

Shape optimization of a breakwater

Moritz Keuthen^{a,*}, Daniel Kraft^b

^a*Chair of Mathematical Optimization, Department of Mathematics, Technische Universität München, Boltzmannstr. 3, 85748 Garching b. München, Germany*

^b*Institute of Mathematics, University of Graz, NAWI Graz, Universitätsplatz 3, 8010 Graz, Austria*

Abstract

In this paper we optimize the shape of a breakwater which protects a harbor basin from incoming waves. More specifically, our objective is reducing the harbor resonance due to long range ocean waves. We will consider the complex valued Helmholtz equation as our model state equation and minimize the average wave height in the harbor basin with the shape of the breakwater as optimization variable. The geometry will be described by the level set method, i.e. the domain is given as the sub-zero level set of a function. In contrast to many publications we use the volume expression of the shape derivative, which lends itself naturally to a level set update via a transport equation. The volume expression requires less regular finite element functions than the Hadamard form. The model problem naturally allows to treat geometric constraints in the form of forbidden regions and we will compare different possibilities to handle them.

Keywords: Shape Optimization, Level Set Method, Helmholtz Equation, Harbor Resonance, Geometric Constraints

1. Introduction

Since the introduction of the level set method to shape optimization at the turn of the century it has developed into one of the most powerful techniques in this area. Some early works are [1, 2, 3, 4, 5]. Many publications deal with structural optimization, usually the Ersatz material approach in the hold-all domain D is used to compute the mechanical properties of the structure and the domain Ω is never explicitly resolved. But there are also approaches where the domain is exactly meshed, consult for instance [6, 7, 8, 9, 10, 11, 12]. The literature is extensive, we refer to the review paper [13] for an overview of level set based methods in structural topology and shape optimization. The survey [14] presents the level set method in combination with inverse problems and optimal design. For a more comprehensive exposition of the broad field of shape optimization we mention the monographs [15, 16, 17, 18, 19, 20, 21, 22, 23].

In this paper our objective is to reduce the resonance of a harbor due to long range ocean waves. In practice many publications (e.g. [24, 25, 26, 27, 28]) employ the mild slope equation to model wave effects in coastal areas. It was first derived by Berkhoff [29] and can be written as

$$\nabla \cdot CC_g \nabla \varphi + k^2 CC_g \varphi = 0, \quad (1)$$

where φ is the horizontal variation in velocity potential, k is the wave number, ω is the wave frequency, $C = \omega/k$ is the wave celerity, $C_g = \frac{C}{2} \left(1 + \frac{2kh}{\sinh 2kh}\right)$ is the group velocity and h is the water depth. Usually it is enriched with additional effects such as partial absorption boundaries, bottom friction, entrance loss, etc. In this note we will assume as simplification that the water depth is constant throughout which leads to the well-known Helmholtz equation

$$\Delta \varphi + k^2 \varphi = 0. \quad (2)$$

*Corresponding author

Email addresses: keuthen@ma.tum.de (Moritz Keuthen), daniel.kraft@uni-graz.at (Daniel Kraft)

Note that the potential φ in this case is complex valued. To the best of our knowledge such a shape optimization problem was only briefly considered in the thesis [30]. The author used the real-valued Helmholtz equation as state equation, an explicit discrete geometry description via the finite element mesh and computed the shape derivative by differentiating with respect to nodal coordinates.

We will consider the complex valued Helmholtz equation as our model state equation and want to minimize the average wave height in the harbor basin with the shape of the breakwater as optimization variable. The geometry will be described by the level set method [31, 32, 33, 34], i.e. the domain Ω is given as the sub-zero level set of a function $\Phi : D \rightarrow \mathbb{R}$, where $D \subset \mathbb{R}^2$ is the hold-all domain. In the numerical realization the level set function is given on a regular grid and we approximate $\partial\Omega$ with one or more polygonal curves. In each iteration we construct a triangulation of D which resolves the interface and use it to compute the state, adjoint state and gradient of the cost functional. The model state equation is given on an exterior domain. Since we want to use finite elements we follow the presentation in [35] and reformulate the problem on a bounded domain. In contrast to many publications we use the volume expression of the shape derivative, which lends itself naturally to a level set update via a transport equation. It requires less regular finite element functions and in our experience the volume expression is numerically more stable than the Hadamard form. This assessment is shared in the recent papers [36, 37]. In [37] the volume expression of the shape derivative and the level set method are also used.

The model problem naturally involves geometric constraints in the form of forbidden regions. We strictly enforce those constraints by projecting the gradient onto a suitable admissible set. From a theoretic point of view the scalar product used for the projection should be at least as smooth as the scalar product used to determine the gradient. In the numerical experiments we will study different choices of gradient and projection and see that sometimes a less regular projection leads to better results.

In the following we will use the shorthand notation \mathbf{X} for the product space X^d .

2. Description of the physical model

We study the situation depicted in Figure 1. We have an isle bounded by the contour Γ_L , some breakwaters given by Γ_B and a surrounding ocean denoted by Ω^+ . We want to compute the scattered wave u induced by an incoming planar monochromatic wave $z(x) = e^{ikd^T x}$ with incident direction $d \in \mathbb{R}^2$ and wave number $k > 0$. The total surface perturbation is given by $y = u + z$. Our mathematical model is given by

$$\begin{aligned} \Delta y + k^2 y &= 0 & \text{in } \Omega^+ \\ ay + \frac{\partial y}{\partial n} &= 0 & \text{on } \Gamma_I := \Gamma_B \cup \Gamma_L. \end{aligned} \quad (3)$$

Here $a : D \rightarrow \mathbb{C}$ describes the absorption coefficient at the boundary. The boundaries Γ_L, Γ_B are assumed to be Lipschitz. Let us discuss appropriate boundary conditions. It is a standard assumption that the scattered wave satisfies the Sommerfeld radiation condition

$$iku - \frac{\partial u}{\partial R} = o(R^{-\frac{1}{2}}), \quad \text{for } R \rightarrow \infty.$$

If we want to study this problem in weak form on the unbounded domain Ω^+ we need to introduce different weighted Sobolev spaces for the test and ansatz functions and include the Sommerfeld radiation condition in the ansatz space. See [38, chapter 4] and [35, section 2.3] for more details. This approach however leads to various difficulties in the numerical realization.

An alternative is to decompose the domain Ω^+ disjointly into a bounded domain Ω and an unbounded domain Ω_a by introducing an artificial smooth boundary Γ_a , such that $\Omega^+ = \Omega \cup \Gamma_a \cup \Omega_a$. The problem (3) is then equivalent to the following coupled problem (c.f. [39])

$$\begin{aligned} \Delta y_- + k^2 y_- &= 0 & \text{in } \Omega \\ ay_- + \frac{\partial y_-}{\partial n} &= 0 & \text{on } \Gamma_I \\ y_- &= y_+ & \text{on } \Gamma_a \\ \frac{\partial y_-}{\partial n} &= \frac{\partial y_+}{\partial n} & \text{on } \Gamma_a \\ \Delta y_+ + k^2 y_+ &= 0 & \text{in } \Omega_a \\ iku_+ - \frac{\partial u_+}{\partial R} &= o(R^{-\frac{1}{2}}), & \text{for } R \rightarrow \infty. \end{aligned} \quad (4)$$

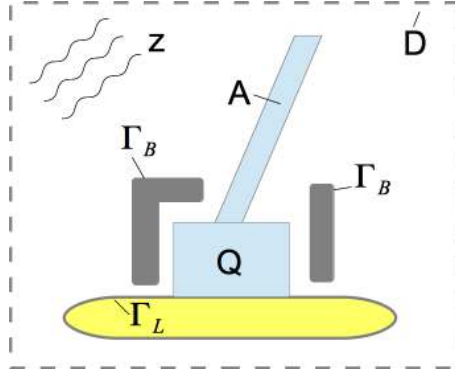


Figure 1: The domain Ω

For a given u_- on Γ_a one can solve the unbounded Dirichlet problem (recall $u = y - z$)

$$\begin{aligned} \Delta u_+ + k^2 u_+ &= 0 && \text{in } \Omega_a \\ u_+ &= u_- && \text{on } \Gamma_a \\ iku_+ - \frac{\partial u_+}{\partial R} &= o(R^{-\frac{1}{2}}), && \text{for } R \rightarrow \infty, \end{aligned}$$

compare [38]. If we have the solution u_+ we can easily compute $\frac{\partial u_+}{\partial n} = \frac{\partial u_-}{\partial n}$ on Γ_a . We denote the mapping $u_- \mapsto \frac{\partial u_-}{\partial n}$ by G_e and observe that $G_e \in \mathcal{L}(H^{\frac{1}{2}}(\Gamma_a), H^{-\frac{1}{2}}(\Gamma_a))$. This operator is called the Dirichlet-to-Neumann (DtN) operator. There exists an integral and a series representation of the non-local operator G_e , the integral variant is also called the Steklov-Poincaré operator. The Neumann condition $\frac{\partial y_-}{\partial n} = \frac{\partial y_+}{\partial n}$ on Γ_a can be reformulated by

$$\frac{\partial y_-}{\partial n} = \frac{\partial u_-}{\partial n} + \frac{\partial z}{\partial n} = G_e u_- + \frac{\partial z}{\partial n} = G_e y_- - G_e z + \frac{\partial z}{\partial n}.$$

Now we replace G_e by some yet unspecified $G \in \mathcal{L}(H^{\frac{1}{2}}(\Gamma_a), H^{-\frac{1}{2}}(\Gamma_a))$, which might be some approximation of the exact solution operator G_e . Hence we arrive at the bounded problem

$$\begin{aligned} \Delta y + k^2 y &= 0 && \text{in } \Omega \\ ay + \frac{\partial y}{\partial n} &= 0 && \text{on } \Gamma_I \\ \frac{\partial y}{\partial n} &= Gy - Gz + \frac{\partial z}{\partial n} && \text{on } \Gamma_a, \end{aligned} \quad (5)$$

which is equivalent to (3) for the choice $G = G_e$.

Let us fix some conventions and notation at this point. We define the usual bilinear L^2 -scalar product for real-valued functions on some set $A \subset \mathbb{R}^d$ as $(\cdot, \cdot)_{L^2(A)}$ and the corresponding sesquilinear form as $(f, g)_{L^2_{\mathbb{C}}(A)} := (f, \bar{g})_{L^2(A)}$ for some complex valued functions f, g . Furthermore we introduce the real-valued scalar product

$$(f, g)_{L^2_{\mathbb{R}}(A)} := \text{Re}(f, g)_{L^2_{\mathbb{C}}(A)}.$$

The norm $\|\cdot\|_{L^2_{\mathbb{R}}(A)}$ induced by this scalar product coincides with the norm induced by the sesquilinear form. Hence the elements of the space

$$L^2_{\mathbb{R}}(A) := \{f : A \mapsto \mathbb{C} \mid \|f\|_{L^2_{\mathbb{R}}(A)} < \infty\}$$

coincide with the elements of $\{f : A \mapsto \mathbb{C} \mid \|f\|_{L^2_{\mathbb{C}}(A)} < \infty\}$, but since we use the $(\cdot, \cdot)_{L^2_{\mathbb{R}}(A)}$ scalar product we have a different Hilbert space structure. Other Hilbert spaces will be treated analogously (e.g. $H^1_{\mathbb{R}}(A)$).

Let us get back to the model problem. We make the following simplifying assumption:

Assumption 1. We choose G as the 0th-order approximation of G_e

$$\langle Gy, \varphi \rangle_{H^{-\frac{1}{2}}(\Gamma_a), H^{\frac{1}{2}}(\Gamma_a)} = (iky, \varphi)_{L^2_{\mathbb{R}}(\Gamma_a)},$$

(c.f. [35, section 3.]). Furthermore the absorption coefficient is set to $a \equiv 0$ (i.e. perfect reflection).

Remark 1. The choice of G is a rather crude simplification, but in this paper our focus is more on methodology and not so much on realistic modeling. The low-order approximation of G_e introduces artificial reflections at Γ_a , but since this makes the shape optimization problem presumably harder we accept that for the moment. For more evolved methods of treating the artificial boundary Γ_a and the operator G_e we refer to [35, chapter 3].

Setting the absorption coefficient to zero implies that there is no damping effect by absorption of energy at the reflecting boundary. We note again that this presumably makes the optimization problem harder since small design changes might have large non-local effects because of wave interference. Furthermore, while breakwaters are efficient in reducing the wave amplitude for waves with short wave periods (like wind waves), this does not hold for long wave periods. Harbor oscillations and resonance due to long waves has been widely studied in coastal engineering literature, see for example [40] and the references therein.

The weak formulation of (5), with G, a chosen to satisfy Assumption 1, reads

$$\begin{cases} \text{Find } y \in H_{\mathbb{R}}^1(\Omega) : \\ b(y, \varphi) = f(\varphi), \quad \forall \varphi \in H_{\mathbb{R}}^1(\Omega), \end{cases} \quad (6)$$

where we define

$$\begin{aligned} b(y, \varphi) &:= (\nabla y, \nabla \varphi)_{L_{\mathbb{R}}^2(\Omega)} - k^2(y, \varphi)_{L_{\mathbb{R}}^2(\Omega)} - (iky, \varphi)_{L_{\mathbb{R}}^2(\Gamma_a)}, \\ f(\varphi) &:= \left(\frac{\partial z}{\partial n} - ikz, \varphi\right)_{L_{\mathbb{R}}^2(\Gamma_a)}. \end{aligned} \quad (7)$$

Results concerning the existence of a unique solution of (6) and its regularity are well-known:

Theorem 1. *Let Ω be a Lipschitz domain. Then there exists a unique solution $y \in H_{\mathbb{R}}^1(\Omega)$ of (6) for any right-hand side $f \in H_{\mathbb{R}}^1(\Omega)^*$ and we have $\|y\|_{H_{\mathbb{R}}^1(\Omega)} \leq c\|f\|_{H_{\mathbb{R}}^1(\Omega)^*}$ for some $c > 0$.*

PROOF. We have the Gelfand-triple $H_{\mathbb{R}}^1(\Omega) \hookrightarrow L_{\mathbb{R}}^2(\Omega) \hookrightarrow H_{\mathbb{R}}^1(\Omega)^*$. Further $b(\cdot, \cdot)$ is $H_{\mathbb{R}}^1(\Omega)$ -coercive. Hence the Fredholm alternative holds: Either there exists a unique solution of (6) for any $f \in H_{\mathbb{R}}^1(\Omega)$ or there exists a nontrivial solution $y_0 \neq 0$ of the homogenous problem

$$b(y, \varphi) = 0, \quad \forall \varphi \in H_{\mathbb{R}}^1(\Omega).$$

For our choice of G the solution of the homogenous problem is unique [35, Theorem 3.2].

Theorem 2. *Let Ω be a C^2 -domain. The solution $y \in H_{\mathbb{R}}^1(\Omega)$ of (6) has the additional regularity $y \in H_{\mathbb{R}}^2(\Omega)$.*

PROOF. Follows from standard regularity results for elliptic equations c.f. [41, Theorem 9.1.20].

3. Shape optimization problem

As announced in the introduction our objective is to minimize the average wave height in the harbor basin Q (compare Figure 1). Given a solution y of (6) we define the cost functional as

$$J(\Omega, y) = \frac{1}{2} \|y\|_{L_{\mathbb{R}}^2(Q)}^2.$$

Obviously enclosing the whole harbor basin by a breakwater is not a feasible solution, so we have to introduce an harbor approach A and demand that $Q \cup A$ is always part of the ocean. Furthermore we do not want to remove parts of the island (the inhabitants might complain). Hence we define the admissible set of domains to be

$$O_{ad} := \{\Omega \subset D \mid (Q \cup A) \subset \Omega, \Omega \cap L = \emptyset\},$$

where $\Omega \in O_{ad}$ represents the ocean. Of course one could impose additional constraints, e.g. a volume constraint on the set of admissible domains. We can now formulate the abstract shape optimization problem

$$\text{minimize } J(\Omega, y) \text{ such that } \Omega \in O_{ad}, \text{ and } y \text{ solves (6) on } \Omega.$$

In this paper we will not concern ourselves with the question of existence of a solution of the shape optimization problem. For a detailed discussion of techniques and conditions which guarantee the existence and uniqueness of solutions to shape optimization problems, we refer to [18, 19]. In order to find a solution (if it exists) we develop a framework for a gradient descent method. For this we follow the usual optimal control approach of introducing a design-to-state mapping $\Omega \mapsto y(\Omega)$ and computing the derivative of the reduced objective $j(\Omega) := J(\Omega, y(\Omega))$. In the next section we outline the theory of computing shape derivatives and apply it to our problem setting.

4. Shape sensitivity analysis

Following the presentation in [42, 18] we introduce a bounded hold-all domain $D \subset \mathbb{R}^d$, $D \neq \emptyset$, which contains all relevant geometrical objects, and study vector fields $V : \bar{D} \rightarrow \mathbb{R}^d$, satisfying the conditions

$$V \text{ is globally Lipschitz continuous in } \bar{D}, \text{ and } \text{supp}(V) \subset D. \quad (8)$$

Note that in [42, 18] a more general setting is considered, but this suffices for our purposes. If the velocity field V satisfies condition (8) we associate it with the so called flow map $T_t(V) : d/dt(T_t(V)(x)) = V(x)$. It transforms a set $\Omega \subset \bar{D}$ into a new domain

$$\Omega_t(V) := T_t(V)(\Omega) = \{T_t(V)(x) \mid \forall x \in \Omega\},$$

which is also contained in \bar{D} . If it is clear which velocity field drives the transformation we will drop it to shorten notation. Furthermore we introduce the spaces of k -times differentiable functions with compact support in D , i.e.

$$\mathcal{D}^k(D) := \{\theta \in \mathbf{C}^k(D) \mid \text{supp } \theta \subset\subset D\}.$$

The result [18, Theorem 4.5.1] shows that the transformation T_t is bi-Lipschitz for some $\tau > 0$ and $t \in [0, \tau]$ if V satisfies (8). If $V \in \mathcal{D}^k(D)$ then T_t is a C^k -diffeomorphism.

Let us recall the classical notions of the Eulerian semiderivative and shape differentiability:

Definition 1. [42, Definition 3.1, Definition 3.2] Let $\mathcal{A} \subset \mathcal{P}(D)$ be an appropriate set of shape variables contained in the hold-all domain $D \subset \mathbb{R}^d$.

1. The *Eulerian semiderivative* of $j : \mathcal{A} \rightarrow \mathbb{R}$ at $\Omega \in \mathcal{A}$ in a direction V satisfying (8) is given by

$$dj(\Omega; V) := \lim_{t \searrow 0} \frac{j(\Omega_t(V)) - j(\Omega)}{t},$$

if the limit exists and is finite.

2. The shape functional $j : \mathcal{A} \rightarrow \mathbb{R}$ is said to be *shape differentiable* at $\Omega \in \mathcal{A}$, if the Eulerian semiderivative exists for all $V \in \mathcal{D}^\infty(D)$ and the map

$$dj(\Omega; \cdot) : \mathcal{D}^\infty(D) \rightarrow \mathbb{R}, \quad V \mapsto dj(\Omega; V),$$

is linear and continuous.

Remark 2. 1. The Hadamard-Zolesio structure theorem (c.f. [42, Theorem 3.2]) shows that the support of the vector distribution $dj(\Omega; \cdot) \in \mathcal{D}^\infty(D)^*$ is contained in $\partial\Omega \cap D$.

2. The proofs of most formulas for the computation of shape derivatives rely on the representation of the shape functional j in terms of the transformation $T_t(V)$. If j depends on the integral over a d -dimensional subset of Ω_t and a shape depended state y the directional derivative has a natural representation as a volume integral. If the boundary $\partial\Omega$ is smooth enough this can be related via Gauß' divergence theorem to a boundary representation in accordance with the structure predicted by the Hadamard-Zolesio theorem.
3. If the boundary $\partial\Omega$ of $\Omega \subset\subset D$ is compact then $dj(\Omega; \cdot)$ is continuous for the $\mathcal{D}^k(D)$ -topology for some $k \geq 0$ and $dj(\Omega; \cdot) \in \mathbf{H}^{-s}(D)$ for some $s \geq 0$ (c.f. [18, Remark 9.3.1]).
4. The derivative $dj(\Omega; \cdot) \in \mathcal{D}^\infty(D)^*$ is often called the shape gradient. We think that the terms derivative and gradient should be clearly separated. Whereas the derivative is an element of the dual space of some vector space, in our terminology the gradient is the Riesz representative of the derivative with respect to some scalar product. So if $dj(\Omega; \cdot) \in \mathbf{H}^*$ for some Hilbert space \mathbf{H} the gradient $\nabla j(\Omega)$ with respect to the \mathbf{H} -scalar-product is given by

$$(\nabla j(\Omega), V)_{\mathbf{H}} = dj(\Omega; V), \quad \forall V \in \mathbf{H}.$$

We now want to compute the shape derivative of $j(\Omega) = J(\Omega, y(\Omega))$ in a direction V . This is not trivial since the states $y(\Omega_t) \in H_{\mathbb{R}}^1(\Omega_t)$ depend on the varying domains and are defined in function spaces which also depend on t . The standard approach to this problem is to introduce a function space parametrization. If there exists a $\tau > 0$ such that for all $t \in [0, \tau]$ the condition

$$T_t(V), T_t(V)^{-1} \in \mathbf{W}^{1,\infty}(D) \quad (9)$$

holds, then $\forall t \in [0, \tau]: \varphi \circ T_t(V) \in H_{\mathbb{R}}^1(\Omega)$ for all $\varphi \in H_{\mathbb{R}}^1(T_t(V)(\Omega))$, and $\psi \circ T_t(V)^{-1} \in H_{\mathbb{R}}^1(T_t(V)(\Omega))$ for all $\psi \in H_{\mathbb{R}}^1(\Omega)$. We will now use this to transform the state equation given on $\Omega_t = T_t(\Omega)$ onto the current domain Ω . This operation is also called pull-back.

Let us consider the solution y_t of the state equation (6) on the varying domains $T_t(\Omega)$ for $t \in [0, \tau]$. We transform the left-hand side

$$b_t(y_t, \varphi_t) := (\nabla y_t, \nabla \varphi_t)_{L_{\mathbb{R}}^2(T_t(\Omega))} - k^2(y_t, \varphi_t)_{L_{\mathbb{R}}^2(T_t(\Omega))} - (iky_t, \varphi_t)_{L_{\mathbb{R}}^2(\Gamma_a)}$$

onto the reference domain via

$$b(t, y, \varphi) := (A(t)\nabla y, \nabla \varphi)_{L_{\mathbb{R}}^2(\Omega)} - k^2(\det(DT_t)y, \varphi)_{L_{\mathbb{R}}^2(\Omega)} - (iky, \varphi)_{L_{\mathbb{R}}^2(\Gamma_a)},$$

where we have used the notation

$$\begin{aligned} DT_t &:= \text{Jacobian matrix of } T_t, \\ A(t) &:= \det(DT_t)DT_t^{-1}DT_t^{-T}, \end{aligned}$$

and made use of the fact that $\det(DT_t) = |\det(DT_t)|$ for $t \geq 0$ small and $T_t(\Gamma_a) = \Gamma_a$, $\forall t \geq 0$. The boundary conditions are defined on the fixed boundary Γ_a and do not change through the transformation

$$f(t, \varphi) = f(0, \varphi) = \left(\frac{\partial z}{\partial n} - ikz, \varphi \right)_{L_{\mathbb{R}}^2(\Gamma_a)}.$$

If (9) is satisfied the state $y^t := y_t \circ T_t \in H_{\mathbb{R}}^1(\Omega)$ solves the transformed state equation

$$b(t, y^t, \varphi) = f(t, \varphi), \quad \forall \varphi \in H_{\mathbb{R}}^1(\Omega). \quad (10)$$

If we want to differentiate the reduced objective

$$j(t) := J(t, y^t) = \frac{1}{2} \|y^t\|_{L_{\mathbb{R}}^2(\Omega)}^2$$

at $t = 0$ it is advantageous to use an adjoint approach. Introducing the Lagrange functional

$$\mathcal{L}(t, y, p) := J(t, y) + b(t, y, p) - f(t, p),$$

we note that for a solution $y^t \in H_{\mathbb{R}}^1(\Omega)$ of the state equation it holds that

$$j(t) = \mathcal{L}(t, y^t, p), \quad \forall p \in H_{\mathbb{R}}^1(\Omega).$$

If further $p^t \in H_{\mathbb{R}}^1(\Omega)$ solves the adjoint equation

$$\left\langle \frac{\partial \mathcal{L}}{\partial y}(t, y^t, p^t), \psi \right\rangle_{H_{\mathbb{R}}^1(\Omega)^*, H_{\mathbb{R}}^1(\Omega)} = (y^t, \psi)_{L_{\mathbb{R}}^2(\Omega)} + b(t, \psi, p^t) = 0, \quad \forall \psi \in H_{\mathbb{R}}^1(\Omega), \quad (11)$$

then the derivative of the reduced objective at $t = 0$ in the direction V is given as

$$\begin{aligned} dj(0; V) &= \frac{\partial J}{\partial t}(0, y^0) + \frac{\partial b}{\partial t}(0, y^0, p^0) - \frac{\partial f}{\partial t}f(0, p^0) \\ &= (A'(0)\nabla y^0, \nabla p^0)_{L_{\mathbb{R}}^2(\Omega)} - k^2(\text{div}(V)y^0, p^0)_{L_{\mathbb{R}}^2(\Omega)}. \end{aligned}$$

Here we denote the identity matrix with Id and

$$A'(0) := \text{div}(V)\text{Id} - DV - DV^T.$$

It is easy to check that $dj(0; \cdot) \in \mathbf{W}^{1,\infty}(D)^*$ and hence $dj(0; \cdot) \in \mathbf{H}^s(D)^*$ for $s > 2$.

If we add a volume functional to the objective e.g. $R(t) = \alpha \int_{D \setminus \Omega_t} 1 \, dx$, which penalizes the volume of the breakwater (and island), we can compute the derivative of the penalty term as

$$dR(0; V) = -\alpha \int_{\Omega} \text{div}(V) \, dx.$$

5. Geometric constraints

In this section we will describe how to incorporate geometric constraints in the shape optimization problem. Geometric constraints appear frequently in applications. In our breakwater optimization example we have the following: The harbor basin Q and the harbor approach A should always be water regions, i.e. be a part of Ω . On the other hand the island L should never be a part of Ω . We choose to implement an algorithm which always stays feasible with regard to those constraints. Let us denote the current iterate Ω_k . When employing a line-search based gradient descent method there are two possibilities to ensure feasibility of the iterates. After computing the negative gradient $-\nabla j(\Omega_k)$ one either projects before or after taking a step with length Δt . To be more precise the latter approach would use $\tilde{P}(T_{\Delta t}(-\nabla j(\Omega_k))(\Omega_k))$, where \tilde{P} denotes a suitable projection onto the set of admissible domains. The label “projected gradient method” is usually associated with this strategy. Since there is no canonical methodology to impose a metric on the space of domains and thus define and implement such a projection we decided to use the alternative approach. We project the gradient onto a suitable set of velocity fields which keep the deformed domain in the admissible set. The easiest way to ensure this is to define

$$\mathcal{V}_{feas} := \{V : D \rightarrow \mathbb{R}^d \mid V|_{\mathcal{F}} = 0, \mathcal{F} = Q \cup A \cup L \cup \partial D\},$$

and to consider the candidates $T_{\Delta t}(P(-\nabla j(\Omega_k)))(\Omega_k)$, where P is the $\mathbf{H}^s(D)$ projection onto $\mathcal{V}_{ad} := \mathbf{H}^s(D) \cap \mathcal{V}_{feas}$. Since \mathcal{V}_{ad} is a linear subspace the projection can be implemented efficiently. In fact, instead of first computing the gradient and then projecting it we can directly compute the gradient with respect to the subspace and obtain the same velocity field.

Theorem 3. *The element $V \in \mathcal{V}_{ad}$ satisfies*

$$(V, W)_{\mathbf{H}^s(D)} = dj(0; W), \quad \forall W \in \mathcal{V}_{ad}, \quad (12)$$

if and only if there holds $V = P(\tilde{V})$, where $P(\cdot)$ denotes the projection onto \mathcal{V}_{ad} with regard to the $\mathbf{H}^s(D)$ -scalar-product and $\tilde{V} \in \mathbf{H}^s$ satisfies

$$(\tilde{V}, \tilde{W})_{\mathbf{H}^s(D)} = dj(0; \tilde{W}), \quad \forall \tilde{W} \in \mathbf{H}^s(D). \quad (13)$$

PROOF. Let $\tilde{V} \in \mathbf{H}^s(D)$ satisfy (13). Note that the projection $P(\tilde{V})$ onto \mathcal{V}_{ad} is uniquely determined by the equality

$$(\tilde{V} - P(\tilde{V}), W)_{\mathbf{H}^s(D)} = 0, \quad \forall W \in \mathcal{V}_{ad}.$$

Hence we see that for $W \in \mathcal{V}_{ad}$

$$(\tilde{V} - V, W)_{\mathbf{H}^s(D)} = (\tilde{V}, W)_{\mathbf{H}^s(D)} - (V, W)_{\mathbf{H}^s(D)} = dj(0; W) - (V, W)_{\mathbf{H}^s(D)}.$$

This shows the equivalence of the statements.

To summarize projecting the gradient is an efficient strategy to strictly enforce the geometric constraints. On the other hand the set of admissible velocity fields is severely restricted, which might hinder the optimization process. In the numerical experiments we will also consider other choices for the projection.

6. Level set method

The level set method was introduced in [31] and is used widely to describe propagating fronts, moving interfaces, image segmentation, morphing bodies and many similar applications. We refer to the monographs [32, 34] for a detailed presentation of this rich topic. The core idea is to describe $\Omega \subset \mathbb{R}^d$ as the subzero level set

$$\Omega := \{x \in \mathbb{R}^d \mid \Phi(x) < 0\}$$

of some continuous function $\Phi : \mathbb{R}^d \rightarrow \mathbb{R}$. Denote $\Gamma = \{x \in \mathbb{R}^d \mid \Phi(x) = 0\}$ and assume $\Gamma = \partial\Omega$. For $\Phi \in C^k(\mathbb{R}^d, \mathbb{R})$, $k \geq 1$, $\Gamma \neq \emptyset$ and assuming $\nabla\Phi(x) \neq 0 \forall x \in \Gamma$, it holds that Γ is a C^k -submanifold of dimension $d - 1$ in \mathbb{R}^d .

We now consider a family of domains $\Omega_t = T_t(\Omega)$ and their level set functions $\Phi(t, \cdot)$. Let $x(t)$ be the solution of

$$\frac{dx}{dt}(t) = V(x(t)), \quad x(0) = x_0.$$

If we differentiate the equation $\Phi(t, x(t)) = d$ for any level d close to 0 with respect to t we obtain

$$\frac{\partial\Phi}{\partial t}(t, x(t)) + \nabla_x\Phi(t, x(t))^T V(x(t)) = 0.$$

This motivates us to use the Hamilton-Jacobi equation

$$\frac{\partial\Phi}{\partial t} + \nabla\Phi^T V = 0 \tag{14}$$

to propagate the subzero level set of the current domain. If the velocity field along which we want to transform our domain is given by the projected negative gradient $V = -P(\nabla j(\Omega))$ we can use (14) to obtain a representation of the next iterate level set function. Is it well-known that the Hamilton-Jacobi equation (14) has a unique viscosity solution under suitable assumptions [33, 43].

7. Optimization and discretization aspects

Let us briefly sketch the algorithm before going into details. Starting with a level-set function Φ given on a regular grid we extract a discretization of the domain Ω and the interface Γ_I . We solve the state and adjoint equations on the discretized domain using piecewise linear finite elements. Now we can compute the shape derivative of the reduced objective and obtain a projected gradient representation from (12). Finally the level set function is evolved according to (14) with $V = -P(\nabla j(\Omega))$ for some time span Δt which we choose such that it satisfies the Armijo condition.

We approximate the interface Γ_I , given by the zero level-set of Φ , by one or multiple polygonal curves. Between each pair of neighbouring points on the regular grid at which the level set function has different signs there is an intersection point of the zero level set with the edges of the grid. We approximate this intersection point using an affine model for Φ along the edge. Connecting all these intersection points, we obtain the polygonal approximation to Γ_I .

In the next step, the domains Ω and D are discretized with triangular meshes which resolve the polygonal boundary $\Gamma_{I,h}$. Furthermore the mesh representing Ω consists of a subset of the triangles of the mesh for D . Cells of the rectangular grid for which all four vertices have the same sign of Φ are split along their diagonal into two triangles, and cells which are intersected by Γ_I are split depending on how they are intersected.

The mesh on Ω is used to solve the state and adjoint equations, and the mesh on D is used to solve (12) for the shape gradient. Our mesh construction ensures that each point of the original regular grid is also a vertex of the triangle mesh, so that we can extract V on each grid point to solve (14).

Let us briefly comment on the discretization of the state and adjoint equation. We use the usual machinery of piecewise linear continuous finite elements to discretize the scalar products $(f, g)_{L^2(\Omega)}$, $(\nabla f, \nabla g)_{L^2(\Omega)}$, and $(f, g)_{L^2(\Gamma_a)}$ for some real valued functions f, g . Recall the notation from section 2. Splitting every complex valued function f into $f = f_1 + if_2$, with $f_i : \Omega \rightarrow \mathbb{R}$, we find

$$(f, g)_{L^2_{\mathbb{R}}(\Omega)} = \text{Re}(f_1 + if_2, g_1 + ig_2)_{L^2_{\mathbb{C}}(\Omega)} = (f_1, g_1)_{L^2(\Omega)} + (f_2, g_2)_{L^2(\Omega)}.$$

Analogously it holds that

$$(\nabla f, \nabla g)_{L^2_{\mathbb{R}}(\Omega)} = (\nabla f_1, \nabla g_1)_{L^2(\Omega)} + (\nabla f_2, \nabla g_2)_{L^2(\Omega)},$$

and

$$(if, g)_{L^2_{\mathbb{R}}(\Gamma_a)} = (f_1, g_2)_{L^2(\Gamma_a)} - (f_2, g_1)_{L^2(\Gamma_a)}. \quad (15)$$

Note that $z(x) = e^{ikd^T x}$ which implies $\frac{\partial z}{\partial n} - ikz = ik(d^T n - 1)z$. Hence we only need to implement the boundary expression (15). Combining these formulas with the usual mass and stiffness matrices we can easily assemble the system matrix and right-hand side of the state equation (10) and adjoint equation (11).

Note that, in order to guarantee the minimum regularity of $\mathbf{W}^{1,\infty}(D)$ for the velocity field, we would need to compute the gradient in $\mathbf{H}^s(D)$ with $s > 2$. In our numerical experiments we neglect to do so. For ease of implementation we only use piecewise linear continuous finite elements to discretize the ansatz and test spaces of (12). We employ the triangulation of D as described above. We experiment with different choices of the scalar product used for the gradient computation, c.f. section 8. The derivative of the reduced objective in the direction V is given by

$$dj(0; V) = (A'(0)\nabla y, \nabla p)_{L^2_{\mathbb{R}}(\Omega)} - k^2(\operatorname{div}(V)y, p)_{L^2_{\mathbb{R}}(\Omega)}.$$

Using simple matrix manipulations we can rewrite this expression such that the discretized version of the directional derivative $dj_h(0)$ can be evaluated by a simple vector multiplication $dj_h(0)^T V$.

Once we have computed the projected gradient, we need to update our geometry. For this we solve

$$\frac{\partial \Phi^t}{\partial t} + V^T \nabla \Phi^t = 0 \quad \forall t \in (0, \Delta t), \quad \Phi^t(0) = \Phi,$$

with $V = -\nabla j(\Omega)$ and use $\Phi^t(\Delta t)$ as the new level set function in the next iteration. The time span Δt is determined by the Armijo rule, i.e.

$$j(\Delta t) \leq j(0) + \Delta t \gamma j'(0; V).$$

As was suggested in [37], we employ the Local Lax-Friedrichs flux (c.f. [44]) and an explicit Euler time stepping scheme to evolve Φ^t . In our setting this leads to the level set function updates

$$\Phi_{ij}^{k+1} = \Phi_{ij}^k - \delta t H^{LLF}. \quad (16)$$

Here Φ_{ij}^k is the value of the level set function in the node (x_i, y_j) of the regular grid at the k -th time step,

$$H^{LLF} = \frac{p^- + p^+}{2} V_1 + \frac{q^- + q^+}{2} V_2 - \frac{1}{2}(p^+ - p^-)|V_1| - \frac{1}{2}(q^+ - q^-)|V_2|,$$

and

$$\begin{aligned} p^- &= \frac{\Phi_{ij}^k - \Phi_{i-1,j}^k}{\Delta x}, & p^+ &= \frac{\Phi_{i+1,j}^k - \Phi_{ij}^k}{\Delta x}, \\ q^- &= \frac{\Phi_{ij}^k - \Phi_{i,j-1}^k}{\Delta y}, & q^+ &= \frac{\Phi_{i,j+1}^k - \Phi_{ij}^k}{\Delta y}. \end{aligned}$$

The time step size δt is chosen to satisfy the Courant-Friedrichs-Lewy condition. If many time steps are necessary it is common to reinitialize the level set function every few steps for numerical stability. We use the signed distance function of the current domain (as defined by Φ^k) which we obtain via a fast marching algorithm [45]. In our numerical experiments the reinitialization was only rarely necessary.

8. Numerical Experiments

We implemented the proposed projected gradient method in GNU Octave [46]. The routines for generating the geometry from the level set function and assembling the finite element mesh are using the Octave package `level-set` [47]. It also provides a method to compute the signed distance function using a fast marching algorithm.

In our numerical experiments the harbor basin corresponds to the box $|x| \leq 0.4$, $-0.3 \leq y \leq 0.0$. The harbor approach is given by $|x| \leq 0.1$, $0 \leq y$ and the isle by $|x| \leq 0.55$, $-0.6 \leq y \leq 0.4$. The computational effort is reduced by imposing that everything outside the box $-1 \leq x, y \leq 1$ is ocean. We study an incoming wave with wave number $k = 7$ and compare the effects of different scalar products for the computation of the projected gradient. We present results for the projected H^1 gradient, for the H^1 gradient which is projected with respect to the L^2 scalar product and an approximation of the projected gradient with respect to the scalar product induced by the bi-Laplacian operator.

For the first experiment the regular grid consists of 501×501 points and the incident wave direction is $d = (\sqrt{1/2}, -\sqrt{1/2})^T$. The initial geometry can be seen on the left side of Figure 2. Note the wave resonance in the harbor basin. In order to be able to compare the results, the L^2 norm of the projected gradient is used as performance criterion in all examples. Note that optimization and discretization do not commute in our approach. Furthermore we are only using a first-order method, hence convergence towards a critical point of the discrete problem can not be expected in a reasonable number of steps. We stop the algorithm either if the L^2 norm of the gradient is below $1.0e-05$ or after 3000 iterations. In general the performance of the different methods reflects the smoothness of the different gradients. Whereas the bi-Laplacian and H^1 gradients smoothly transform the initial geometry, the L^2 projected gradient leads to more drastic shape and topology changes at the beginning of the optimization. This is not only due to the decreased regularity. The L^2 projected gradient is not pulled down to zero in the vicinity of the forbidden region, thus allowing the breakwater to touch the forbidden region and to move away again in later iterations. If we introduce a rescaling of the gradient if it is larger than a given threshold the L^2 projected gradient method is well behaved and actually often leads to better results than its smoother counterparts. This might be explained by a greater flexibility of the gradient which allows more local changes.

A note regarding topology changes: Although the algorithm allows topology changes we do not compute a topological derivative. Hence, the sensitivity with regard to topology changes is not represented in the geometry update. This can lead to very small step sizes or even a stalling of the algorithm if this update leads to topology changes which increase the objective value.

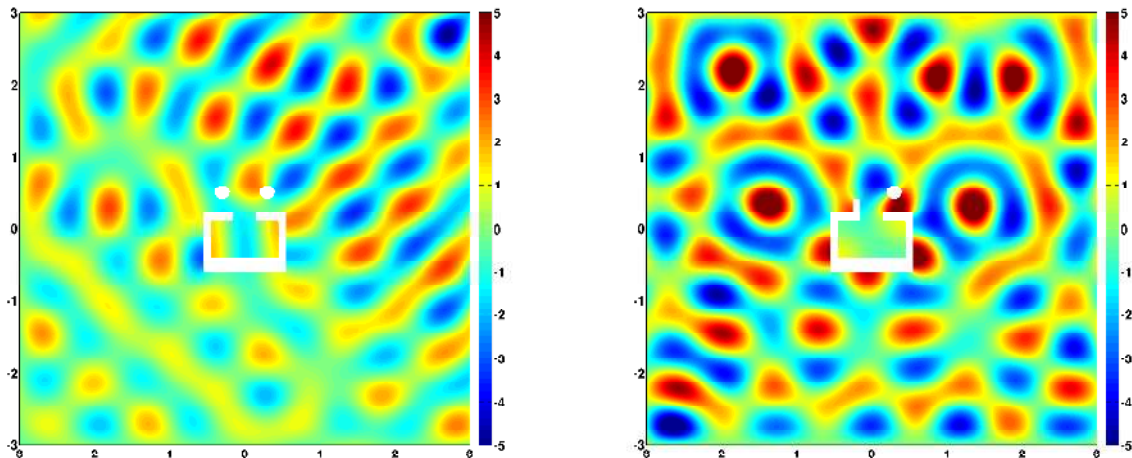


Figure 2: Wave pattern for initial geometry of the first (left) and second experiment (right)

The H^1 gradient method reaches in iteration 1312 an L^2 norm of the gradient below $1.0e-05$ and is terminated. The objective value of $3.8e-07$ is very close to the lower bound of 0. The bi-Laplacian gradient method gets stuck after 467 iterations. Whereas the L^2 norm of the gradient $6.5e-05$ is already pretty small the objective $2.6e-04$ is several orders of magnitude higher than the result of the H^1 gradient method. If we take a look at the convergence plots in Figure 3 we see that in the beginning the bi-Laplacian gradient method reduces the L^2 norm of the gradient much faster than the H^1 gradient method. Comparing the final geometries in figure 4 shows that the lower regularity of the H^1 gradient allows for more flexible changes which lead to a better objective value. The same behavior could be

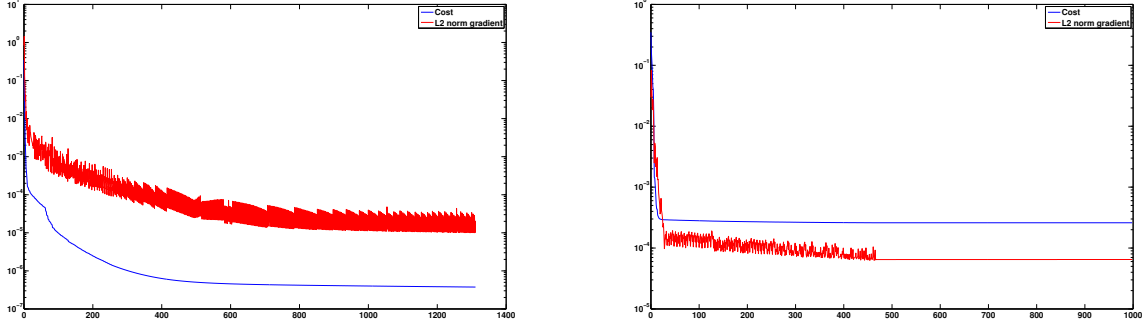


Figure 3: Experiment 1: Convergence of the H^1 projected gradient method (left) and the bi-Laplacian gradient method (right)

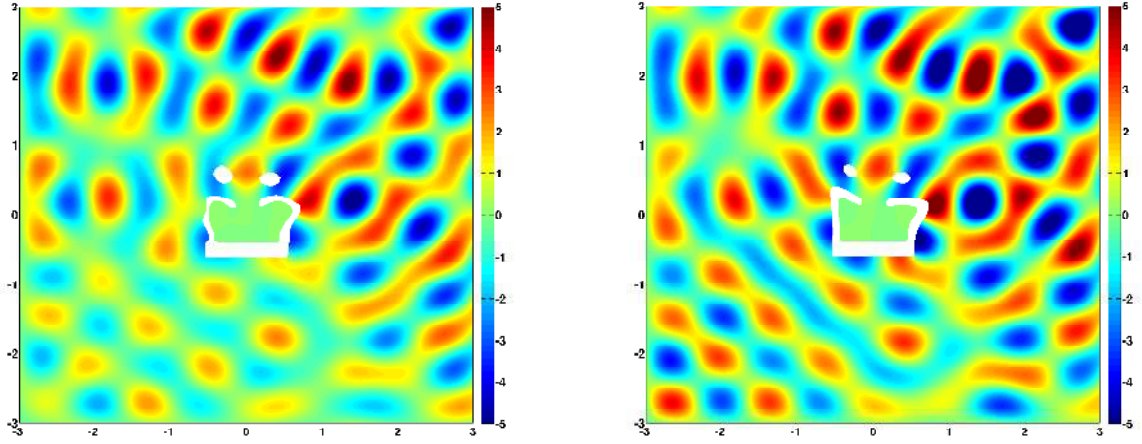


Figure 4: Experiment 1: Wave pattern for the optimized geometry using the H^1 projected gradient (left) and the bi-Laplacian gradient (right)

observed in other tests as well. The bi-Laplacian gradient method reduces the L^2 norm of the gradient quickly in few iterations but then gets stuck. If we project the H^1 gradient with respect to the L^2 scalar product (effectively simply cutting it off in the forbidden regions) we observe a heavily deformed breakwater with topology changes, c.f. Figure 5. But the convergence plot, c.f. Figure 6, shows that the descent is very slow, after the first iterations only very small step sizes are chosen. In iteration 3000 the objective value is $7.4e-03$ and the L^2 norm of the gradient is $3.3e-02$. If the mentioned rescaling of large gradients is introduced a much better performance can be observed. After 3000 iterations the objective value is $8.3e-06$ and the L^2 norm of the gradient is $2.5e-04$.

In the second experiment the regular grid consists of 601×601 points and the incident wave direction is $d = (0, -1)^T$. The initial geometry can be seen on the right side of Figure 2. After a good start the H^1 projected gradient method is slowly decreasing the objective and after 3000 iterations achieves an objective value of $1.7e-05$ and the L^2 norm of the gradient is $4.7e-4$. The bi-Laplacian gradient method gets stuck already after 7 iterations. The final geometries are shown in Figure 7. In this example the L^2 projected gradient performs the best. After 3000 iterations the rescaled method achieves an objective value of $1.5e-06$ and the L^2 norm of the gradient is $8.4e-05$. And without rescaling of the gradient it is even better. After choosing small step sizes in the first iterations the algorithm terminates in iteration 1337 with an L^2 norm of the gradient below $1.0e-05$ and an objective value of $6.3e-07$. The convergence plots are shown in Figure 8 and the optimized geometries in Figure 9.

In most of our experiments the volume of the breakwater did not vary much during the optimization. Still it might be interesting to add a volume constraint which might be realized by e.g. an Augmented Lagrange method.

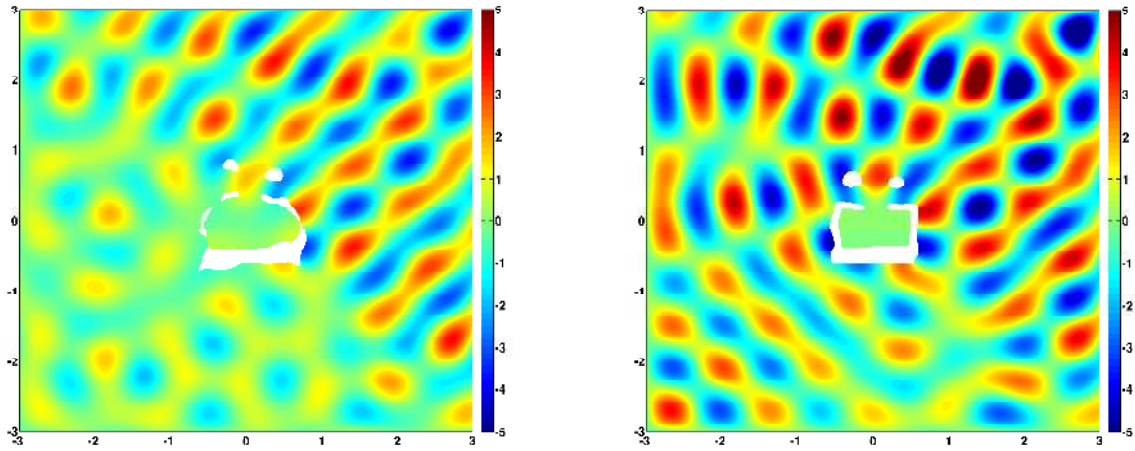


Figure 5: Experiment 1: Wave pattern for the optimized geometry using the L^2 projected gradient without rescaling (left) and with rescaling (right)

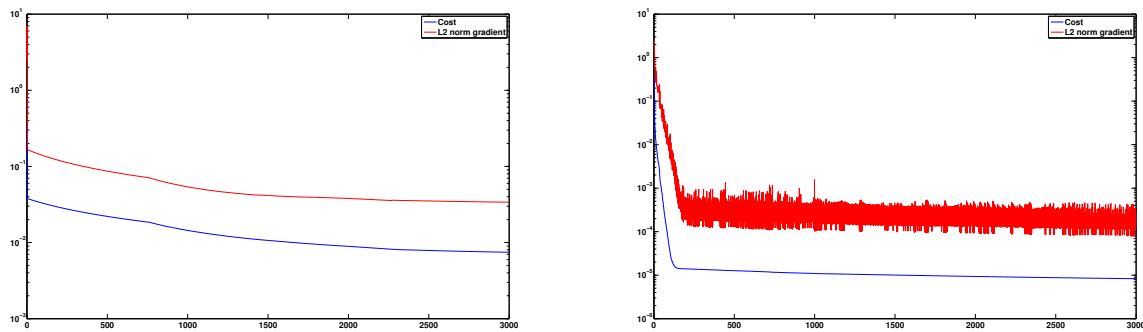


Figure 6: Experiment 1: Convergence of the L^2 projected gradient method without rescaling (left) and with rescaling (right)

9. Acknowledgement

We wish to thank Prof. Wolfgang Ring (University of Graz) for valuable discussions and a careful proofreading of this article. We gratefully acknowledge support from the International Research Training Group IGDK1754, funded by the German Science Foundation (DFG) and the Austrian Science Fund (FWF). The computations were carried out on a Linux cluster that was partially funded by the grant DFG INST 95/919-1 FUGG.

- [1] G. Allaire, F. Jouve, A.-M. Toader, A level-set method for shape optimization, *Comptes Rendus Mathématique* 334 (12) (2002) 1125 – 1130. doi:[http://dx.doi.org/10.1016/S1631-073X\(02\)02412-3](http://dx.doi.org/10.1016/S1631-073X(02)02412-3). URL <http://www.sciencedirect.com/science/article/pii/S1631073X02024123>
- [2] G. Allaire, F. Jouve, A.-M. Toader, Structural optimization using sensitivity analysis and a level-set method, *J. Comput. Phys.* 194 (1) (2004) 363–393. doi:10.1016/j.jcp.2003.09.032. URL <http://dx.doi.org/10.1016/j.jcp.2003.09.032>
- [3] S. J. Osher, F. Santosa, Level set methods for optimization problems involving geometry and constraints: I. frequencies of a two-density inhomogeneous drum, *Journal of Computational Physics* 171 (1) (2001) 272 – 288. doi:<http://dx.doi.org/10.1006/jcph.2001.6789>. URL <http://www.sciencedirect.com/science/article/pii/S0021999101967890>
- [4] J. Sethian, A. Wiegmann, Structural boundary design via level set and immersed interface methods, *Journal of Computational Physics* 163 (2) (2000) 489 – 528. doi:<http://dx.doi.org/10.1006/jcph.2000.6581>. URL <http://www.sciencedirect.com/science/article/pii/S0021999100965811>
- [5] M. Y. Wang, X. Wang, D. Guo, A level set method for structural topology optimization, *Computer Methods in Applied Mechanics and Engineering* 192 (1–2) (2003) 227 – 246. doi:[http://dx.doi.org/10.1016/S0045-7825\(02\)00559-5](http://dx.doi.org/10.1016/S0045-7825(02)00559-5). URL <http://www.sciencedirect.com/science/article/pii/S0045782502005595>

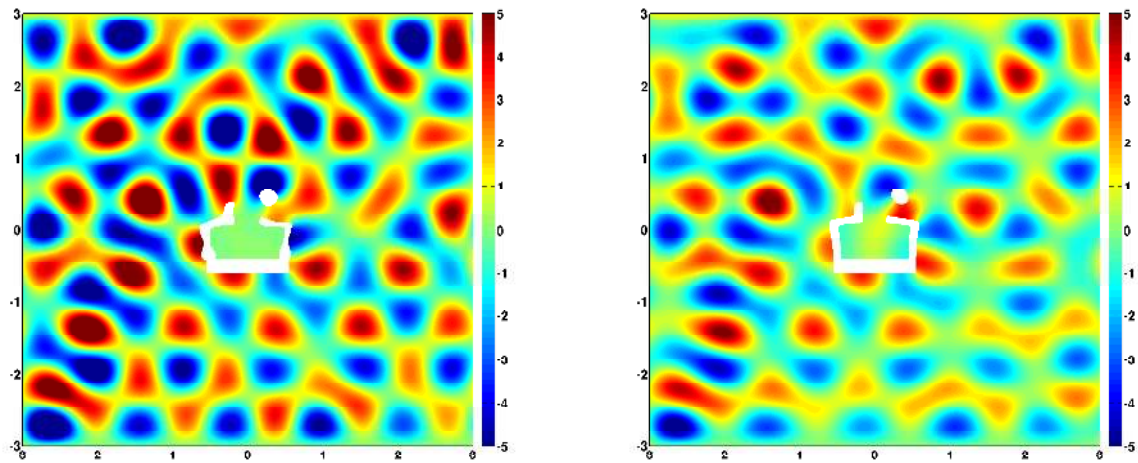


Figure 7: Experiment 2: Wave pattern for the final geometry using the H^1 projected gradient (left) and the bi-Laplacian gradient (right)

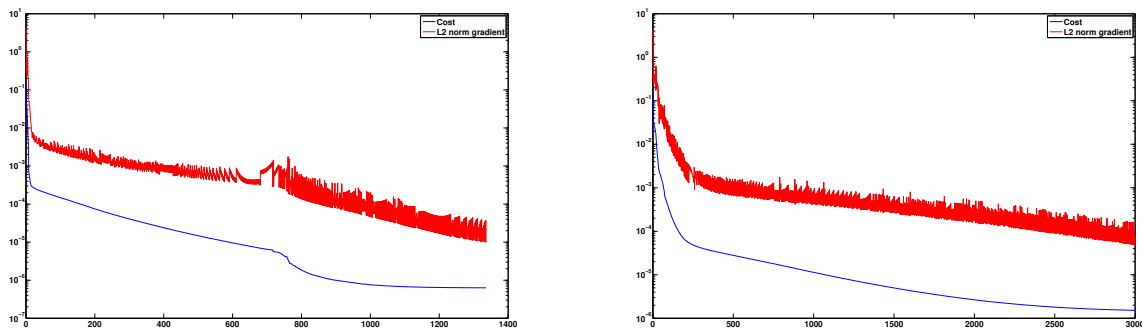


Figure 8: Experiment 2: Convergence of the L^2 projected gradient method without rescaling (left) and with rescaling (right)

- [6] G. Allaire, C. Dapogny, P. Frey, Topology and geometry optimization of elastic structures by exact deformation of simplicial mesh, *Comptes Rendus Mathématique* 349 (17–18) (2011) 999 – 1003. doi:<http://dx.doi.org/10.1016/j.crma.2011.08.012>.
URL <http://www.sciencedirect.com/science/article/pii/S1631073X1100241X>
- [7] G. Allaire, C. Dapogny, P. Frey, A mesh evolution algorithm based on the level set method for geometry and topology optimization, *Structural and Multidisciplinary Optimization* 48 (4) (2013) 711–715.
URL <http://dx.doi.org/10.1007/s00158-013-0929-2>
- [8] G. Allaire, C. Dapogny, P. Frey, Shape optimization with a level set based mesh evolution method, Tech. rep., Université Pierre et Marie Curie - Paris VII (2014).
- [9] S.-H. Ha, S. Cho, Level set based topological shape optimization of geometrically nonlinear structures using unstructured mesh, *Computers & Structures* 86 (13–14) (2008) 1447 – 1455, structural Optimization. doi:<http://dx.doi.org/10.1016/j.compstruc.2007.05.025>.
URL <http://www.sciencedirect.com/science/article/pii/S0045794907001757>
- [10] P.-O. Persson, Mesh generation for implicit geometries, Ph.D. thesis, Massachusetts Institute of Technology (2004).
- [11] Q. Xia, T. Shi, S. Liu, M. Y. Wang, A level set solution to the stress-based structural shape and topology optimization, *Computers & Structures* 90–91 (0) (2012) 55 – 64. doi:<http://dx.doi.org/10.1016/j.compstruc.2011.10.009>.
URL <http://www.sciencedirect.com/science/article/pii/S0045794911002562>
- [12] S. Yamasaki, T. Nomura, A. Kawamoto, K. Sato, S. Nishiwaki, A level set-based topology optimization method targeting metallic waveguide design problems, *International Journal for Numerical Methods in Engineering* 87 (9) (2011) 844–868. doi:10.1002/nme.3135.
URL <http://dx.doi.org/10.1002/nme.3135>
- [13] N. van Dijk, K. Maute, M. Langelaar, F. van Keulen, Level-set methods for structural topology optimization: a review, *Structural and Multidisciplinary Optimization* 48 (3) (2013) 437–472.
URL <http://dx.doi.org/10.1007/s00158-013-0912-y>

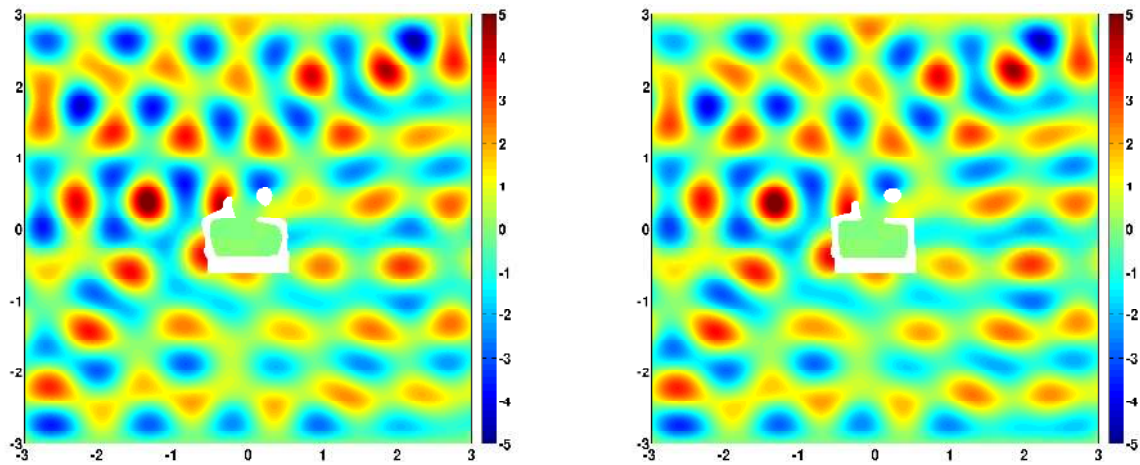


Figure 9: Experiment 2: Wave pattern for the optimized geometry using the L^2 projected gradient without rescaling (left) and with rescaling (right)

- [14] M. BURGER, S. J. OSHER, A survey on level set methods for inverse problems and optimal design, *European Journal of Applied Mathematics* null (2005) 263–301. doi:10.1017/S0956792505006182.
URL http://journals.cambridge.org/article_S0956792505006182
- [15] G. Allaire, *Shape Optimization by the Homogenization Method*, Vol. 146 of *Applied Mathematical Sciences*, Springer New York, 2002.
- [16] G. Allaire, *Conception optimale de structures*, Vol. 58 of *Mathématiques & Applications*, Springer Berlin Heidelberg, 2007.
- [17] M. Bendsoe, *Optimization of Structural Topology, Shape, and Material*, Springer Berlin Heidelberg, 1995.
- [18] M. C. Delfour, J.-P. Zolésio, *Shapes and Geometries*, 2nd Edition, Society for Industrial and Applied Mathematics, Philadelphia, 2011.
arXiv:<http://epubs.siam.org/doi/pdf/10.1137/1.9780898719826>, doi:10.1137/1.9780898719826.
URL <http://epubs.siam.org/doi/abs/10.1137/1.9780898719826>
- [19] J. Haslinger, R. A. E. Mäkinen, *Introduction to Shape Optimization*, Society for Industrial and Applied Mathematics, Philadelphia, 2003.
arXiv:<http://epubs.siam.org/doi/pdf/10.1137/1.9780898718690>, doi:10.1137/1.9780898718690.
URL <http://epubs.siam.org/doi/abs/10.1137/1.9780898718690>
- [20] A. Henrot, M. Pierre, *Variation et optimisation de formes. Une analyse géométrique*, Vol. 48 of *Mathématiques & Applications*, Springer, 2005.
- [21] B. Mohammadi, O. Pironneau, *Applied Shape Optimization for Fluids*, Oxford University Press, 2001.
- [22] O. Pironneau, *Optimal Shape Design For Elliptic Systems*, Springer Series in Computational Physics, Springer Berlin Heidelberg, 1984.
- [23] J. Sokolowski, J.-P. Zolésio, *Introduction to Shape Optimization*, Series in Computational Mathematic, Springer, 1992.
- [24] J. Fernandes, M. V. dos Santos, C. Fortes, An element-by-element mild-slope model for wave propagation studies, in: *ICS 2004 (Proceedings)*, Journal of Coastal Research, 2004, pp. 1869–1874.
- [25] J.-J. Lee, C.-P. Lai, Y. Li, Application of computer modeling for harbor resonance studies of long beach & los angeles harbor basins, *Coastal Engineering Proceedings* 1 (26).
- [26] J.-Y. Maa, H.-H. Hwung, A wave transformation model for harbor planning, in: *Proceedings, Waves '97*, Vol. 1, 1997, pp. 256–270.
- [27] X. Xing, *Computer modeling for wave oscillation problems in harbors and coastal regions*, Ph.D. thesis, University of Southern California (2009).
- [28] X. Xing, J.-J. Lee, F. Raichlen, Harbor resonance: a comparison of field measurements to numerical results, *Coastal Engineering Proceedings* 1 (32).
- [29] J. Berkhoff, Computation of combined refraction-diffraction, in: *Proceedings 13th Coastal Engineering Conference*, American Society of Civil Engineers, 1972, pp. 471–490.
- [30] F. J. Baron Lopez, *Quelque problèmes d'optimisation de formes en électromagnétisme et mécanique de fluides*, Ph.D. thesis, Paris VI, Grenoble, th. : analyse numérique (1998).
URL <http://opac.inria.fr/record=b1065779>
- [31] S. Osher, J. A. Sethian, Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulations, *Journal of Computational Physics* 79 (1988) 12–49.
- [32] R. Fedkiw, S. Osher, *Level Set Methods and Dynamic Implicit Surfaces*, Vol. 153 of *Applied Mathematical Sciences*, Springer New York, 2003.
- [33] Y. Giga, *Surface Evolution Equations: a level set approach*, Monographs in mathematics, Birkhäuser, 2006.
- [34] J. Sethian, *Level Set Methods and Fast Marching Methods : Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*, 2nd Edition, Cambridge University Press., 1999.
- [35] F. Ihlenburg, *Finite Element Analysis of Acoustic Scattering*, Springer New York, 1998.
- [36] R. Hiptmair, A. Paganini, S. Sargheini, Comparison of approximate shape gradients, *BIT Numerical Mathematics* (2014) 1–

27doi:10.1007/s10543-014-0515-z.

- [37] A. Laurain, K. Sturm, Domain expression of the shape derivative and application to electrical impedance tomography, Technical Report 1863, Weierstrass Institute for Applied Analysis and Stochastics (2013).
- [38] R. Leis, Initial Boundary Value Problems in Mathematical Physics, J. Wiley & Teubner Verlag, 1986.
- [39] C. Johnson, J. C. Nedelec, On the coupling of boundary integral and finite element methods, *Mathematics of Computation* 35 (152) (1980) 1063–1079.
URL <http://www.jstor.org/stable/2006375>
- [40] A. Rabinovich, Seiches and harbor oscillations, in: Y. Kim (Ed.), *Handbook of Coastal and Ocean Engineering*, World Scientific Publishing Company, 2009, pp. 93–236.
- [41] W. Hackbusch, *Elliptic differential equations: theory and numerical treatment*, Vol. 18, Springer series in computational mathematics, 1992.
- [42] M. Delfour, J. Zolésio, Structure of shape derivatives for nonsmooth domains, *Journal of Functional Analysis* 104 (1) (1992) 1 – 33.
doi:[http://dx.doi.org/10.1016/0022-1236\(92\)90087-Y](http://dx.doi.org/10.1016/0022-1236(92)90087-Y).
URL <http://www.sciencedirect.com/science/article/pii/002212369290087Y>
- [43] M. G. Crandall, I. Hitoshi, P. L. Lions, User’s guide to viscosity solutions of second order partial differential equations, *Bulletin of the American Mathematical Society* 27 (1) (1992) 1–67.
- [44] S. Osher, C. Shu, High-order essentially nonoscillatory schemes for hamilton–jacobi equations, *SIAM Journal on Numerical Analysis* 28 (4) (1991) 907–922. arXiv:<http://epubs.siam.org/doi/pdf/10.1137/0728049>, doi:10.1137/0728049.
URL <http://epubs.siam.org/doi/abs/10.1137/0728049>
- [45] J. A. Sethian, A fast marching level set method for monotonically advancing fronts, *Proceedings of the National Academy of Sciences* 93 (4) (1996) 1591–1595.
- [46] J. W. Eaton, D. Bateman, S. Hauberg, *GNU Octave version 3.0.1. manual: a high-level interactive language for numerical computations*, CreateSpace Independent Publishing Platform, 2009, ISBN 1441413006.
URL <http://www.gnu.org/software/octave/doc/interpreter>
- [47] D. Kraft, The ”level-set” package for gnu octave. octave forge, 2014.
URL <http://octave.sourceforge.net/level-set/>