 = u_2 \quad \text{on} \quad \Gamma \\ A_2(u_2) = f_2 \quad \text{and} \quad \dfrac{\partial u_2}{\partial n_2} = 0 \text{ on } \partial\Omega_2 \setminus \Gamma \end{cases} \tag{9}$$

which is the so-called *multi-domain formulation* of the original problem $A(u) = f$. Interestingly, if A would be a linear operator, the optimization problem (8) could therefore be associated with the well known class of non-overlapping domain decomposition methods, *substructuring methods*, since the multi-domain formulation serves as basis of the Steklov-Poincaré interface equation there, cf. [25]. Although, since one assumption for the latter is linearity of the operator, it is not applicable here and hence classical substructuring methods are not feasible in this case. On the other hand, by following the constrained optimization-based approach Eq. (8), we are also able to exploit parallelization for nonlinear problems.

Concerning the second integral in Eq. (8), its reason is to prevent from getting arbitrarily large solutions for g, since its magnitude is not involved in the first integral. However, experiments with the chosen model problem on two and four subdomains showed convergence for $\gamma = 0$, see details below.

Independent from that, the problem in Eq. (8) belongs to the well known class of *optimal control problems*, cf., e.g., [23], where g here is a *boundary control*; J_Γ is the *objective functional*; Eqs. (6) are the *state equations* and $\big(u_1(g), u_2(g)\big)$ are denoted as the *state*.

2.2 Lagrange Relaxation and the Optimality System

A direct approach to the optimal control problem (8) could be to solve the optimality system which belongs to the Lagrange relaxation of the constrained optimization problem, see, e.g., [14].

In order to simplify the Lagrange multiplier function to be defined below, we will make use of the weak formulation of the constraint equation system (6):

$$\begin{aligned} a_1(u_1, v_1) &= b_1(v_1) + \big(u_1, v_1\big)_\Gamma, \qquad \forall v_1 \in V_1, u_1 \in V_1 \\ a_2(u_2, v_2) &= b_2(v_2) - \big(u_2, v_2\big)_\Gamma, \qquad \forall v_2 \in V_2, u_2 \in V_2, \end{aligned} \tag{10}$$

cf., e.g., [1], which can be written in more compact form

$$\big(F_1(u_1, g), v_1\big)_{\Omega_1} = 0, \qquad \forall v_1 \in V_1,\ u_1 \in V_1 \tag{11}$$

$$\big(F_2(u_2, g), v_2\big)_{\Omega_2} = 0, \qquad \forall v_2 \in V_2,\ u_2 \in V_2 \tag{12}$$

by the introduction of the nonlinear operators $F_i(\cdot, \cdot)$,

$$\big(F_i(u_i, g), v_i\big)_{\Omega_i} := a_i(u_i, v_i) - b_i(v_i) + (-1)^{i-1}(g, v_i)_\Gamma, \quad \forall v_i \in V_i,\ i = 1, 2. \tag{13}$$

Now, we are ready to give a compact definition of the Lagrange functional by

$$L(u_1, u_2, g, \lambda_1, \lambda_2) := J_\Gamma(u_1, u_2, g) - \big(F_1(u_1, g), \lambda_1\big)_{\Omega_1} - \big(F_2(u_2, g), \lambda_2\big)_{\Omega_2}, \tag{14}$$

where $\lambda_1 \in V_1, \lambda_2 \in V_2$ are the Lagrange multiplier functions.

First-order necessary conditions for finding a solution $(\hat{u}_1, \hat{u}_2, \hat{g})$ to the original problem (8) is to find a stationary point $(\hat{u}_1, \hat{u}_2, \hat{g}, \hat{\lambda}_1, \hat{\lambda}_2)$ of the Lagrange functional L. That is, one has to solve the system

$$\nabla^5 L(\hat{u}_1, \hat{u}_2, \hat{g}, \hat{\lambda}_1, \hat{\lambda}_2) = 0,$$

where $\nabla^5 := (\partial/\partial u_1, \partial/\partial u_2, \partial/g, \partial/\partial\lambda_1, \partial/\partial\lambda_2)$, which gives the *optimality system* to the Lagrange relaxation and is derived in detail in the following.

Partially deriving $L(u_1, u_2, g, \lambda_1, \lambda_2)$ for the Lagrange functions λ_1 and λ_2, respectively, yield the so-called *state equations*

$$\left\langle \frac{\partial L}{\partial \lambda_i}, v_i \right\rangle_{\Omega_i} = 0, \qquad \forall v_i \in V_i,\ i = 1, 2 \tag{15}$$

$$\Leftrightarrow \qquad \big(F_i(u_i, g), v_i\big)_{\Omega_i} = 0, \qquad \forall v_i \tag{16}$$

$$\Leftrightarrow \qquad a_i(u_i, v_i) = b_i(v_i) + (-1)^{i-1}(g, v_i)_\Gamma, \qquad \forall v_i \tag{17}$$

which are just the constraint equations of the original problem. On the other hand, deriving

with respect to u_1 and u_2, respectively, results in the *adjoint* or *co-state equations*

$$\left\langle \frac{\partial L}{\partial u_i}, v_i \right\rangle_{\Omega_i} = 0, \qquad \forall v_i \in V_i,\ i = 1,2 \tag{18}$$

$$\Leftrightarrow \qquad \left\langle \frac{\partial J_\Gamma}{\partial u_i}, v_i \right\rangle_{\Omega_i} - \left(\left\langle \frac{\partial F_i}{\partial u_i}, v_i \right\rangle_{\Omega_i}, \lambda_i \right)_{\Omega_i} = 0, \qquad \forall v_i \tag{19}$$

$$\Leftrightarrow \qquad a_i'(u_i, v_i, \lambda_i) = \left(u_{1|\Gamma} - u_{2|\Gamma}, (-1)^{i-1} v_{i|\Gamma}\right)_\Gamma, \qquad \forall v_i \tag{20}$$

where $a_i'(u_i, v_i, \lambda_i)$ is defined as follows:

$$a_i'(u_i, v_i, \lambda_i) := \left(\left\langle \frac{\partial F_i(u_i, g)}{\partial u_i}, v_i \right\rangle_{\Omega_i}, \lambda_i \right)_{\Omega_i} \tag{21}$$

$$= \int_{\Omega_i} v_i\ \lambda_i + \alpha \left(\frac{1}{(\nabla u_i^\top \nabla u_i + \beta)^{1/2}} \nabla v_i^\top - \frac{1}{(\nabla u_i^\top \nabla u_i + \beta)^{3/2}} \nabla u_i^\top \nabla v_i \nabla u_i^\top \right) \nabla \lambda_i \, dx \tag{22}$$

$$= \int_{\Omega_i} v_i \lambda_i + \frac{\alpha}{|\nabla u_i|_\beta} \nabla v_i^\top \left(I - \frac{\nabla u_i \nabla u_i^\top}{|\nabla u_i|_\beta^2} \right) \nabla \lambda_i \, dx. \tag{23}$$

Additionally, solutions to u_1 and u_2 to Eq. (20) are denoted as *co-states*. Finally, by partially deriving for the control g we obtain the *optimality condition*

$$\left\langle \frac{\partial L}{\partial g}, \tilde{g} \right\rangle_\Gamma = 0, \qquad \forall \tilde{g} \in L^2(\Gamma),\ i = 1,2 \tag{24}$$

$$\Leftrightarrow \qquad \left\langle \frac{\partial J_\Gamma}{\partial g}, \tilde{g} \right\rangle_\Gamma - \left(\left\langle \frac{\partial F_1}{\partial g}, \tilde{g} \right\rangle_\Gamma, \lambda_1 \right)_\Gamma - \left(\left\langle \frac{\partial F_2}{\partial g}, \tilde{g} \right\rangle_\Gamma, \lambda_2 \right)_\Gamma = 0, \quad \forall \tilde{g} \tag{25}$$

$$\Leftrightarrow \qquad \gamma \left(g, \tilde{g}\right)_\Gamma + \left(\lambda_{1|\Gamma} - \lambda_{2|\Gamma}, \tilde{g}\right)_\Gamma = 0, \quad \forall \tilde{g}. \tag{26}$$

To summarize, in order to find a solution for the Lagrange relaxation with the model problem one has to solve the following coupled system of five equations

$$\begin{cases} \displaystyle\int_{\Omega_i} u_i\ v_i\ + \alpha \frac{\nabla u_i^\top \nabla v_i}{|\nabla u_i|_\beta}\, dx = \int_{\Omega_i} f_i v_i \, dx + (-1)^{i-1} \left(g, v_i\right)_\Gamma, & \forall v_i \in V_i, \\ \displaystyle\int_{\Omega_i} v_i \lambda_i + \frac{\alpha}{|\nabla u_i|_\beta} \nabla v_i^\top \left(I - \frac{\nabla u_i \nabla u_i^\top}{|\nabla u_i|_\beta^2} \right) \nabla \lambda_i \, dx = \left(u_{1|\Gamma} - u_{2|\Gamma}, v_{i\Gamma}\right)_\Gamma, & \forall v_i \in V_i, \\ \gamma \left(g, \tilde{g}\right)_\Gamma = -\left(\lambda_{1|\Gamma} - \lambda_{2|\Gamma}, \tilde{g}\right)_\Gamma, & \forall \tilde{g} \in L^2(\Gamma) \end{cases} . \tag{27}$$

for $i = 1, 2$.

Although it would be feasible to discretize and solve this system directly, which is denoted *one-shot method* in literature, cf. [14], it is generally an ineffective solving approach,

since it yields large nonlinear equations systems, which are usually solved by slow, iterative methods, as it is the case with TV-based denoising considered here. Additionally, such a solving approach provides no direct clues for parallel computation, since u_1 and λ_1 as well as u_2 and λ_2 are coupled on the whole of Ω_1 and Ω_2, respectively, and these two groups among each other via their dependencies on g at the common boundary Γ.

In contrast to solving solving the optimality system directly, an iterative procedure provides an alternative, where, starting with some initial guess for the control g, first the state equations are solved for u_1 and u_2; second in using the results of the first step, the adjoint equations are solved for the co-state functions λ_1 and λ_2; and third the control g is updated by solving the optimality condition. These three steps are repeated until convergence is reached due to some appropriate error threshold. Although this procedure provides the advantages of parallelizability of the first and second step, as well as having to solve only small nonlinear systems for u_1 and u_2 within the first step, convergence is not guaranteed, or may happen with a bad convergence rate, since it has been shown [15], that this procedure is equivalent to a steepest descent algorithm for g using a fixed step size.

2.3 Gradient-Based Solving

In contrast to methods solving the optimality system directly, iterative gradient-based methods are usually the methods of choice in practice. Starting with an initial guess for the control g, the repetitive part of these algorithms is outlined by the following four steps:

(i) solve the state equations to obtain the current state $\big(u_1(g^k), u_2(g^k)\big)$

(ii) compute the gradient of J_Γ w.r.t. to g: $\frac{d}{dg} J_\Gamma\big(u_1(g^k), u_2(g^k), g\big)$

(iii) use the results of the first two steps to compute an update δg

(iv) update g: $g \leftarrow g + \delta g$,

where k is the iteration count, see also [14]. Whereas step (i) just corresponds to solving the local problems in the constraint equations, step (ii) needs further considerations, which are detailed in the following. Once the gradient has been computed, an update of the control is determined in step (iii), the details of which depend on the optimization method used, e.g. the gradient method, the conjugate gradient or quasi-Newton method.

2.3.1 Gradient Calculation by Sensitivities versus by Co-States

Before presenting the computation of the gradient in step (ii) for our model example explicitly, the general mathematical structure for the two-subdomains case is outlined next.

We start with the formal structure of the total derivate of J_Γ with respect to g at (u_1, u_2, g) in an arbitrary direction $\tilde{g} \in L^2(\Gamma)$:

$$\left\langle \frac{dJ_\Gamma}{dg}, \tilde{g} \right\rangle_\Gamma = \left\langle \frac{\partial J_\Gamma}{\partial u_1}, \left\langle \frac{\partial u_1}{\partial g}, \tilde{g} \right\rangle_\Gamma \right\rangle_\Gamma + \left\langle \frac{\partial J_\Gamma}{\partial u_2}, \left\langle \frac{\partial u_2}{\partial g}, \tilde{g} \right\rangle_\Gamma \right\rangle_\Gamma + \left\langle \frac{\partial J_\Gamma}{\partial g}, \tilde{g} \right\rangle_\Gamma . \tag{28}$$

Defining $\tilde{u}_i(g) := \left\langle \frac{\partial u_i}{\partial g}, \tilde{g} \right\rangle_\Gamma$, $i = 1, 2$, this can be rewritten as

$$\left\langle \frac{dJ_\Gamma}{dg}, \tilde{g} \right\rangle_\Gamma = \left\langle \frac{\partial J_\Gamma}{\partial u_1}, \tilde{u}_1 \right\rangle_\Gamma + \left\langle \frac{\partial J_\Gamma}{\partial u_2}, \tilde{u}_2 \right\rangle_\Gamma + \left\langle \frac{\partial J_\Gamma}{\partial g}, \tilde{g} \right\rangle_\Gamma . \tag{29}$$

Here, $\tilde{u}_1(g)$ and $\tilde{u}_2(g)$ are the directions of infinitesimal change of the state functions u_1 and u_2, respectively, in dependence of the direction of infinitesimal change $\tilde{g}$. Or, in other words, $\tilde{u}_1$ and $\tilde{u}_2$ represent the variation directions in the state in dependence of a variation direction in the control g. Therefore, $\tilde{u}_1(g)$ and $\tilde{u}_2(g)$ are commonly referred to as *sensitivity* in literature [14].

In general, two ways of calculating $\tilde{u}_1(g)$ and $\tilde{u}_2(g)$ exist. The first one follows from considering the total derivative of the state equations with respect to g, which is given by

$$\left(P_i \left\langle \frac{d}{dg} F_i(u_i, g), \tilde{g} \right\rangle_\Gamma , v_i \right)_{\Omega_i} = 0, \quad \forall v_i \in V_i,\ i = 1, 2 \tag{30}$$

where, due to u_1 and u_2 depending on the control g, the chain rule applies again, yielding

$$\Leftrightarrow \qquad \left(\left\langle \frac{\partial F_i}{\partial u_i}, P_i \left\langle \frac{\partial u_i}{\partial g}, \tilde{g} \right\rangle_\Gamma \right\rangle_{\Omega_i} + P_i \left\langle \frac{\partial F_i}{\partial g}, \tilde{g} \right\rangle_\Gamma , v_i \right)_{\Omega_i} = 0, \quad \forall v_i. \tag{31}$$

By substitution of $\left\langle \frac{\partial u_i}{\partial g}, \tilde{g} \right\rangle_\Gamma$, $i = 1, 2$ through $\tilde{u}_1$ and $\tilde{u}_2$, respectively, we obtain

$$\Leftrightarrow \qquad \left(\left\langle \frac{\partial F_i}{\partial u_i}, \tilde{u}_i \right\rangle_{\Omega_i} + P_i \left\langle \frac{\partial F_i}{\partial g}, \tilde{g} \right\rangle_\Gamma , v_i \right)_{\Omega_i} = 0, \quad \forall v_i \tag{32}$$

$$\Leftrightarrow \qquad \left(\left\langle \frac{\partial F_i}{\partial u_i}, \tilde{u}_i \right\rangle_{\Omega_i} , v_i \right)_{\Omega_i} = - \left(P_i \left\langle \frac{\partial F_i}{\partial g}, \tilde{g} \right\rangle_\Gamma , v_i \right)_{\Omega_i} , \quad \forall v_i . \tag{33}$$

Hence, with equation (33), denoted as *sensitivity equation*, we have given the dependency between $\tilde{u}_1$, $\tilde{u}_2$ and $\tilde{g}$. Unfortunately, it is clear that, since equation (33) is not formally inverted, $\tilde{u}_1$, $\tilde{u}_2$ can be determined only for particular $\tilde{g}$, through solving Eq. (33). In recapitulation of the gradient formulation in Eq. (29) it becomes obvious, that by Eq. (33) we are not able to compute $\langle dJ/dg, \tilde{g} \rangle$ for arbitrary directions $\tilde{g}$, and therefore only projections of the gradient onto fixed directions $\tilde{g}$.

However, in most continuous cases $\tilde{g}$ need to be arbitrary, therefore we are interested in a formulation without the incorporation of the sensitivities $\tilde{u}_1$ and $\tilde{u}_2$, which is referred to as 'calculation of the gradient through adjoint equations' in literature [14]. Such a formulation can be reached by considering the adjoint equations (20),

$$\left(\left\langle \frac{\partial F_i}{\partial u_i}, v_i \right\rangle_{\Omega_i} , \lambda_i \right)_{\Omega_i} = \left\langle \frac{\partial J_\Gamma}{\partial u_i}, v_i \right\rangle_\Gamma , \qquad \forall v_i \in V_i,\ i = 1, 2, \tag{34}$$

of the previous section again. Since this holds for any v_i, it is in particular true for setting $v_i = \tilde{u}_i$. Analogical, Eq. (33) is true for an arbitrary v_i and thus for an particular $v_i = \lambda_i$.

In applying these substitutions to Eqs. (34) and Eqs. (33), respectively, and comparing the outcomes, one can deduce that

$$\left\langle \frac{\partial J_\Gamma}{\partial u_i}, \tilde{u}_i \right\rangle_\Gamma = -\left(P_i \left\langle \frac{\partial F_i}{\partial g}, \tilde{g} \right\rangle_\Gamma, \lambda_i \right)_{\Omega_i}. \tag{35}$$

Thus, terms in the gradient formulation (29) involving the sensitivities $\tilde{u}_1$ and $\tilde{u}_2$ can now be substituted due to (35), which yields the gradient formulation

$$\left\langle \frac{dJ_\Gamma}{dg}, \tilde{g} \right\rangle_\Gamma = -\left(P_1 \left\langle \frac{\partial F_1}{\partial g}, \tilde{g} \right\rangle_\Gamma, \lambda_1 \right)_{\Omega_1} - \left(P_2 \left\langle \frac{\partial F_2}{\partial g}, \tilde{g} \right\rangle_\Gamma, \lambda_2 \right)_{\Omega_2} + \left\langle \frac{\partial J_\Gamma}{\partial g}, \tilde{g} \right\rangle_\Gamma \tag{36}$$

$$\Leftrightarrow \quad \left\langle \frac{dJ_\Gamma}{dg}, \tilde{g} \right\rangle_\Gamma = \left\langle \frac{\partial J_\Gamma}{\partial g}, \tilde{g} \right\rangle_\Gamma - \left(P_1 \left\langle \frac{\partial F_1}{\partial g}, \tilde{g} \right\rangle_\Gamma, \lambda_1 \right)_{\Omega_1} - \left(P_2 \left\langle \frac{\partial F_2}{\partial g}, \tilde{g} \right\rangle_\Gamma, \lambda_2 \right)_{\Omega_2}. \tag{37}$$

Obviously, we have now given a closed-form of the gradient for arbitrary $\tilde{g}$ and not for particular ones only, since there is no need to solve the sensitivity equations in an intermediate step any longer. Instead, one has to solve the adjoint equations (34) for λ_1 and λ_2, respectively. Since the latter only involves the state functions u_1, u_2 and the control g, but not the change direction $\tilde{g}$, this solving has to be done only *once* for any $\tilde{g}$.

2.3.2 Application to the Model Problem

After the general description of the gradient-based approach, the concrete application to the model problem is explained in the following.

Obviously, in considering the definition of J_Γ as given in Eq. (8), the gradient formulation in Eq. (29) for the model problems here reads

$$\left\langle \frac{dJ_\Gamma}{dg}, \tilde{g} \right\rangle_\Gamma = \left(u_{1|\Gamma} - u_{2|\Gamma}, \tilde{u}_{2|\Gamma} - \tilde{u}_{2|\Gamma} \right)_\Gamma + \gamma \left(g, \tilde{g} \right)_\Gamma. \tag{38}$$

Furthermore, the sensitivity equations corresponding to (33) is of the form

$$\begin{aligned} a_1'(u_1, \tilde{u}_1, \xi_1) &= (\tilde{g}, \xi_1)_\Gamma, \quad \forall \xi_1 \in V_1 \\ a_2'(u_2, \tilde{u}_2, \xi_2) &= -(\tilde{g}, \xi_2)_\Gamma, \quad \forall \xi_2 \in V_2, \end{aligned} \tag{39}$$

with $a_i'(\cdot, \cdot, \cdot)$ as defined in (21) in connection with the adjoint equations.

Alternatively, in following the adjoint equations approach, the gradient formulation in Eq. (37) here reads

$$\left\langle \frac{dJ_\Gamma}{dg}, \tilde{g} \right\rangle_\Gamma = \left(\tilde{g}, \lambda_1 \right)_\Gamma - \left(\tilde{g}, \lambda_2 \right)_\Gamma + \gamma (g, \tilde{g})_\Gamma \tag{40}$$

$$= \left(\tilde{g}, \lambda_{1|\Gamma} - \lambda_{2|\Gamma} \right)_\Gamma + \gamma \left(g, \tilde{g} \right)_\Gamma \tag{41}$$

or, in explicit formulation

$$\frac{dJ_\Gamma}{dg} = \left(\lambda_{1|\Gamma} - \lambda_{2|\Gamma} \right) + \gamma g, \tag{42}$$

where the co-states λ_1, λ_2 are to be determined by solving the adjoint equations

$$\begin{aligned} a_1'(u_1, v_1, \lambda_1) &= \left(u_{1|\Gamma} - u_{2|\Gamma}, v_{1|\Gamma}\right)_\Gamma, \qquad \forall v_1 \in V_1 \\ a_2'(u_2, v_2, \lambda_2) &= \left(u_{1|\Gamma} - u_{2|\Gamma}, -v_{2|\Gamma}\right)_\Gamma, \qquad \forall v_2 \in V_2 \end{aligned} \tag{43}$$

which have been already introduced with the Lagrange multiplier approach, Eq. (20).

The relation between the sensitivity and the co-state functions, as shown generally in Eq. (35), can be reproduced here by setting $\xi_1 = \lambda_1$, $\xi_2 = \lambda_2$ in (39), and $v_1 = \tilde{u}_1$, $v_2 = \tilde{u}_2$ in (43), resulting in

$$\begin{aligned} (\tilde{g}, \lambda_1)_\Gamma &= \left(u_{1|\Gamma} - u_{2|\Gamma}, v_{1|\Gamma}\right)_\Gamma \\ (\tilde{g}, \lambda_2)_\Gamma &= \left(u_{1|\Gamma} - u_{2|\Gamma}, -v_{2|\Gamma}\right)_\Gamma \end{aligned} \tag{44}$$

and thus

$$\left(\tilde{g}, \lambda_{1|\Gamma} - \lambda_{2|\Gamma}\right)_\Gamma = \left(u_{1|\Gamma} - u_{2|\Gamma}, \tilde{u}_{2|\Gamma} - \tilde{u}_{2|\Gamma}\right)_\Gamma, \tag{45}$$

which, utilized to substitute the first term at the right-hand side of the first gradient formulation (38), just leads to the formulation in (40).

To summarize, Step (ii) of the algorithm outline in the beginning of this section, consists of the following sub-steps:

(a) solve the state equations (10)

(b) solve the co-state equations (43)

(c) calculate the gradient by Eq. (40) or Eq. (42) .

2.3.3 Outer CG Iteration

After having explained Step (ii) in detail, a complete gradient-based solving algorithm for two subdomains can be constructed. In utilizing conjugate gradient iteration for nonlinear problems, see, e.g., [4], the algorithm then reads as given in Algorithm 1. There, (a) corresponds to step (i) in the algorithm outline above; (b) and (c) to step (ii), (d) to step (iii), and (f) to step (iv). Moreover, the line search along the update direction δg in (e), implemented by ten nested iterations starting with an interval of $[0, 5]$, was employed to guarantee global convergence.

With respect to coarse-grained parallelization, the solvings in step (a) and (b) are obviously for parallelization, since both system are not coupled. In contrast, the steps (c)–(d) and (f)–(g) provide no clues and must be carried out sequentially, but comprise much less operations than steps (a) and (b). Since the many-subdomains case will be explained later on, details on communication patterns and considerations about scalability will be given there.

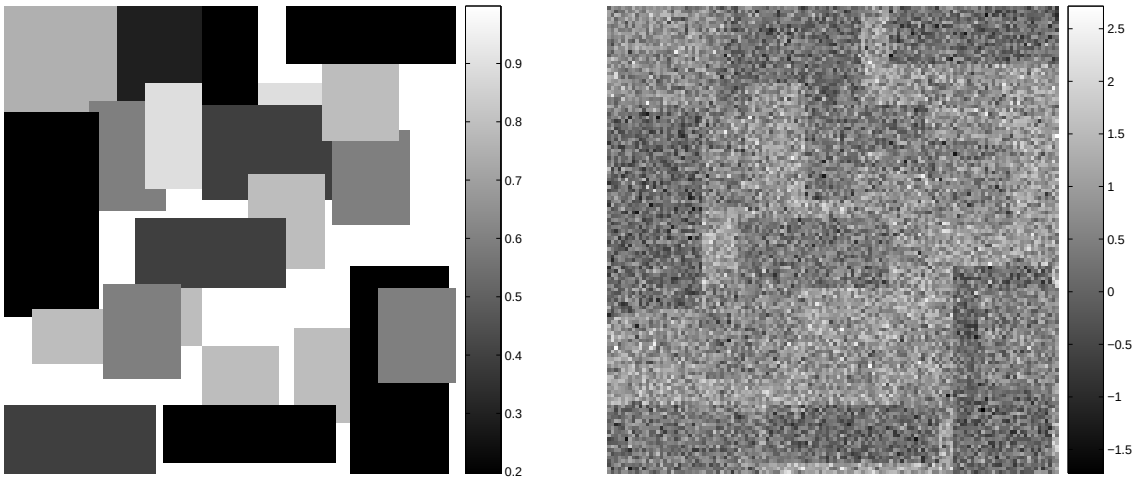
In terms of discretization, state and adjoint equations were discretized by conformal first-order finite elements, whereas the steps (c)–(f) were applied to the nodal weights directly. Since the adjoint equations are linear in λ_1 and λ_2, respectively, their corresponding linear systems were solved by LU decomposition; whereas the primal-dual Newton method, also in connection with LU decomposition, was utilized to solve the state equations up to a final (nonlinear) relative

residual error of less then 10^{-10} for all experiments. The regularization strength α was set to the relatively high value 1.0, in order to show the regularization preservation across the subdomain boundaries. Furthermore, we set $\beta = 10^{-6}$ in all experiments. Figure 2(b) served as input image, which was generated by adding Gaussian noise to the synthetic image depicted in Fig. 2(a). All error measurements refer to the solution of the original problem (4) computed without domain decomposition the same parameter values.

2.3.4 Experimental Results

The goal of the experiments, whose results are shown in Figure 3, were to study the general feasibility of the proposed domain decomposition approach, i.e. convergence, convergence rate as well as the distribution of the error in comparison of a sequential reference solution with the same parameters and accuracy (in terms of error thresholds of the local solvers).

The results in diagram 3(a) reveal that the relative L^2 error linearly drops to 10^{-4} with a relatively good rate of ≈ 0.89 until iteration 50, but then deteriorates to ≈ 0.99. Furthermore, setting $\gamma = 0$, i.e. switching off the regularization of g, does not lead to divergence here. In contrast, experiments show that the convergence behavior does not change significantly for setting γ to values smaller than 10^{-4} or equal to zero. Despite the worsening convergence rate, the density plots of the resulting image in Fig. 3(b) show that the remaining error after 50 iterations is acceptable for most image processing applications. To sum up, the experiments have shown, that the domain decomposition approach is practically feasible, but convergence rate can be improved further. Most promising in this respect is the use of preconditioned conjugate gradient iteration, which will be the subject of future work.



(a) (b) $\beta = 10^{-6}$

Figure 2: Ground truth and noised input image. 128 × 128 pixels, signal-to-noise ratio: -4.2 dB.

$g^0 \leftarrow 0$

$k \leftarrow 0$

do

(a) solve the state equations for u_1^k, u_2^k (in parallel):

$$\begin{aligned} a_1(u_1^k, v_1) &= b_1(v_1) + (g^k, v_1)_\Gamma, \qquad \forall v_1 \in V(\Omega_1) \\ a_2(u_2^k, v_2) &= b_2(v_2) - (g^k, v_2)_\Gamma, \qquad \forall v_2 \in V(\Omega_2) \end{aligned} \tag{46}$$

(b) solve the adjoint equations for λ_1^k, λ_2^k (in parallel):

$$\begin{aligned} a_1'(u_1^k, v_1, \lambda_1^k) &= (u_1^k - u_2^k, v_1)_\Gamma, \qquad \forall v_1 \in V(\Omega_1) \\ a_2'(u_2^k, v_2, \lambda_2^k) &= (u_1^k - u_2^k, v_2)_\Gamma, \qquad \forall v_2 \in V(\Omega_2) \end{aligned} \tag{47}$$

(c) calculate the gradient:

$$\nabla J_\Gamma^k \leftarrow (\lambda_{1|\Gamma}^k - \lambda_{2|\Gamma}^k) + \gamma g^k \tag{48}$$

(d) calculate a new update direction δd^k:

$$\beta^k \leftarrow \frac{\nabla J_\Gamma^k \cdot \left(\nabla J_\Gamma^k - \nabla J_\Gamma^{k-1}\right)}{\nabla J_\Gamma^{k-1} \cdot \nabla J_\Gamma^{k-1}}, \qquad \text{where} \quad \nabla J_\Gamma^k := J_\Gamma^k(u_1^k, u_2^k, g^k) \tag{49}$$

$$\delta g^k \leftarrow -\nabla J_\Gamma^k + \beta^k \delta g^{k-1} \tag{50}$$

(e) do a line search along δd^k for determining a step size τ^k:

$$\tau^k \leftarrow \min_{0<\tau\leq\tau_{\max}} J_\Gamma\big(u_1(g^k + \tau\delta g^k), u_2(g^k + \tau\delta g^k), g^k + \tau\delta g^k\big) \tag{51}$$

(f) update the control:

$$g^{k+1} \leftarrow g^k + \tau^k \delta g^k \tag{52}$$

(g) $k \leftarrow k + 1$

while $\|\nabla J_\Gamma^{k-1}\| / \|\nabla J_\Gamma^0\| > \epsilon$

merge the local solutions:

$$\begin{aligned} u_{|\Omega_1} &\leftarrow u_1^{k-1} \\ u_{|\Omega_2} &\leftarrow u_2^{k-1} \end{aligned} \tag{53}$$

Algorithm 1: Two-subdomains control-theoretic domain decomposition with CG iteration.

3 The Multiple-Subdomains Case

We may extend our approach to the case of multiple subdomains. The main difference is the presence of points Γ_Π which are shared by more than two subdomain boundaries, and hence unknowns at those places occur in more than two subproblems. See Figure (1)(b) for an illustration of a four-subdomains example. Once the special characteristics of unknowns at Γ_Π for a 2×2 decomposition is understood, it can be directly extended to cases of more subdomains. Therefore, we will restrict our further considerations, to the 2×2 case here.

3.1 Problem Statement

Induced by the four-subdomains partition of Ω, as defined in Section 1.1, we decompose the original problem $A(u) = f$ on Ω into the subproblems

$$\begin{aligned}
A(u_1) = f_1 \quad &\text{and} \quad \frac{\partial u_1}{\partial n_1} = g \text{ on } \Gamma, \quad \frac{\partial u_1}{\partial n} = 0 \text{ on } \partial\Omega_1 \setminus \Gamma \\
A(u_2) = f_2 \quad &\text{and} \quad \frac{\partial u_2}{\partial n_2} = -g \text{ on } \Gamma, \quad \frac{\partial u_2}{\partial n} = 0 \text{ on } \partial\Omega_2 \setminus \Gamma \\
A(u_3) = f_3 \quad &\text{and} \quad \frac{\partial u_3}{\partial n_3} = -g \text{ on } \Gamma, \quad \frac{\partial u_3}{\partial n} = 0 \text{ on } \partial\Omega_3 \setminus \Gamma \\
A(u_4) = f_4 \quad &\text{and} \quad \frac{\partial u_4}{\partial n_4} = g \text{ on } \Gamma, \quad \frac{\partial u_4}{\partial n} = 0 \text{ on } \partial\Omega_4 \setminus \Gamma
\end{aligned} \tag{54}$$

for $g \in L^2(\Gamma)$, where $\Gamma = \Gamma_{12} \cup \Gamma_{13} \cup \Gamma_{42} \cup \Gamma_{43} \cup \Gamma_\Pi$ here. As with the two-subdomains case, the natural Neumann boundary conditions have been modified. In order to reach a more compact formulation of the Lagrange functional later on, we will make again use of the weak formulation of Eqs. (54),

$$\begin{aligned}
a_1(u_1, v_1) &= b_1(v_1) + (g, v_1)_\Gamma, \qquad \forall v_1 \in V_1,\ u_1 \in V(\Omega_1) \\
a_2(u_2, v_2) &= b_2(v_2) - (g, v_2)_\Gamma, \qquad \forall v_2 \in V_2,\ u_2 \in V(\Omega_2) \\
a_3(u_3, v_3) &= b_3(v_3) - (g, v_3)_\Gamma, \qquad \forall v_3 \in V_3,\ u_3 \in V(\Omega_3) \\
a_4(u_4, v_4) &= b_4(v_4) + (g, v_4)_\Gamma, \qquad \forall v_4 \in V_4,\ u_4 \in V(\Omega_4)
\end{aligned} \tag{55}$$

in the sequel. Furthermore, the objective functional reads

$$\begin{aligned}
J_\Gamma(u_1, u_2, u_3, u_4, g) := \frac{1}{2}\Bigg\{ &\sum_{i=2,3} \int_{\Gamma_{1i}} (u_{1|\Gamma_{1i}} - u_{i|\Gamma_{1i}})^2 \, dx + (u_{1|\Gamma_\Pi} - u_{i|\Gamma_\Pi})^2 + \\
&\sum_{i=2,3} \int_{\Gamma_{4i}} (u_{4|\Gamma_{4i}} - u_{4|\Gamma_{4i}})^2 \, dx + (u_{4|\Gamma_\Pi} - u_{i|\Gamma_\Pi})^2 + \gamma \int_\Gamma g^2 \, dx \Bigg\}.
\end{aligned} \tag{56}$$

In contrast to the two-subdomains case, four constraints for the unknowns at $\Gamma_{\Gamma\Pi}$ are implicitly applied here: $u_{1|\Gamma} = u_{2|\Gamma}, u_{1|\Gamma} = u_{3|\Gamma}, u_{4|\Gamma} = u_{2|\Gamma}$, and $u_{4|\Gamma} = u_{3|\Gamma}$, which turned out to have an impact on the step size selection of gradient-based methods in experiments, as will be described later on.

Finally, the problem of optimal control is stated as

$$\min_{u_1,\dots,u_4,g} J_\Gamma(u_1,u_2,u_3,u_4,g) \tag{57}$$
$$\text{subject to (54)}.$$

3.2 The Optimality System

Although an iterative, gradient-based method shall be used for the solving of Eq. (57), the Lagrange functional

$$L(u_1,\dots,u_4,g,\lambda_1,\dots,\lambda_4) := J_\Gamma(u_1,u_2,u_3,u_4,g) - \tag{58}$$
$$\sum_{i=1}^{4} a_i(u_i,\lambda_i) + b_i(\lambda_i) + (-1)^{i-1}\big(g,\lambda_i\big)_\Gamma \tag{59}$$

and its optimality system are first elaborated here, since the involved adjoint equations do also appear with the gradient-based algorithm. Again, partially deriving for the Lagrange multiplier functions $\lambda_1,\dots,\lambda_4$ yields the constraint or state equations

$$\left\langle \frac{\partial L}{\partial \lambda_i}, v_i \right\rangle = 0 \quad \Leftrightarrow \quad a_i(u_i,v_i) = b_i(v_i) + (-1)^{i-1}\big(g,v_i\big)_\Gamma, \quad i=1,\dots,4\,, \tag{60}$$

whereas deriving for the control function g results in the optimality condition

$$\left\langle \frac{\partial L}{\partial g}, \tilde{g} \right\rangle = 0, \quad \forall \tilde{g} \quad \Rightarrow \quad \gamma(g,\tilde{g})_\Gamma + \sum_{i=1}^{4}(-1)^{i-1}\big(\tilde{g},\lambda_i\big)_\Gamma = 0\,. \tag{61}$$

The adjoint equations here read as follows:

$$\begin{aligned}
a_1'(u_1,v_1,\lambda_1) &= \sum_{i=2,3} \big(u_{1|\Gamma_{1i}} - u_{2|\Gamma_{1i}}, v_{1|\Gamma_{1i}}\big)_{\Gamma_{1i}} + \big(u_{1|\Gamma_\Pi} - u_{i|\Gamma_\Pi}\big)v_{1|\Gamma_\Pi} \\
a_2'(u_2,v_2,\lambda_2) &= \sum_{i=1,4} \big(u_{i|\Gamma_{i2}} - u_{2|\Gamma_{i2}}, -v_{2|\Gamma_{i2}}\big)_{\Gamma_{i2}} - \big(u_{i|\Gamma_\Pi} - u_{2|\Gamma_\Pi}\big)v_{2|\Gamma_\Pi} \\
a_3'(u_3,v_3,\lambda_3) &= \sum_{i=1,4} \big(u_{i|\Gamma_{i3}} - u_{3|\Gamma_{i3}}, -v_{3|\Gamma_{i3}}\big)_{\Gamma_{i3}} - \big(u_{i|\Gamma_\Pi} - u_{3|\Gamma_\Pi}\big)v_{3|\Gamma_\Pi} \\
a_4'(u_4,v_4,\lambda_4) &= \sum_{i=2,3} \big(u_{4|\Gamma_{4i}} - u_{i|\Gamma_{4i}}, v_{4|\Gamma_{4i}}\big)_{\Gamma_{4i}} + \big(u_{4|\Gamma_\Pi} - u_{i|\Gamma_\Pi}\big)v_{4|\Gamma_\Pi}
\end{aligned} \tag{62}$$

with $a'(\cdot,\cdot,\cdot)$ as defined in Eq. (21).

3.3 Calculation of the Gradient

Since the general mathematical structure for computation of the gradient $\left\langle \frac{dJ_\Gamma}{dg}, \tilde{g} \right\rangle$, is described in Section 2.3.1, we immediately proceed to the explicit calculation for the model problem.

The gradient formulation involving the sensitivities, i.e. variation directions, $\tilde{u}_1, \ldots, \tilde{u}_4$, here reads

$$\left\langle \frac{dJ_\Gamma}{dg}, \tilde{g} \right\rangle = \sum_{i=2,3} \left(u_{1|\Gamma_{1i}} - u_{i|\Gamma_{1i}}, \tilde{u}_{1|\Gamma_{1i}} - \tilde{u}_{i|\Gamma_{1i}}\right)_{\Gamma_{1i}} + \sum_{i=2,3} \left(u_{1|\Gamma_\Pi} - u_{i|\Gamma_\Pi}, \tilde{u}_{1|\Gamma_\Pi} - \tilde{u}_{i|\Gamma_\Pi}\right)_{\Gamma_\Pi}$$
$$+ \sum_{i=2,3} \left(u_{4|\Gamma_{4i}} - u_{i|\Gamma_{4i}}, \tilde{u}_{4|\Gamma_{4i}} - \tilde{u}_{i|\Gamma_{4i}}\right)_{\Gamma_{4i}} + \sum_{i=2,3} \left(u_{4|\Gamma_\Pi} - u_{i|\Gamma_\Pi}, \tilde{u}_{4|\Gamma_\Pi} - \tilde{u}_{i|\Gamma_\Pi}\right)_{\Gamma_\Pi}$$
$$+ \gamma(g, \tilde{g})_\Gamma. \quad (63)$$

The associated sensitivity equations are of the form

$$\begin{aligned}
a_1'(u_1, \tilde{u}_1, v_1) &= \sum_{i=2,3} (\tilde{g}_{1|\Gamma_{1i}}, v_{1|\Gamma_{1i}})_{\Gamma_{1i}} + \tilde{g}_{|\Gamma_\Pi} v_{1|\Gamma_\Pi} \\
a_2'(u_2, \tilde{u}_2, v_2) &= -\sum_{i=1,i4} (\tilde{g}_{|\Gamma_{i2}}, v_{2|\Gamma_{i2}})_{\Gamma_{i2}} - \tilde{g}_{|\Gamma_\Pi} v_{2|\Gamma_\Pi} \\
a_3'(u_3, \tilde{u}_3, v_3) &= -\sum_{i=1,4} (\tilde{g}_{|\Gamma_{i3}}, v_{3|\Gamma_{i3}})_{\Gamma_{i3}} - \tilde{g}_{|\Gamma_\Pi} v_{3|\Gamma_\Pi} \\
a_4'(u_4, \tilde{u}_4, v_4) &= \sum_{i=2,3} (\tilde{g}_{|\Gamma_{4i}}, v_{4|\Gamma_{4i}})_{\Gamma_{4i}} + \tilde{g}_{|\Gamma_\Pi} v_{4|\Gamma_\Pi}
\end{aligned} \quad (64)$$

corresponding to Equations (33), for $i = 1, \ldots, 4$, in the generic formulation.

Again, by setting $v_i = \tilde{u}_i$ in Eq. (62), and $v_i = \lambda_i$ in Eq. (64) we gain relations

$$\begin{aligned}
\sum_{i=2,3} \left(u_{1|\Gamma_{1i}} - u_{2|\Gamma_{1i}}, \tilde{u}_{1|\Gamma_{1i}}\right)_{\Gamma_{1i}} + \left(u_{1|\Gamma_\Pi} - u_{i|\Gamma_\Pi}\right) \tilde{u}_{1|\Gamma_\Pi} &= \sum_{i=2,3} (\tilde{g}_{1|\Gamma_{1i}}, \lambda_{1|\Gamma_{1i}})_{\Gamma_{1i}} + \tilde{g}_{|\Gamma_\Pi} \lambda_{1|\Gamma_\Pi} \\
\sum_{i=1,4} \left(u_{i|\Gamma_{i2}} - u_{2|\Gamma_{i2}}, -\tilde{u}_{2|\Gamma_{i2}}\right)_{\Gamma_{i2}} - \left(u_{i|\Gamma_\Pi} - u_{2|\Gamma_\Pi}\right) \tilde{u}_{2|\Gamma_\Pi} &= -\sum_{i=1,4} (\tilde{g}_{|\Gamma_{i2}}, \lambda_{2|\Gamma_{i2}})_{\Gamma_{i2}} - \tilde{g}_{|\Gamma_\Pi} \lambda_{2|\Gamma_\Pi} \\
\sum_{i=1,4} \left(u_{i|\Gamma_{i3}} - u_{3|\Gamma_{i3}}, -\tilde{u}_{3|\Gamma_{i3}}\right)_{\Gamma_{i3}} - \left(u_{i|\Gamma_\Pi} - u_{3|\Gamma_\Pi}\right) \tilde{u}_{3|\Gamma_\Pi} &= -\sum_{i=1,4} (\tilde{g}_{|\Gamma_{i3}}, \lambda_{3|\Gamma_{i3}})_{\Gamma_{i3}} - \tilde{g}_{|\Gamma_\Pi} \lambda_{3|\Gamma_\Pi} \\
\sum_{i=2,3} \left(u_{4|\Gamma_{4i}} - u_{i|\Gamma_{4i}}, \tilde{u}_{4|\Gamma_{4i}}\right)_{\Gamma_{4i}} + \left(u_{4|\Gamma_\Pi} - u_{i|\Gamma_\Pi}\right) \tilde{u}_{4|\Gamma_\Pi} &= \sum_{i=2,3} (\tilde{g}_{|\Gamma_{4i}}, \lambda_{4|\Gamma_{4i}})_{\Gamma_{4i}} + \tilde{g}_{|\Gamma_\Pi} \lambda_{4|\Gamma_\Pi}
\end{aligned} \quad (65)$$

between the sensitivities $\tilde{u}_i$ and the co-states λ_i. Applying these relations to replace the sensitivities in Eq. (63) by the co-states, we obtain

$$\left\langle \frac{dJ}{dg}, \tilde{g} \right\rangle = \sum_{i=2,3} \left(\tilde{g}_{|\Gamma_{1i}}, \lambda_{1|\Gamma_{1i}} - \lambda_{i|\Gamma_{1i}}\right)_{\Gamma_{1i}} + \tilde{g}_{|\Gamma_\Pi} \left(\lambda_{1|\Gamma_\Pi} - \lambda_{i|\Gamma_\Pi}\right)$$
$$+ \sum_{i=2,3} \left(\tilde{g}_{|\Gamma_{i4}}, \lambda_{4|\Gamma_{i4}} - \lambda_{i|\Gamma_{i4}}\right)_{\Gamma_{i4}} + \tilde{g}_{|\Gamma_\Pi} \left(\lambda_{4|\Gamma_\Pi} - \lambda_{i|\Gamma_\Pi}\right) + \gamma(g, \tilde{g})_\Gamma, \quad (66)$$

i.e. the gradient formulation independent of $\tilde{g}$, which is analogical to Eq. (40) for the two-subdomains case. The explicit formulation reads

$$\frac{dJ}{dg} = \sum_{i=2,3} P_{\Gamma_{1i}}\big(\lambda_{1|\Gamma_{1i}} - \lambda_{i|\Gamma_{1i}}\big) + P_{\Gamma_\Pi}\big(\lambda_{1|\Gamma_\Pi} - \lambda_{i|\Gamma_\Pi}\big) + \sum_{i=2,3} P_{\Gamma_{4i}}\big(\lambda_{4|\Gamma_{i4}} - \lambda_{i|\Gamma_{i4}}\big)_{\Gamma_{i4}} + P_{\Gamma_\Pi}\big(\lambda_{4|\Gamma_\Pi} - \lambda_{i|\Gamma_\Pi}\big) + \gamma g. \quad (67)$$

Note that the gradient at Γ_Π is the sum of four differences, in contrast to the other points $\Gamma \setminus \Gamma_\Pi$. I.e. the magnitude of $\frac{dJ}{dg}$ at this corner point is in average four times larger as it is for the remaining locations. Since the step size in step (e) of Alg. (1) is selected for the whole gradient, experiments have shown that it is systematically chosen too large at Γ_Π, which leads to divergence even when employing line search. As a remedy, an additional scaling factor ν for $\frac{dJ}{dg}$ at Γ_Π is introduced, where a value of $\frac{1}{4}$ has shown to be, at least empirically.

3.4 Outer CG Iteration

To sum up, for the four-subdomains case, steps (a)–(c) of Algorithm 1 have to be replaced by:

(a) solve the state equations for u_1^k, u_2^k (in parallel):

$$\begin{aligned} a_1(u_1, v_1) &= b_1(v_1) + \big(g, v_1\big)_\Gamma, && \forall v_1 \in V(\Omega_1), \\ a_2(u_2, v_2) &= b_2(v_2) - \big(g, v_2\big)_\Gamma, && \forall v_2 \in V(\Omega_2), \\ a_3(u_3, v_3) &= b_3(v_3) - \big(g, v_3\big)_\Gamma, && \forall v_3 \in V(\Omega_3), \\ a_4(u_4, v_4) &= b_4(v_4) + \big(g, v_4\big)_\Gamma, && \forall v_4 \in V(\Omega_4), \end{aligned} \quad (68)$$

(b) solve the adjoint equations for λ_1^k, λ_2^k (in parallel):

$$\begin{aligned} a_1'(u_1, v_1, \lambda_1) &= \sum_{i=2,3} \big(u_{1|\Gamma_{1i}} - u_{2|\Gamma_{1i}}, v_{1|\Gamma_{1i}}\big)_{\Gamma_{1i}} + \big(u_{1|\Gamma_\Pi} - u_{i|\Gamma_\Pi}\big) v_{1|\Gamma_\Pi}, && \forall v_1 \in V(\Omega_1), \\ a_2'(u_2, v_2, \lambda_2) &= \sum_{i=1,4} \big(u_{i|\Gamma_{i2}} - u_{2|\Gamma_{i2}}, -v_{2|\Gamma_{i2}}\big)_{\Gamma_{i2}} - \big(u_{i|\Gamma_\Pi} - u_{2|\Gamma_\Pi}\big) v_{2|\Gamma_\Pi}, && \forall v_2 \in V(\Omega_2), \\ a_3'(u_3, v_3, \lambda_3) &= \sum_{i=1,4} \big(u_{i|\Gamma_{i3}} - u_{3|\Gamma_{i3}}, -v_{3|\Gamma_{i3}}\big)_{\Gamma_{i3}} - \big(u_{i|\Gamma_\Pi} - u_{3|\Gamma_\Pi}\big) v_{3|\Gamma_\Pi}, && \forall v_3 \in V(\Omega_3), \\ a_4'(u_4, v_4, \lambda_4) &= \sum_{i=2,3} \big(u_{4|\Gamma_{4i}} - u_{i|\Gamma_{4i}}, v_{4|\Gamma_{4i}}\big)_{\Gamma_{4i}} + \big(u_{i|\Gamma_\Pi} - u_{4|\Gamma_\Pi}\big) v_{4|\Gamma_\Pi}, && \forall v_4 \in V(\Omega_4) \end{aligned} \quad (69)$$

(c) calculate the gradient:

$$\nabla J_\Gamma^k \leftarrow \sum_{i=2,3} P_{\Gamma_{1i}}\big(\lambda_{1|\Gamma_{1i}} - \lambda_{i|\Gamma_{1i}}\big) + \nu P_{\Gamma_\Pi}\big(\lambda_{1|\Gamma_\Pi} - \lambda_{i|\Gamma_\Pi}\big) + \sum_{i=2,3} P_{\Gamma_{4i}}\big(\lambda_{4|\Gamma_{i4}} - \lambda_{i|\Gamma_{i4}}\big)_{\Gamma_{i4}} + \nu P_{\Gamma_\Pi}\big(\lambda_{4|\Gamma_\Pi} - \lambda_{i|\Gamma_\Pi}\big) + \gamma g. \quad (70)$$

3.5 Experimental Results

Although having given a theoretical approximation of the scalability characteristics of the proposed method, the focus of the experimental studies was restricted to the feasibility for the 2×2 case and the influence of the parameters ν and γ.

The algorithm was run on the same input image, Fig. 2(b), as with the two-subdomains case, where the common boundary Γ here included the 64th row and 64th column of the discretized image plane. The discretization, the local solving method as well as all the parameter values, except the starting interval of the step size selection which was $[0, 10]$, have been the same as with the previous experiments. Results are depicted in Fig. 4.

Besides the influence of the control regularization strength γ, also the impact of the step size reduction ν at Γ_Π were studied, see the diagrams in Fig. 4(a) and (b), respectively. Figure 4(a) shows that the convergence behavior for the 2×2 case has not changed significantly in comparison to the two-subdomains case, despite the fact that the rate has deteriorated to ≈ 0.93 after 50 iterations. Leaving out the control regularization has no significant impact on the L^2 error in comparison to setting γ to values small than 10^{-4}. Moreover, studies for the step size reduction factor ν revealed that only values less or equal to $\frac{1}{2}$ led to convergence, whereas greater values always led to divergence at Γ_Π.

Again, the result after 50 iterations in Fig. 4(c), as well as the per-pixel relative L^2 error shown in Fig. 4(d), are satisfactory for denoising purposes.

3.6 Complexity

As in the two-subdomains case, step (a) and (b) can be obviously carried out on different processors, which holds also for the case of $M_x \times M_y$ subdomains ($M_x > 2, M_y > 2$).

With respect to inter-process communication for the $M_x \times M_y$ case, note that only variables lying on the interface Γ have to be exchanged within the main loop. To be explicit, the control vector g^k has to be distributed by a central process, carrying out the steps (c)–(g), to the subdomain processes before step (a), which amounts to sending approximately $\frac{4\sqrt{N}}{\sqrt{M}}$ unknowns per subdomain, where $M := M_x \cdot M_y$ and N is the total number of unknowns, i.e. $M\frac{4\sqrt{N}}{\sqrt{M}} = 4\sqrt{MN}$ in total. Before step (b), again only variables on Γ have to be communicated in order to set-up the right-hand sides, but this time only mutually among processes which have adjacent subdomains; i.e. a distributed communication step can be applied, which amounts to interchanging only approx. $\frac{4\sqrt{N}}{\sqrt{M}}$ variables sequentially. Furthermore, before step (c), the local co-states vectors λ_i^k, being restricted to their local interfaces Γ_i, are gathered by the central process, again resulting in a sequential communication volume of $4\sqrt{MN}$ variables. Finally, the total amount of bytes to be communicated sequentially is given by the expression

$$V(M,k) = k\left(8\sqrt{MN} + 4\frac{\sqrt{N}}{\sqrt{M}}\right)v_v + V_c$$

where k denotes the total number of iterations, v_v denotes the size of one variable in bytes and V_c the constant amount of bytes for the initial distribution of the input vector f as well as the final collection of the local solutions u_i. Obviously, as it is with standard non-overlapping

domain decomposition methods, the communication volume scales only with the square root of the number of subdomains here, which is due to exchanging only the interface variables and indicates very good scalability properties.

With respect to the total computation time, we have the approximation

$$T(M) = T_{NL}(N/M) + T_L(N/M) + t_{byte}V(M, k(M)) + T_c$$

where $T_{NL}(n)$ and $T_L(n)$ denote the average computation time for solving the nonlinear and linear state systems, respectively, in parallel, T_c the computation time for the steps (c)–(g), which are independent from M, and with t_{byte} denoting the communication time per byte. Note that communication latency times as well as synchronization times have been neglected here. Furthermore, although the outer iteration number $k(M)$ naturally increases with the number of subdomains M, the processing time of the computationally demanding steps (a) and (b) is supposed to decrease strongly, depending on the complexity of the inner solving methods.

Finally, note that depending on the applied line search procedure, coarse-grained parallelization can be employed also there. In the case of nested iteration for example, the two necessary evaluations of J_Γ within each iteration can be carried out simultaneously.

4 Conclusion and Further Work

We investigated the parallel solution of a class of nonlinear variational approaches to image processing on non-overlapping domains. The basic idea was to incorporate a control variable into the overall problem to enforce the compatibility of locally computed solutions at boundaries of the subdomains. The approach is similar to substructuring methods that are established for linear (systems of) partial differential equations. It can, however, applied to nonlinear problems without modification. Inter-processor communication is minimized by restricting data interchange to lower-dimensional subdomain boundaries. The theoretical derivation was experimentally illustrated by solving in parallel a nonlinear variational model problem (TV-based image denoising) on 2×2 subdomains, implying validity of the approach for an arbitrary number of subdomains.

Further work will focus on preconditioners for improving the convergence rate of the outer iteration loop, and on nonlinear variational problems in various application areas of image processing.

Acknowledgment

Support by the Deutsche Forschungsgemeinschaft (grant: Schn 457/4) is gratefully acknowledged.

References

[1] K. Atkinson and W. Han. *Theoretical Numerical Analysis*. Springer, Heidelberg, Germany, 2001.

[2] J.P. Aubin. *Approximation of Elliptic Boundary-Value Problem*. John Wiley & Sons, New York, USA, 1972.

[3] L. Badea. On the Schwarz alternating method with more than two subdomains for nonlinear monotone problems. *SIAM J. of Numer. Anal.*, 28(179–204), 1991.

[4] D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, Belmont, MA, USA, 2 edition, 1999.

[5] X.C. Cai and M. Dryja. Domain decomposition methods for monotone nonlinear elliptic problems. In D. Keyes and J. Xu, editors, *Domain Decomposition Methods in Scientific and Engineering Computing*, pages 335–360, Providence, RI, USA, 1994. AMS.

[6] T.F. Chan, G.H. Golub, and P. Mulet. A nonlinear primal-dual method for Total Variation-based image restoration. In M. Berger, R. Deriche, I. Herlin, J. Jaffre, and J. Morel, editors, *ICAOS'96, 12th Int. Conf. on Analysis and Optimization of sytems: Images, wavelets, and PDEs*, volume 219 of *Lecture Notes in Control and Information Sciences*, pages 241–252, Heidelberg, Germany, 1996. Springer.

[7] T.F. Chan, G.H. Golub, and P. Mulet. A nonlinear primal-dual method for tv-based image restoration. *SIAM J. of Scientifc Computing*, 20(6):1964–1977, 1999.

[8] T.F. Chan, H. Zhou, and R. Chan. A continuation method for total variation denoising problems. In F.T. Luk, editor, *Proc. of the SPIE Conference on Advanced Signal Processing Algorithms*, pages 314–325. SPIE, 1995.

[9] P.G. Ciarlet. *The Finite Element Method for Elliptic Problem*. North-Holland, Amsterdam, The Netherlands, 1978.

[10] D.C. Dobson and C.R. Vogel. Convergence of an iterative method for total variation denoising. *SIAM J. of Numer. Anal.*, 34(5):1779–1791, 1997.

[11] M. Fortin and R. Aboulaich. Schwarz's decomposition method for incompressible flow problems. In R. Glowinski, G. Golub, G. Meurant, and J.Periaux, editors, *First Int. Symp. on Domain Decomposition Methods. for Part. Diff. Equations*, pages 333–349, Philadelphia, PA, USA, 1988. SIAM.

[12] S. Fučik, A. Kratochvil, and J. Nečcas. Kačanov-galerkin method. *Comment. Math. Univ. Carolinae*, 14(4):651–659, 1973.

[13] M. Gunzburger, H. Lee, and J. Peterson. An optimization based domain decomposition method for partial differential equations. *Comput. Math. Appl.*, 37(10):77–93, 1999.

[14] Max D. Gunzburger. *Perspectives in Flow Control and Optimization*. SIAM, Philadelphia, USA, 2003.

[15] M.D. Gunzburger and H.K. Lee. Analysis, approximation, and comutation of a coupled solid/fluid temperature control problem. *Comput. Meth. Appl. Mech. Engrg.*, 118:133–152, 1993.

[16] M.D. Gunzburger and H.K. Lee. An optimization-based domain decomposition method for the navier-stokes equations. *SIAM J. of Numer. Anal.*, 37(5):1455–1480, 2000.

[17] J. Heers, C. Schnörr, and H.S. Stiehl. Globally convergent iterative numerical schemes for nonlinear variational image smoothing and segmentation on a multiprocessor machine. *IEEE Trans. Image Processing*, 10(6):852–864, June 2001.

[18] J. Kačúr, J. Nečas, J. Polák, and J. Souček. Convergence of a method for solving the magnetostatic filed in nonlinear media. *Aplikace Mathematiky*, 13:456–465, 1968.

[19] T. Kohlberger, C. Schnörr, A. Bruhn, and J. Weickert. Domain decomposition for parallel variational optical flow compuation. In B. Michaelis and G. Krell, editors, *Pattern Recognition, Proc. 25th DAGM Symposium*, volume 2781 of *LNCS*, pages 196–202. Springer, 2003.

[20] T. Kohlberger, C. Schnörr, A. Bruhn, and J. Weickert. Parallel variational motion estimation by domain decomposition and cluster computing. In T. Pajdla and J. Matas, editors, *8th European Conf. on computer Vision (ECCV 2004)*, volume 3024 of *Springer LNCS*, pages 205–216, Prague, Czech Republic, 2004.

[21] T. Kohlberger, C. Schnörr, A. Bruhn, and J. Weickert. Domain decomposition for variational optical flow computation. *IEEE Trans. Image Processing*, 2005.

[22] H.K. Lee. *Optimization Based Domain Decomposition Methods for Linear and Nonlinear Problems*. PhD thesis, Faculty of the Virginia Polytechnic Inst. and State Univ., Blacksburg, Virginia, USA, 1997.

[23] J.L. Lions. *Optimal Control of Systems Governed by Partial Differential Equations*, volume 170 of *Die Grundlagen der math. Wissenschaften in Einzeldarstellungen*. Springer, Heidelberg, Germany, 1971.

[24] S.-H. Lui. On monotone iteration and Schwarz methods for nonlinear parabolic pdes. *J. Comput. Appl. Math.*, 161:449–468, 2003.

[25] A. Quarteroni and A. Valli. *Domain Decomposition Methods for Partial Differential Equations*. Oxford University Press, Oxford, UK, 1999.

[26] L.I. Rudin and S. Osher. Total variation based image restoration with free local constraints. In *IEEE Proc. Int. Conf. Image Proc.*, volume 1, pages 31–35, 1994.

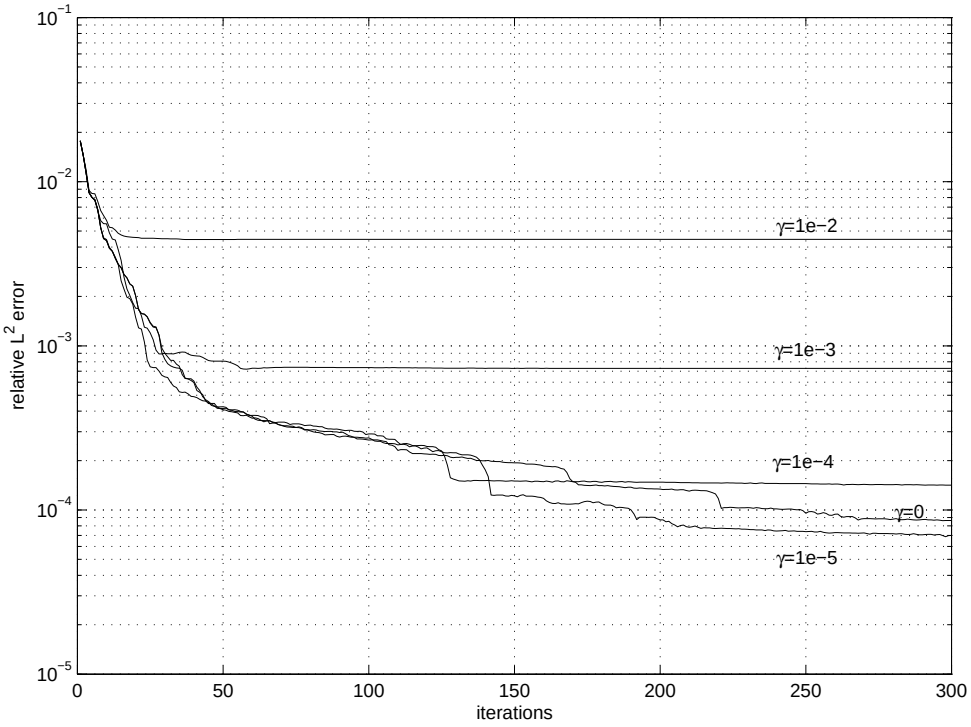
[27] L.I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60:259–268, 1992.

[28] H.R. Schwarz. *Methode der finiten Elemente*. B.G. Teubner, Stuttgart, Germany, 1980.

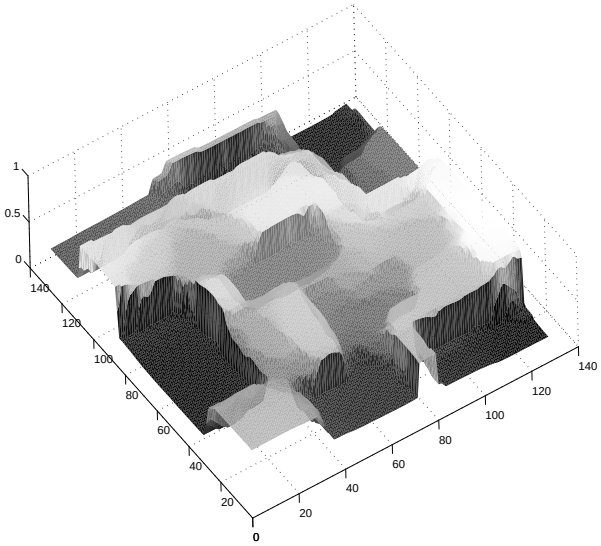
[29] C.R. Vogel and M.E. Oman. Iterative methods for total variation denoising. *SIAM J. of Scientifc Computing*, 17(1):227–238, 1996.

[30] C.R. Vogel and M.E. Oman. Fast, robust total variation-based reconstruction of noisy, blurred images. *IEEE Trans. Image Processing*, 7(6):813–824, 1998.

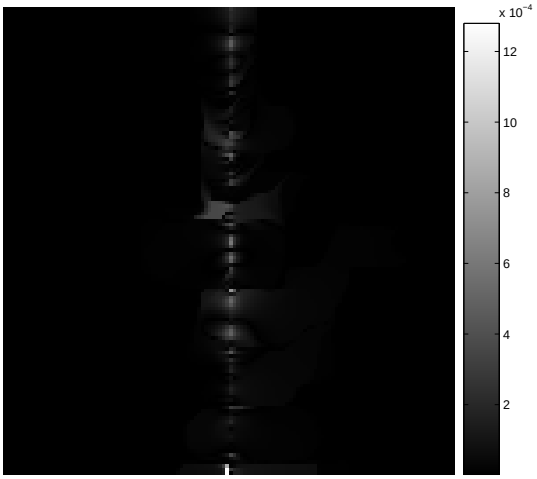
[31] W.P. Ziemer. *Weakly differentiable functions*. Springer, Heidelberg, Germany, 1989.



(a) Relative L^2 errors for varying control regularization strengths γ

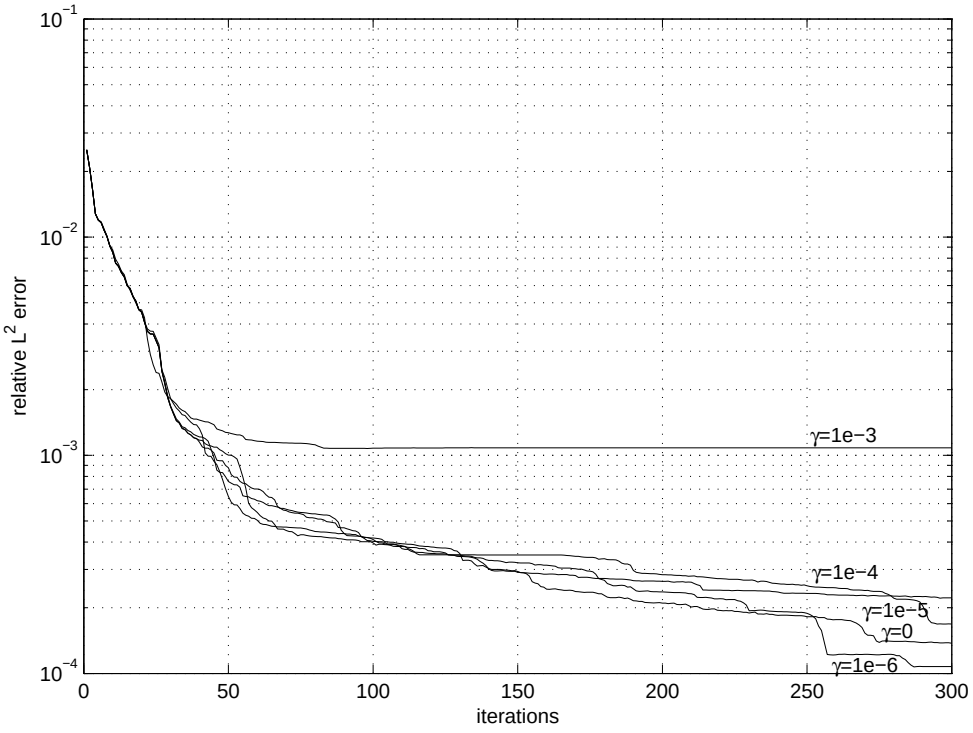


(b) Result after 50 iterations ($\gamma = 10^{-5}$)

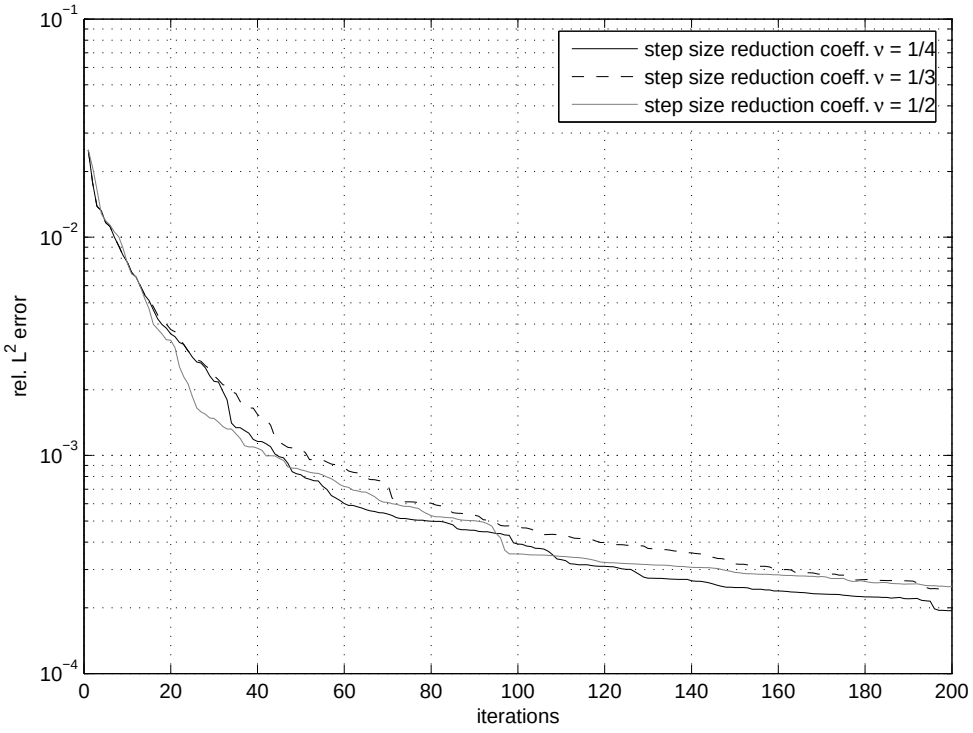


(c) Per-pixel L^2 error after 300 iterations ($\gamma = 10^{-5}$)

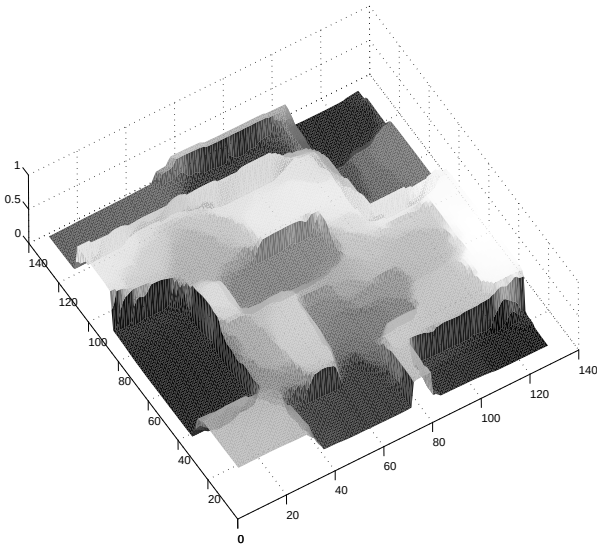
Figure 3: L^2 errors and result for the two-subdomains decomposition. TV regularization strength $\alpha = 1.0$, perturbation $\beta = 10^{-6}$.



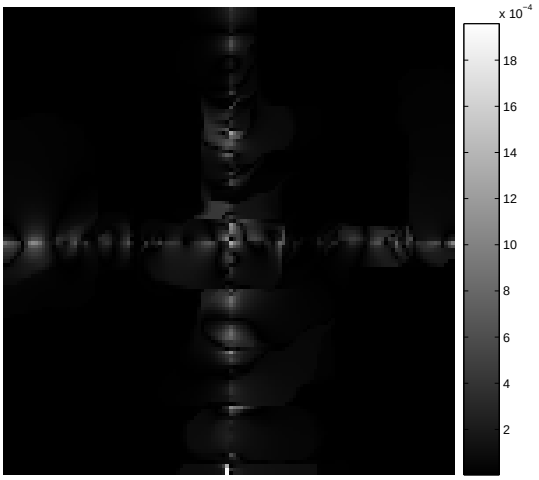
(a) Relative L^2 error for varying control regularization strengths γ



(b) Relative L^2 error for varying step size reduction factors ν ($\gamma = 0$)



(c) Result after 50 iterations ($\gamma = 10^{-5}$)



(d) Per-pixel L^2 error after 300 iterations ($\gamma = 10^{-5}$)

Figure 4: L^2 errors and result for the 2×2-decomposition. TV regularization strength $\alpha = 1.0$, perturbation $\beta = 10^{-6}$, step size reduction factor $\nu = \frac{1}{4}$ for (b)–(d).