

**Adaptive Cross Approximation of Tensors Arising
in the Discretization of Boundary Integral
Operator Shape Derivatives**

Mario Bebendorf, Sean Hardesty

no. 509

Diese Arbeit ist mit Unterstützung des von der Deutschen Forschungsgemeinschaft getragenen Sonderforschungsbereichs 611 an der Universität Bonn entstanden und als Manuskript vervielfältigt worden.

Bonn, Dezember 2011

Adaptive Cross Approximation of Tensors Arising in the Discretization of Boundary Integral Operator Shape Derivatives*

M. Bebendorf[†] S. Hardesty[‡]

December 5, 2011

The shape derivative of a dense $N \times N$ BEM matrix is a sparse three-way tensor with $O(N^2)$ non-zero entries, to which standard BEM acceleration techniques such as the Adaptive Cross Approximation (ACA) and FMM cannot be directly applied. The tensor can be used to compute shape sensitivities, or via adjoint equations, the gradient of an objective function. Although for many PDEs, calculation of the tensor can be avoided by expressing the shape derivative of the solution as the solution of a related PDE, this approach is not always easily amenable to BEM. Therefore, the computation of shape derivatives via the sparse three-way tensor is a valuable alternative, provided that efficient acceleration techniques exist. We propose a new algorithm for approximation of BEM shape derivative tensors based on 2d ACA that achieves the same complexity and error bounds as ACA for the BEM matrix itself. Numerical examples show that despite the much larger amount of data involved, the tensor approximation is only moderately slower than the matrix approximation. We also demonstrate the method on a shape optimization problem from the literature.

*This work has been supported in part by AFOSR grant FA9550-09-1-0225 and DFG Collaborative Research Center (SFB) 611.

[†]Institute for Numerical Simulation, University of Bonn, Wegelerstraße 6, 53115 Bonn, Germany. E-mail. bebendorf@ins.uni-bonn.de

[‡]Department of Computational and Applied Mathematics, MS-134, Rice University, 6100 Main Street, Houston, TX 77005-1892. E-mail. hardesty@rice.edu

1 Introduction

The original motivation for this work was to solve shape optimization problems in structural acoustics under the framework of [21]. These involve Neumann boundary conditions, and are amenable to modeling using BEM-FEM coupling [22, 13, 14]. More generally, shape optimization problems with PDE constraints expressed in the form of boundary integral equations (BIE) have been studied in [24, 23, 9, 10, 12]. For unbounded exterior domains, the BIE yield solutions exactly satisfying the Sommerfeld radiation condition and therefore avoid the need for artificial truncation of the domain. BIE also lend themselves to point-measurement of the external field via a representation formula. In addition, re-meshing during optimization can be avoided so long as the design changes are not too large. In the case of multiphysics coupling, or when the objective function requires fast access to the state over a bounded domain, BEM-FEM coupling can maintain the advantages of BEM for the exterior domain [11].

Discretization of BIE leads to dense BEM matrices, and to an $O(N^2)$ operation count. Computation of all matrix entries becomes impractically costly, even for moderately-sized problems. Many methods for acceleration of BEM are given in the literature, e.g., ACA [3], panel-clustering [20], FMM [16, 17, 8, 32], FFT-based methods [6], and wavelets [5]. Fundamentally, all are based on the same observation - that the interaction of well-separated portions of the surface is of low rank. In the context of PDE-constrained optimization algorithms, it is essential to be able to accelerate computation of objective function derivatives in a similar manner.

In §2.1, we will consider as a model problem the Laplace equation on an unbounded exterior domain. It is well-known [30, Ch. 3] that the shape derivative of the state can be expressed as the solution of a related boundary value problem, with boundary data involving the state and its derivatives: in the best case, one need only know the state or its normal derivative on the boundary. Since these quantities are already known, or can be accurately computed using BEM, shape optimization problems of this type can be solved using BEM and standard acceleration techniques such as ACA. This approach is followed in [24, 9, 11]. Things become more involved when the original boundary value problem is of Neumann type, or when second derivatives of the objective function are required: then, the extraction technique of Schwab and Wendland [29] can be used to compute tangential derivatives of the state on the boundary. In general, this involves the solution of additional integral equations, but allows acceleration of BEM via standard techniques. This approach is followed in [12], and is detailed further in §2.2.

An alternative approach to the adjoint calculus, detailed in §2.3, is to calculate the shape derivative of the boundary integral operator directly, and then apply it to the state and to the adjoint state in order to compute a gradient. If under discretization, the boundary integral operator becomes an $N \times N$ BEM matrix, then its shape derivative becomes a sparse three-way tensor of size $N \times N \times O(N)$. This approach avoids some of the complexity of the extraction technique, and does not require the solution of additional integral equations, but is of no practical utility if one must compute all $O(N^2)$ non-zero entries of the tensor. In §3, we present a new algorithm for approximation of these tensors based on 2d ACA. Finally, we demonstrate the approach in §4 via a numerical

example.

2 Operator Shape Derivatives

Consider the formal operator equation representation

$$\mathcal{A}(\phi)u = r(\phi)$$

of a PDE with solution $u[\phi]$, where ϕ specifies the shape of the domain boundary. The shape sensitivity $Du[\phi]$ satisfies

$$Du[\phi] = \mathcal{A}^{-1}(\phi)(Dr(\phi) - D\mathcal{A}(\phi)u[\phi]).$$

The sensitivity could itself be the quantity of interest, or the derivative of the mapping $\phi \mapsto J(u[\phi])$ to an objective function J could be computed via adjoint equations. For many problems (e.g., Laplace, Helmholtz), the sensitivity $Du[\phi]$ can be characterized as the solution of a related PDE, in which case discretization of the operator derivative $D\mathcal{A}(\phi)$ is not essential. However, when such a characterization is not known, or when it requires boundary data that is not easily computed using BEM, discretization of $D\mathcal{A}(\phi)$ is advantageous.

Let $A \in \mathbb{R}^{N \times N}$ be the matrix resulting from a discretization of \mathcal{A} , and $\nabla A \in \mathbb{R}^{N \times N \times P}$ be the three-way tensor of derivatives of all matrix entries with respect to some discrete shape parameter set of dimension P . Assuming that each of the P shape parameters is associated with a local variation in ϕ , i.e., that the support of the variation intersects the support of only a constant-order number of the N basis functions, then the complexity of computing ∇A is the same as that of computing A itself. If \mathcal{A} is discretized using FEM so that A is sparse, computation of ∇A presents no real problem. But if \mathcal{A} is discretized using BEM, then the resulting matrix A is dense, and its calculation must be accelerated using an approximation algorithm such as ACA or FMM. In this case, computation of ∇A must likewise be accelerated. However, standard matrix methods cannot be applied to the tensor ∇A . While kernel-independent FMM [32] can handle generic vector-valued kernels, it assumes the same size and structure of the kernel throughout the computation. We have therefore developed an approximation method for the tensor based on knowledge of its sparsity structure, and on the purely algebraic method of 2d ACA.

2.1 Adjoint Equation Formulation

In this section, we will present the model problem, and standard PDE characterizations of the shape derivative of the state u . In §2.2 and §2.3, we will contrast two different formulations of the adjoint calculus, and discuss their implications for numerical solution methods.

Let Ω^- be a bounded, simply-connected domain with boundary Γ , and complement $\Omega^+ = \mathbb{R}^3 \setminus \overline{\Omega^-}$. Consider the exterior Neumann problem for Laplace's equation: find

$u \in H_{\text{loc}}^1(\Omega^+)$ s.t.

$$-\Delta u = 0 \quad \text{in } \Omega^+, \quad (1a)$$

$$\frac{\partial u}{\partial n} = \psi \quad \text{on } \Gamma, \quad (1b)$$

$$u(x) = O(1/|x|) \quad \text{as } |x| \rightarrow \infty. \quad (1c)$$

The radiation condition (1c) ensures unique solvability: otherwise, u is only determined up to a constant. We define the boundary integral operators $\mathcal{K} : H^{1/2}(\Gamma) \rightarrow H^{1/2}(\Gamma)$, and $\mathcal{V} : H^{-1/2}(\Gamma) \rightarrow H^{1/2}(\Gamma)$ via

$$(\mathcal{K}\varphi)(x) = \int_{\Gamma} \partial_{n_y} S(x, y) \varphi(y) \, ds_y, \quad (\mathcal{V}\psi)(x) = \int_{\Gamma} S(x, y) \psi(y) \, ds_y,$$

where $S(x, y) = (4\pi|x - y|)^{-1}$ is the singularity function for the Laplace equation. Then, the integral equation

$$\mathcal{A}\varphi := \left(\frac{1}{2}\mathcal{I} - \mathcal{K}\right)\varphi = -\mathcal{V}\psi \quad (2)$$

can be solved for the unknown Dirichlet data $\varphi = u|_{\Gamma}$. The state u can then be expressed via the representation formula

$$u(x) = \int_{\Gamma} \frac{\partial S}{\partial n_y}(x, y) \varphi(y) - S(x, y) \psi(y) \, ds_y; \quad (3)$$

see [28, 29] for further details.

This problem can be formulated with Γ a Lipschitz surface, but in order to use Potthast's shape calculus [25, 26] for the boundary integral operators \mathcal{V}, \mathcal{K} , we will assume that Γ is of class C^2 . Consider a perturbed boundary shape Γ_{ϕ} , defined by

$$\Gamma \ni x \mapsto x + \phi(x).$$

Solving (1) on Γ_{ϕ} gives the state $u[\phi]$. Under the assumption that

$$\phi \in \mathcal{C} := \{\eta \in [C^2(\Gamma)]^3 : \|\eta\|_{[C^2(\Gamma)]^3} \leq l\},$$

for some fixed $l > 0$, Potthast [26] showed (see also [30, §3.2]) that the Fréchet derivative $v = Du[\phi = 0]\delta\phi$ (in the direction $\delta\phi$) solves

$$-\Delta v = 0 \quad \text{in } \Omega^+, \quad (4a)$$

$$\frac{\partial v}{\partial n} = - \left(\frac{\partial n}{\partial \phi} \delta\phi \right) \cdot \nabla u - \delta\phi \cdot \nabla^2 u \cdot n \quad \text{on } \Gamma, \quad (4b)$$

$$v(x) = O(1/|x|) \quad \text{as } |x| \rightarrow \infty. \quad (4c)$$

The result (4) thus characterizes the Fréchet derivative of the state u , the solution to (1), as the solution to another Laplace problem with Neumann boundary data. We have access to the given Neumann data $\psi = \frac{\partial u}{\partial n}|_{\Gamma}$, and to the Dirichlet data $\varphi = u|_{\Gamma}$ computed

via solution of (2), but (4b) requires $\nabla u|_\Gamma$ and $\nabla^2 u|_\Gamma$. The condition (4b) can also be expressed in a weak form using [30, (3.12)]. This avoids the need to evaluate $\nabla^2 u|_\Gamma$, but the surface gradient $\nabla_\Gamma u$ is still needed, and can be computed using the extraction approach of Schwab and Wendland [29]. Thus, for first-order shape calculus associated with the Neumann problem (4), only first-order derivatives of u on Γ are needed; this approach is detailed in §2.2. The extraction approach can also be used to compute second-order derivatives of the state on the boundary required for second-order shape calculus, as is done in [12] for a shape optimization problem on a two-dimensional domain (i.e., with a one-dimensional boundary).

The situation is significantly different from that of a Dirichlet model problem:

$$\begin{aligned} -\Delta u &= 0 && \text{in } \Omega^+, \\ u &= \varphi && \text{on } \Gamma, \\ u(x) &= O(1/|x|) && \text{as } |x| \rightarrow \infty. \end{aligned}$$

In this case, $v = Du[\phi = 0]\delta\phi$ solves (see [25] or [30, §3.1])

$$-\Delta v = 0 \quad \text{in } \Omega^+, \quad (5a)$$

$$v = -(\delta\phi \cdot n) \frac{\partial u}{\partial n} \quad \text{on } \Gamma, \quad (5b)$$

$$v(x) = O(1/|x|) \quad \text{as } |x| \rightarrow \infty. \quad (5c)$$

Here, the situation is much nicer. Once again, the Fréchet derivative can be characterized as the solution of a related (Dirichlet) Laplace problem, but this time, the boundary data depends on $\frac{\partial u}{\partial n}$, the Neumann data for the state, which would already be known as the solution of an integral equation related to (2). Thus, adjoint equations can easily be formulated using (5). This approach is used in [24, 9, 11].

Returning to the Neumann problem (1), we will consider minimization of an objective function J by suitable choice of ϕ in some admissible set Φ :

$$\min_{\phi \in \Phi} J(\varphi[\phi], \psi[\phi]; \phi), \quad (6)$$

with the Neumann data $\psi[\phi]$ given, and the unknown Dirichlet data $\varphi[\phi]$ determined as the solution to (2). Through the representation formula (3), this allows generic dependence of the objective function on u , although in practice, since we intend to use BEM, the objective function should depend only on the boundary data, or on u evaluated at a small number of points in Ω^+ .

We will not discuss existence of solutions to (6), as it depends on assumptions on J and on Φ . Rather, our focus is on evaluation of the objective function derivative

$$D_\phi J(\varphi[\phi], \psi[\phi]; \phi) + D_\varphi J(\varphi[\phi], \psi[\phi]; \phi) D\varphi[\phi] + D_\psi J(\varphi[\phi], \psi[\phi]; \phi) D\psi[\phi].$$

We will examine two different ways to evaluate the derivative

$$\langle D_\varphi J(\varphi[\phi], \psi[\phi]; \phi) D\varphi[\phi], \delta\phi \rangle_{\Phi^* \times \Phi} \quad (7)$$

in the direction $\delta\phi$:

1. Extraction: using the solution $v = Du[\phi = 0]\delta\phi$ to (4), in §2.2.
2. Using standard adjoint calculus based on the integral equation (2), in §2.3.

2.2 Extraction Approach

In this section, we describe the formulation of adjoint equations for (7) via the extraction technique of Schwab and Wendland [29].

Solution of (4) via (2) allows the calculation of the shape derivative

$$\widehat{\varphi} := v|_{\Gamma_\phi} = D\varphi[\phi]\delta\phi$$

via

$$\mathcal{A}\widehat{\varphi} = -\mathcal{V}\widehat{\psi}, \quad \widehat{\psi} := -\frac{\partial n}{\partial\phi}\delta\phi \cdot \nabla u - \delta\phi \cdot \nabla^2 u \cdot n.$$

Thus, the objective function derivative can be computed as

$$\begin{aligned} & \langle D_\varphi J(\varphi[\phi], \psi[\phi]; \phi) D\varphi[\phi], \delta\phi \rangle_{\Phi^* \times \Phi} \\ &= \langle -D_\varphi J(\varphi[\phi], \psi[\phi]; \phi), \mathcal{A}^{-1}\mathcal{V}\widehat{\psi} \rangle_{H^{-1/2}(\Gamma_\phi) \times H^{1/2}(\Gamma_\phi)} \\ &= \underbrace{\langle -\mathcal{A}^* D_\varphi J(\varphi[\phi], \psi[\phi]; \phi), \mathcal{V}\widehat{\psi} \rangle_{H^{-1/2}(\Gamma_\phi) \times H^{1/2}(\Gamma_\phi)}}_{=:\widehat{p}}, \end{aligned}$$

where $\widehat{p} \in H^{-1/2}(\Gamma_\phi)$ solves the adjoint equation

$$\mathcal{A}^*\widehat{p} = -D_\varphi J(\varphi[\phi], \psi[\phi]; \phi) \quad \text{in } H^{-1/2}(\Gamma_\phi). \quad (8)$$

As mentioned in §2.1, application of

$$\langle \widehat{p}, \mathcal{V}\widehat{\psi} \rangle_{H^{-1/2}(\Gamma_\phi) \times H^{1/2}(\Gamma_\phi)} = \langle \mathcal{V}^*\widehat{p}, \widehat{\psi} \rangle_{H^{1/2}(\Gamma_\phi) \times H^{-1/2}(\Gamma_\phi)}$$

can be performed weakly via [30, (3.12)], so as to avoid calculation of $\nabla^2 u|_{\Gamma_\phi}$. The gradient $\nabla u|_{\Gamma_\phi}$ is still required, and the non-tangential part is already available as the given Neumann data ψ . The tangential part $\nabla_{\Gamma_\phi} u$ can be computed using the extraction technique described in detail by Schwab and Wendland [29]. Formally, the idea is simply to differentiate the integral equation (2) with respect to local coordinates $\nu = 1, 2$:

$$\mathcal{A}\varphi_\nu = -\mathcal{V}\psi_\nu - \mathcal{A}_{(\nu)}\varphi - \mathcal{V}_{(\nu)}\psi. \quad (9)$$

This requires the solution of two integral equations (for the two tangential components) of the same type as (2).

2.3 Standard Adjoint Calculus

Here, we formulate adjoint equations for (2) using the following constraint functional. Let $\varphi[\phi]$ be the solution to

$$c(\phi, \varphi) := \mathcal{A}[\phi]\varphi + \mathcal{V}[\phi]\psi[\phi] = 0 \quad \text{in } H^{1/2}(\Gamma_\phi).$$

Using the implicit function theorem, we have

$$D_\phi c(\phi, \varphi[\phi]) + D_\varphi c(\phi, \varphi[\phi])D\varphi[\phi] = 0.$$

Thus, the objective function derivative can be computed via

$$\begin{aligned} & \langle D_\varphi J(\varphi[\phi], \psi[\phi]; \phi) D\varphi[\phi], \delta\phi \rangle_{\Phi^* \times \Phi} \\ &= \langle -D_\varphi J(\varphi[\phi], \psi[\phi]; \phi), D_\varphi c(\phi, \varphi[\phi])^{-1} D_\phi c(\varphi[\phi], \phi) \delta\phi \rangle_{H^{-1/2}(\Gamma_\phi) \times H^{1/2}(\Gamma_\phi)} \\ &= \underbrace{\langle -D_\varphi c(\phi, \varphi[\phi])^{-*} D_\varphi J(\varphi[\phi], \psi[\phi]; \phi), D_\phi c(\varphi[\phi], \phi) \delta\phi \rangle_{H^{-1/2}(\Gamma_\phi) \times H^{1/2}(\Gamma_\phi)}}_{=: p}, \end{aligned}$$

where $p \in H^{-1/2}(\Gamma_\phi)$ solves the adjoint equation

$$\underbrace{D_\varphi c(\phi, \varphi[\phi])^{-*}}_{=: \mathcal{A}^*} p = -D_\varphi J(\varphi[\phi], \psi[\phi]; \phi) \quad \text{in } H^{-1/2}(\Gamma_\phi). \quad (10)$$

As one would expect, (10) is the same as (8): what differs is the manner in which the two approaches account for the operator shape derivatives.

This approach does not require computation of $\nabla_{\Gamma_\phi} u$. However, it does need the Fréchet derivative $D_\phi c$ of the boundary integral operators \mathcal{K}, \mathcal{V} . These Fréchet derivatives are shown to exist in [25, 26]. For numerical methods, this compares favorably with the additional linear system solves required in the method of §2.2, so long as fast approximation methods for tensor discretizations of the Fréchet derivatives $D_\phi \mathcal{K}$ and $D_\phi \mathcal{V}$ are available. Our algorithm for this approximation is described in §3.

2.4 Comparison

For Dirichlet problems, characterization of the shape derivative via (5) has obvious advantages: it is necessary neither to use the extraction approach (since the required boundary data is already available), nor to approximate the Fréchet derivative of the integral operators. We compare the two approaches (§2.2, 2.3) for the Neumann problem (1).

The extraction approach §2.2 requires calculation of the boundary data $\nabla_{\Gamma_\phi} u$, but it has the advantage that these quantities can be computed accurately using standard integral equations: the right-hand side of (9) involves tangential derivatives of the boundary integral operators on the surface, but the left-hand side involves only the operator \mathcal{A} , so the same solution procedure can be used for these equations as for the state equation (2). The downside is that in addition to solving (2), the solution of two additional integral

equations is required. It also provides little practical advantage from the point of view of implementation, as the same kernels and surface integration required for the Fréchet derivative $D_\phi c$ are also required in the right-hand side of (9).

The approach of §2.3 on the other hand, does not require the solution of additional integral equations. It can also be easily extended to coupled systems, where the PDE in the external domain that is being solved via boundary integral equations is coupled to a PDE on a bounded domain, possibly with different physics, to be solved via FEM: it avoids the analytical work of coming up with a coupled system of PDE to replace the characterization (4) of the shape derivative. However, in order to use this approach, we need fast approximation of the tensor discretizations ∇K , ∇V of the Fréchet derivatives $D_\phi \mathcal{K}$, $D_\phi \mathcal{V}$. Our ACA-based approach to this approximation is described in §3.

3 Adaptive Cross Approximation

We will consider a Galerkin approximation of a boundary integral operator

$$\langle \mathcal{B}[\phi]u, v \rangle := \iint_{\Gamma_\phi} \kappa(x, y)u(y)v(x) \, ds_y \, ds_x$$

over Γ_ϕ with generic kernel $\kappa : \mathbb{R}^3 \times \mathbb{R}^3 \rightarrow \mathbb{R}$. Treatment of complex-valued kernels does not require any modification to our algorithm. The integrals are written over the reference domain Γ via the mapping

$$\chi := I + \phi : \Gamma \rightarrow \Gamma_\phi.$$

This necessitates the use of the Jacobian factors $\sqrt{a^x}, \sqrt{a^y}$; see, e.g., [7] for details. Hence, \mathcal{B} reads

$$\langle \mathcal{B}[\phi]u, v \rangle := \iint_{\Gamma} \kappa(\chi(x), \chi(y))u(\chi(y))v(\chi(x))\sqrt{a^y}\sqrt{a^x} \, dy \, dx. \quad (11)$$

Using the assumptions of §2.1 [25, 26], the Fréchet derivative of \mathcal{B} in the direction $\delta\phi$ can be computed via

$$\begin{aligned} \langle D_\phi \mathcal{B}[\phi, \delta\phi]u, v \rangle = & \iint_{\Gamma} \left([\partial_1 \kappa(\chi(x), \chi(y))\delta\phi(x) + \partial_2 \kappa(\chi(x), \chi(y))\delta\phi(y)] + \right. \\ & \left. \kappa(\chi(x), \chi(y)) \left[\frac{D_\phi \sqrt{a^x}}{\sqrt{a^x}} \delta\phi(x) + \frac{D_\phi \sqrt{a^y}}{\sqrt{a^y}} \delta\phi(y) \right] \right) \\ & u(\chi(y))v(\chi(x))\sqrt{a^y}\sqrt{a^x} \, dy \, dx. \end{aligned} \quad (12)$$

In order to discretize (11) and (12), we will introduce the local basis functions

$$\varrho_i, \quad i \in \{1, 2, \dots, M\}, \quad \xi_j, \quad j \in \{1, 2, \dots, N\}, \quad \Upsilon_l, \quad l \in \{1, 2, \dots, P\}.$$

The functions ϱ_i, ξ_j are used for the finite element spaces, whereas the Υ_l are used to specify the shape function ϕ via

$$\phi(x) = \sum_{l=1}^P \Upsilon_l(x) \phi_l.$$

Note that the functions Υ_l are scalar-valued, whereas the ϕ_l are vector-valued coefficients. The entries of the Galerkin matrix $B \in \mathbb{R}^{M \times N}$ corresponding to \mathcal{B} appear

$$B_{ij} = \iint_{\Gamma} \kappa(\chi(x), \chi(y)) \xi_j(y) \varrho_i(x) \sqrt{a^y} \sqrt{a^x} dy dx. \quad (13)$$

Direct calculation of this matrix requires $O(MN)$ operations, but this can of course be accelerated using methods such as the adaptive cross approximation; see §3.2. To this end, $\Gamma \times \Gamma$ has to be partitioned into sub-domains $\Gamma_1 \times \Gamma_2 \subset \Gamma \times \Gamma$ that are geometrically well-separated, i.e.,

$$\eta \text{dist}(\Gamma_1, \Gamma_2) \geq \min\{\text{diam } \Gamma_1, \text{diam } \Gamma_2\} \quad (14)$$

with some parameter $\eta > 0$. The blocks of B corresponding to $\Gamma_1 \times \Gamma_2$ are called admissible. The partitioning of $\Gamma \times \Gamma$ can be done via the algorithms presented in [4]. As a result, large patches $\Gamma_1 \times \Gamma_2$ will satisfy (14). Patches for which (14) is not valid are small, and the corresponding blocks are treated without approximation. The resulting matrix is a hierarchical matrix [18, 19].

In order to discretize (12), it is necessary to index the components of the vectors ϕ_l . The total number of shape parameters is $3P$, since each ϕ_l has three Cartesian components. We index the shape parameters via $k = 1, \dots, 3P$, corresponding to basis functions Υ_{l_k} , and Cartesian components q_k . Further, denote by e_q , $q = 1, 2$, or 3 , the Cartesian basis vectors. The result is the tensor ∇B_{ijk} , which is the derivative of B_{ij} with respect to shape parameter k :

$$\begin{aligned} \nabla B_{ijk} = \iint_{\Gamma} & \left([\partial_1 \kappa(\chi(x), \chi(y)) \Upsilon_{l_k}(x) e_{q_k} + \partial_2 \kappa(\chi(x), \chi(y)) \Upsilon_{l_k}(y) e_{q_k}] + \right. \\ & \left. \kappa(\chi(x), \chi(y)) \left[\frac{D_\phi \sqrt{a^x}}{\sqrt{a^x}} \Upsilon_{l_k}(x) e_{q_k} + \frac{D_\phi \sqrt{a^y}}{\sqrt{a^y}} \Upsilon_{l_k}(y) e_{q_k} \right] \right) \\ & \xi_j(y) \varrho_i(x) \sqrt{a^y} \sqrt{a^x} dy dx. \end{aligned} \quad (15)$$

Note that since the integrals (13), (15) can be computed over $\text{supp}(\varrho_i) \times \text{supp}(\xi_j)$ instead of $\Gamma \times \Gamma$, the only contributing shape parameters ϕ_l are those for which $\text{supp}(\varrho_i) \cap \text{supp}(\Upsilon_l) \neq \emptyset$ or $\text{supp}(\xi_j) \cap \text{supp}(\Upsilon_l) \neq \emptyset$.

Since the number of Υ_l with which e.g., a particular ϱ_i shares support, is of constant order, $\nabla B \in \mathbb{R}^{M \times N \times 3P}$ has $O(MN)$ non-zero entries. Next, we describe its structure, and propose our ACA algorithm for its approximation.

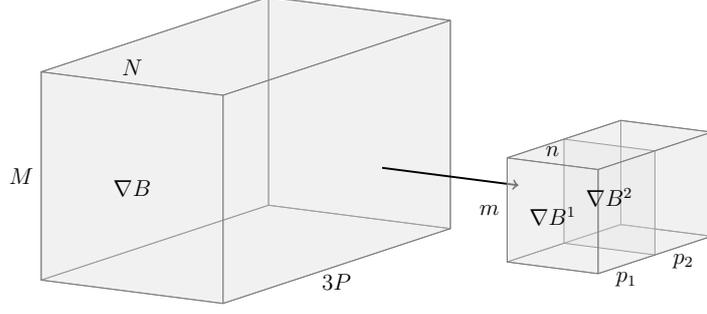


Figure 1: Extraction from ∇B of admissible blocks $\nabla B^1 \in \mathbb{R}^{m \times n \times p_1}$ and $\nabla B^2 \in \mathbb{R}^{m \times n \times p_2}$. A key feature of the algorithm is that the shape parameter indices (the third way) of ∇B^1 are associated with the rows, while the shape parameter indices of ∇B^2 are associated with the columns.

3.1 Structure of the tensor

We select an admissible block of the tensor ∇B corresponding to surface patches $\Gamma_1, \Gamma_2 \subset \Gamma$ that satisfy (14). The decomposition into such pairs can be done using the same hierarchical decomposition of the surface used to approximate the matrix (13). Let $\{\varrho_{i_1}, \dots, \varrho_{i_m}\}$ be the subset of $\{\varrho_1, \dots, \varrho_M\}$ with support in Γ_1 , and $\{\Upsilon_{l_k} : k \in \{k_1^1, k_2^1, \dots, k_{p_1}^1\}\}$ be the subset of $\{\Upsilon_1, \dots, \Upsilon_P\}$ with support in Γ_1 . Likewise, let $\{\xi_{j_1}, \dots, \xi_{j_n}\}$ be the subset of $\{\xi_1, \dots, \xi_N\}$ with support in Γ_2 , and $\{\Upsilon_{l_k} : k \in \{k_1^2, k_2^2, \dots, k_{p_2}^2\}\}$ be the subset of $\{\Upsilon_1, \dots, \Upsilon_P\}$ with support in Γ_2 . This leads to tensor blocks $\nabla B^1 \in \mathbb{R}^{m \times n \times p_1}$, and $\nabla B^2 \in \mathbb{R}^{m \times n \times p_2}$, with entries

$$\nabla B_{rst}^1 = \nabla B_{i_r j_s k_t^1}, \quad \nabla B_{rst}^2 = \nabla B_{i_r j_s k_t^2}.$$

The key to approximating these tensor blocks is to consider their sparsity structure, which is determined a priori by the mesh connectivity pattern. Since the shape parameters of ∇B^1 are associated with the rows, the matrices $\nabla B_{:s}^1$, $s = 1, \dots, n$, have the same sparsity pattern for each s . More precisely, entry ∇B_{rst}^1 is non-zero whenever

$$\text{supp}(\varrho_{i_r}) \cap \text{supp}(\Upsilon_{l_{k_t^1}}) \neq \emptyset.$$

Thus, ∇B^1 can be stored as a dense matrix of size $m_1 \times n$, where $m_1 \sim m$ is the number of non-zero entries in $\nabla B_{:s}^1$. Likewise, the matrices $\nabla B_{r::}^2$, $r = 1, \dots, m$, have the same sparsity pattern for each r , i.e., ∇B_{rst}^2 is non-zero whenever

$$\text{supp}(\xi_{j_s}) \cap \text{supp}(\Upsilon_{l_{k_t^2}}) \neq \emptyset,$$

and so ∇B^2 can be stored as a dense matrix of size $m \times n_2$, where $n_2 \sim n$ is the number of non-zero entries in $\nabla B_{r::}^2$. Thus, ∇B^1 and ∇B^2 can be approximated using a standard matrix adaptive cross approximation.

3.2 ACA Algorithm

Due to the structure of the tensors ∇B^1 and ∇B^2 , we may confine ourselves to dense matrices $C \in \mathbb{R}^{m \times n}$. If C corresponds to a non-admissible pair of domains $\Gamma_1 \times \Gamma_2$, then all the entries of C are stored without approximation. For admissible pairs, the adaptive cross approximation (ACA) [3] constructs sequences of vectors $u_k := \hat{u}_k / (\hat{u}_k)_{i_k} \in \mathbb{R}^m$ and $v_k \in \mathbb{R}^n$ via the following recursion

$$\hat{u}_k := C_{:j_k} - \sum_{\ell=1}^{k-1} (v_\ell)_{j_k} u_\ell \quad \text{and} \quad v_k := C_{i_k:} - \sum_{\ell=1}^{k-1} (u_\ell)_{i_k} v_\ell.$$

The row index i_k can be chosen as the index of the maximum entry in modulus of \hat{u}_k , i.e.

$$0 \neq |(\hat{u}_k)_{i_k}| \geq |(\hat{u}_k)_i| \quad \text{for all } i = 1, \dots, m;$$

for the choice of the column index j_k see [4]. In particular, this means that only k columns $C_{:j_\ell}$ and rows $C_{i_\ell:}$, $\ell = 1, \dots, k$, of the matrix C have to be computed for its approximation.

The required rank of the approximation k to satisfy a prescribed accuracy ε can be found from inspecting the norms of u_k and v_k , i.e., the recursion is stopped if

$$\|u_{k+1}\|_2 \|v_{k+1}\|_2 \leq \varepsilon \|UV^T\|_F,$$

where $U := [u_1, \dots, u_k]$, $V := [v_1, \dots, v_k]$ define the approximation

$$UV^T \approx C. \tag{16}$$

The number of operations required to construct UV^T is of the order $k^2(m+n)$, while the storage required for UV^T is of the order $k(m+n)$. Possible redundancies among the vectors u_ℓ, v_ℓ , $\ell = 1, \dots, k$, can be removed by orthogonalization.

Since the approximation resulting from ACA can be proved to be as good (up to constants) as the approximation in any other system of functions (e.g. polynomials, spherical harmonics, etc.), the low-rank approximation UV^T can be regarded as quasi-optimal. Hence, exponential convergence of the method can be proved if, e.g., the kernel function is asymptotically smooth, i.e., if it satisfies

$$|\partial_x^\alpha \kappa(x, y)| \leq c \gamma^p p! |x - y|^{-(s+p)} \quad \text{for all } \alpha \in \mathbb{N}_0^3,$$

where $p = |\alpha|$ and $c, \gamma, s > 0$ are constants, with respect to x or y ; see [4]. Notice that the arising kernel functions κ in \mathcal{B} satisfy this property. The following arguments show that also the kernel function of the shape derivative $D_\phi \mathcal{B}[\phi, \delta\phi]$ satisfies this prerequisite of ACA. Rewriting (12), we can decompose $D_\phi \mathcal{B}[\phi, \delta\phi]$ into the sum of two integral operators (to simplify the notation, we take $\phi = 0$)

$$\langle D_\phi \mathcal{B}[\phi = 0, \delta\phi] u, v \rangle = \iint_{\Gamma} \kappa_1(x, y) u(y) v(x) \, ds_y \, ds_x + \iint_{\Gamma} \kappa_2(x, y) u(y) v(x) \, ds_y \, ds_x$$

with kernel functions

$$\begin{aligned}\kappa_1(x, y) &:= \left(\partial_1 \kappa(x, y) + \kappa(x, y) D_\phi \sqrt{a^x} \right) \delta\phi(x), \\ \kappa_2(x, y) &:= \left(\partial_2 \kappa(x, y) + \kappa(x, y) D_\phi \sqrt{a^y} \right) \delta\phi(y).\end{aligned}$$

Furthermore, if $u, v, \delta\phi$ are linear combinations of basis functions associated with ∇B^1 , we may assume that

$$\text{supp}(v) \cap \text{supp}(\delta\phi) \neq \emptyset,$$

which due to (14) implies that $\text{supp}(u) \cap \text{supp}(\delta\phi) = \emptyset$, and hence that

$$\iint_{\Gamma_1 \times \Gamma_2} \kappa_2(x, y) u(y) v(x) \, ds_y \, ds_x = 0.$$

Similarly, if $u, v, \delta\phi$ are linear combinations of basis functions associated with ∇B^2 , the integral over κ_1 vanishes. Notice that κ_1 inherits the smoothness properties of κ with respect to y , while κ_2 inherits them with respect to x . Hence, ∇B^1 and ∇B^2 may be regarded as sub-blocks of discretizations of integral operators with asymptotically smooth kernel functions with respect to either y or x . This guarantees exponential convergence of ACA (i.e. $k \sim |\log \varepsilon|^2$) when applied to the blocks of the matrix generated from (15), as well as to blocks of the standard BEM matrix generated from (13).

Treating each block of the matrix $B \in \mathbb{R}^{M \times N}$ defined in (13) this way, the total number of operations for the construction of an approximation is of the order $k^2(M + N) \log(M + N)$ and the total amount of storage required is of the order $k(M + N) \log(M + N)$; cf. [4].

Since the sub-blocks of the tensor $\nabla B \in \mathbb{R}^{M \times N \times 3P}$ defined in (15) have the structure explained in §3.1, each of the blocks $\nabla B^1 \in \mathbb{R}^{m \times n \times p_1}$ and $\nabla B^2 \in \mathbb{R}^{m \times n \times p_2}$ can be regarded as a matrix block $C \in \mathbb{R}^{\hat{m} \times \hat{n}}$ with $\hat{m} \sim m$ and $\hat{n} \sim n$. Therefore, the complexity for treating each block of the tensor ∇B is of the same order as the complexity for approximating each block of B . Since B and ∇B are partitioned the same way, the storage complexity for ∇B is also of the order $k(M + N) \log(M + N)$ and the number of operations is of the order $k^2(M + N) \log(M + N)$.

3.3 Numerical Example

Figure 2 shows numerical results for the approximation of the single-layer potential matrix V on a sphere using N piecewise-constant basis functions, and of the corresponding shape derivative tensor ∇V with error tolerance $\varepsilon = 10^{-4}$, using \mathcal{AHMED} [2] on an AMD Opteron 2222 system with four cores.

Both V and ∇V are approximated to a fixed accuracy with complexity of order $N \log(N)$ in storage and time. Relative to V , there is an increase in the amount of computing time required to approximate ∇V by a constant factor of ~ 4 . This is due to the added cost associated with evaluation of the tensor kernel, and to the larger amount of data that must be processed. The same would also be true of evaluation of the right-hand side in (9).

N	full	compressed	ratio	time
996	3.79 MB	1.97 MB	52.10%	0.65 s
2776	29.41 MB	7.12 MB	24.20%	1.08 s
5932	134.26 MB	18.38 MB	13.69%	2.19 s
11286	485.94 MB	40.63 MB	8.36%	3.80 s
32352	3992.78 MB	141.98 MB	3.56%	12.42 s
130766	65230.9 MB	715.75 MB	1.10%	55.70 s

N	full	compressed	ratio	time
996	68.12 MB	14.53 MB	21.33%	0.89 s
2776	529.14 MB	46.80 MB	8.84%	2.84 s
5932	2416.21 MB	117.38 MB	4.86%	6.77 s
11286	8746.06 MB	251.17 MB	2.87%	14.23 s
32352	71867.9 MB	852.40 MB	1.19%	46.51 s
130766	1174150 MB	4141.34 MB	0.35%	224.20 s

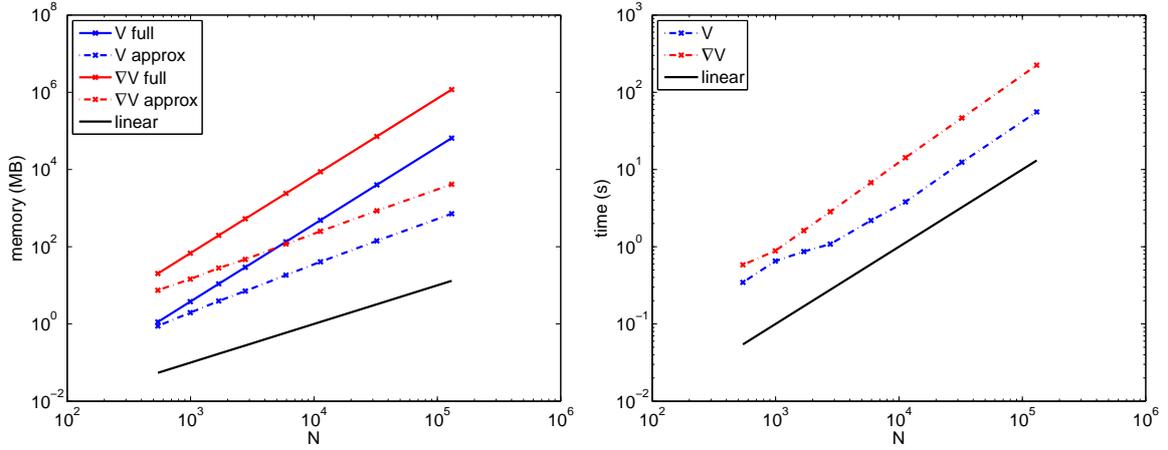


Figure 2: Memory usage and time required for approximation of V and ∇V for the Laplace problem with AHMED, and $\varepsilon = 10^{-4}$.

4 Optimization Example

As a numerical example, we will consider the free surface problem addressed by Eppler and Harbrecht in [10]. The model problem is a Laplace equation with Neumann boundary data determined by known current sources, and the goal is to determine, via shape optimization, the free surface of a liquid metal bubble in a magnetic field produced by the current sources.

Eppler and Harbrecht assume a star-shaped domain, with a smooth boundary described by spherical harmonics. Due to the nature of the objective function, they are able to avoid using a characterization such as (4) to compute derivatives of the state. However, they must still compute tangential derivatives of the state on the surface, which is feasible due to the smoothness of their basis functions. Using the adjoint approach described in §2.3 allows us to consider general boundary shapes with different topologies, and to avoid computing tangential derivatives of the state entirely: we apply adjoint equations to the gradient of [10, (1.17)] rather than computing [10, (1.18)].

The state problem is to find the scalar potential u via

$$-\Delta u = 0 \quad \text{in } \Omega^+, \quad (17a)$$

$$\frac{\partial u}{\partial n} = -(\nabla \times A) \cdot n \quad \text{on } \Gamma, \quad (17b)$$

$$u = O(1/|x|) \quad \text{as } |x| \rightarrow \infty, \quad (17c)$$

where the vector potential A is determined via integration over the current distribution \vec{j} via

$$A(x) = \frac{\mu}{4\pi} \int_{\mathbb{R}^3} \frac{\vec{j}(y)}{|x-y|} dy. \quad (18)$$

The magnetic field is given by

$$B = \nabla \times A + \nabla u.$$

Equilibrium conditions can be formulated as an optimization problem for the boundary shape, by minimization of the total energy stored in the magnetic field, the surface tension, and the gravitational potential; see [10] for further details. Boundary data of the Neumann problem (17) can be expressed as a solution to the boundary integral equation (2), which was discretized using piecewise-linear Galerkin boundary elements, treating the boundary integral operator $\frac{1}{2}I - \mathcal{K}$ as a mapping $L^2(\Gamma) \rightarrow L^2(\Gamma)$; see [28, §3.8]. All quantities in the objective function and their derivatives can also be formulated in terms of boundary integrals, so that calculations need only be performed on the boundary.

The mesh was generated using Gmsh [15]. Approximation of the BEM matrices and tensors was done with A \mathcal{H} MED [2]. Optimization was done using the interior-point solver Ipopt [31]. Since the optimization variables are the individual mesh nodes,

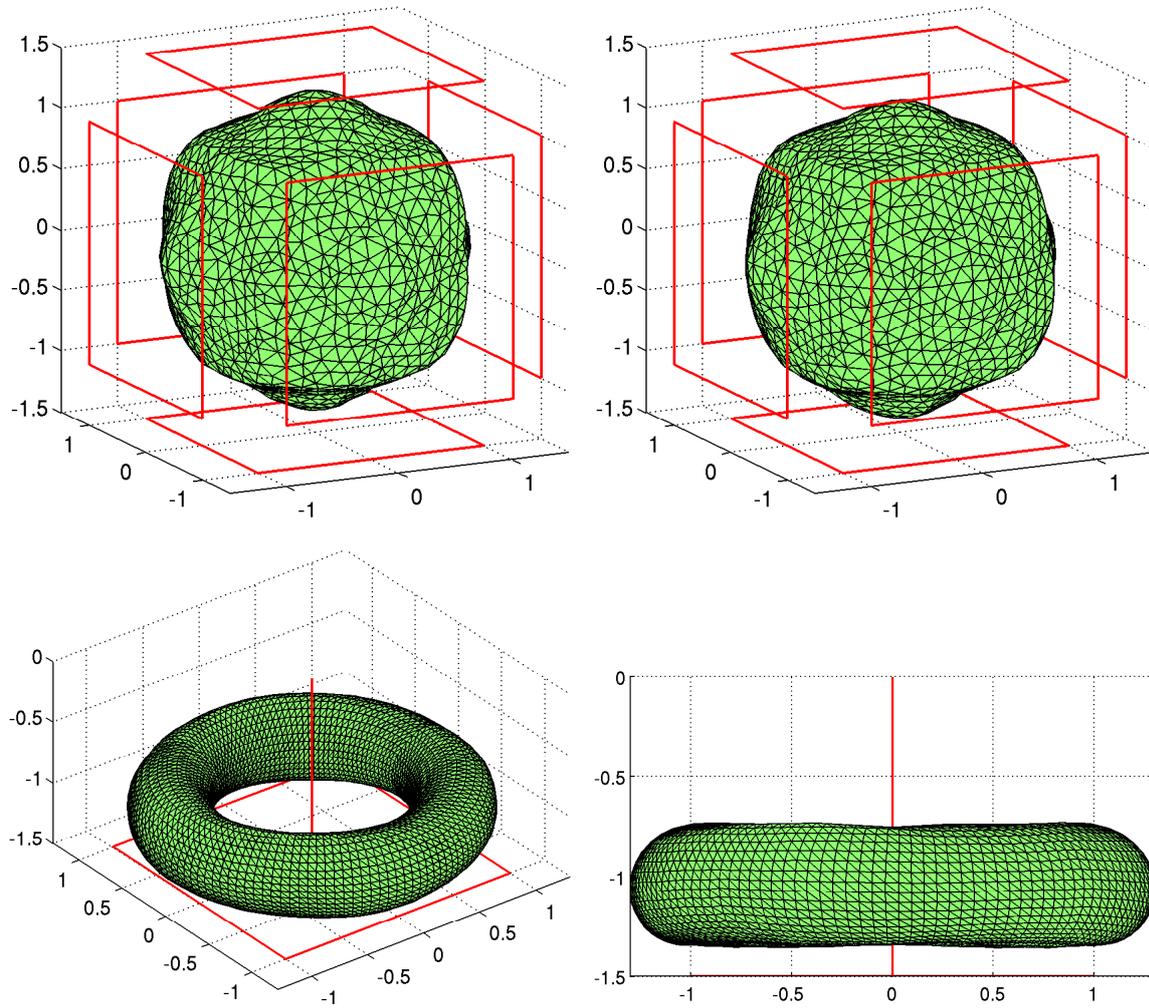


Figure 3: Above from left to right, the optimized shapes, respectively without and with gravity, reproducing the results of [10, Fig. 4]. Below: an optimized torus, with gravity, using only the bottom loop, but with an added infinite wire through the center.

smoothness constraints were needed, as is typical for elliptic shape optimization problems. We used bounds on the tangential derivatives of the surface displacement ϕ , computed using the discrete curvature estimation algorithm of [27].

Optimization results are shown in figure 3. In order to produce results similar to those of Eppler and Harbrecht, [10], we scaled the constants A, B associated with the relative strength of the surface tension and gravitational potential by a factor of $(8\pi)^{-2}$: our best guess is that they omitted the factor $\mu/(4\pi)$ in their implementation of (18). Using a mesh of 2776 triangles for the sphere and a stopping tolerance of 10^{-5} , the optimization problem with gravity required 53 minutes on an AMD Opteron 2222 system with four cores. Without gravity, the tolerance 10^{-5} was too large, and with a tolerance of 10^{-6} , it stopped after 96 minutes. The times required are competitive with those reported by Eppler and Harbrecht despite the fact that their approach does not require any tensor approximations. However, the generality of our approach allows us also to consider the torus example shown at the bottom of figure 3: with 9072 triangles and a stopping tolerance of 10^{-5} , this calculation took 9 hours, 38 minutes. With a stopping tolerance of $2 \cdot 10^{-5}$, it would require only about 3 hours.

We emphasize that this example problem is chosen for its relative simplicity, and because it can be formulated using the same integral equations as in §2.3. Full practical benefits of our approach will be realized when characterizations such as (4) cannot be avoided, and for coupled problems with Neumann data, such as in structural acoustics.

Acknowledgment

Numerical experiments leading to the development of the algorithm presented in this article were performed using the Matlab tensor toolbox [1].

References

- [1] B. W. Bader and T. G. Kolda. Matlab tensor toolbox version 2.4. Technical report, Sandia National Laboratories, 2010. <http://csmr.ca.sandia.gov/~tgkolda/TensorToolbox/>.
- [2] M. Bebendorf. *AHMED* – another software library on hierarchical matrices for elliptic differential equations. available online <http://bebendorf.ins.uni-bonn.de/AHMED.html>.
- [3] M. Bebendorf. Approximation of boundary element matrices. *Numer. Math.*, 86(4):565–589, 2000.
- [4] M. Bebendorf. *Hierarchical matrices*, volume 63 of *Lecture Notes in Computational Science and Engineering*. Springer-Verlag, Berlin, 2008.
- [5] G. Beylkin, R. Coifman, and V. Rokhlin. Fast wavelet transforms and numerical algorithms. I. *Comm. Pure Appl. Math.*, 44(2):141–183, 1991.

- [6] O. P. Bruno and L. A. Kunyansky. A fast, high-order algorithm for the solution of surface scattering problems: basic implementation, tests, and applications. *J. Comput. Phys.*, 169(1):80–110, 2001.
- [7] P. G. Ciarlet. An introduction to differential geometry with application to elasticity. *J. Elasticity*, 78/79(1-3):iv+215, 2005.
- [8] R. Coifman, V. Rokhlin, and S. Wandzura. The fast multipole method for the wave equation: a pedestrian prescription. *IEEE Antennas Propagat. Mag.*, 35:7–12, 1993.
- [9] K. Eppler and H. Harbrecht. Numerical solution of elliptic shape optimization problems using wavelet-based BEM. *Optimization Methods and Software*, 18(1):105–123, 2003.
- [10] K. Eppler and H. Harbrecht. Fast wavelet BEM for 3d electromagnetic shaping. *Appl. Numer. Math.*, 54(3–4):537–554, 2005.
- [11] K. Eppler and H. Harbrecht. Coupling of FEM and BEM in shape optimization. *Numer. Math.*, 104(1):47–68, 2006.
- [12] K. Eppler and H. Harbrecht. Efficient treatment of stationary free boundary problems. *Appl. Numer. Math.*, 56(10):1326–1339, 2006.
- [13] M. Fischer. *The fast multipole boundary element method and its application to structure-acoustic field interaction*. PhD thesis, Bericht aus dem Institut A für Mechanik 2004/2. Universität Stuttgart, Stuttgart, Germany, 2004.
- [14] M. Fischer and L. Gaul. Fast BEM-FEM mortar coupling for acoustic-structure interaction. *Internat. J. Numer. Methods Engrg.*, 62(12):1677–1690, 2005.
- [15] C. Geuzaine and J.-F. Remacle. Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities. *Internat. J. Numer. Methods Engrg.*, 79(11):1309–1331, 2009.
- [16] L. F. Greengard and V. Rokhlin. A fast algorithm for particle simulations. *J. Comput. Phys.*, 73(2):325–348, 1987.
- [17] L. F. Greengard and V. Rokhlin. A new version of the fast multipole method for the Laplace equation in three dimensions. In *Acta numerica, 1997*, volume 6 of *Acta Numer.*, pages 229–269. Cambridge Univ. Press, Cambridge, 1997.
- [18] W. Hackbusch. A sparse matrix arithmetic based on \mathcal{H} -matrices. Part I: Introduction to \mathcal{H} -matrices. *Computing*, 62(2):89–108, 1999.
- [19] W. Hackbusch and B. N. Khoromskij. A sparse \mathcal{H} -matrix arithmetic. Part II: Application to multi-dimensional problems. *Computing*, 64(1):21–47, 2000.
- [20] W. Hackbusch and Z. P. Nowak. On the fast matrix multiplication in the boundary element method by panel clustering. *Numer. Math.*, 54(4):463–491, 1989.

- [21] S. Hardesty. *Optimization of Shell Structure Acoustics*. PhD thesis, Department of Computational and Applied Mathematics, Rice University, Houston, TX, 2010. Available as CAAM TR10-16 from http://www.caam.rice.edu/tech_reports.html.
- [22] J. B. Mariem and M. A. Hamdi. A new boundary finite element method for fluid-structure interaction problems. *International Journal for Numerical Methods in Engineering*, 24:1251–1267, 1987.
- [23] N. Nemitz and M. Bonnet. Topological sensitivity and fimm-accelerated bem applied to 3d acoustic inverse scattering. *Engineering Analysis with Boundary Elements*, 32(11):957 – 970, 2008.
- [24] G. Of, A. Schwaigkofler, and O. Steinbach. Boundary integral equation methods for inverse problems in electrical engineering. *Elektrotechnik und Informationstechnik*, 124:254–259, 2007. 10.1007/s00502-007-0451-6.
- [25] R. Potthast. Fréchet differentiability of boundary integral operators in inverse acoustic scattering. *Inverse Problems*, 10(2):431–447, 1994.
- [26] R. Potthast. Fréchet differentiability of the solution to the acoustic Neumann scattering problem with respect to the domain. *J. Inverse Ill-Posed Probl.*, 4(1):67–84, 1996.
- [27] S. Rusinkiewicz. Estimating curvatures and their derivatives on triangle meshes. *3D Data Processing Visualization and Transmission, International Symposium on*, pages 486–493, 2004.
- [28] S. Sauter and C. Schwab. *Boundary Element Methods*, volume 39 of *Springer Series in Computational Mathematics*. Springer-Verlag, New York, 2011.
- [29] C. Schwab and W. L. Wendland. On the extraction technique in boundary integral equations. *Math. Comp.*, 68(225):91–122, 1999.
- [30] J. Sokołowski and J.-P. Zolésio. *Introduction to shape optimization*, volume 16 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, 1992. Shape sensitivity analysis.
- [31] A. Wächter and L. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Math. Program.*, 106(1, Ser. A):25–57, 2006.
- [32] L. Ying, G. Biros, and D. Zorin. A kernel-independent adaptive fast multipole algorithm in two and three dimensions. *J. Comput. Phys.*, 196(2):591–626, 2004.

Bestellungen nimmt entgegen:

Sonderforschungsbereich 611
der Universität Bonn
Endenicher Allee 60
D - 53115 Bonn

Telefon: 0228/73 4882

Telefax: 0228/73 7864

E-Mail: astrid.avila.aguilera@ins.uni-bonn.de

<http://www.sfb611.iam.uni-bonn.de/>

Verzeichnis der erschienenen Preprints ab No. 490

490. Bartels, Sören; Mielke, Alexander; Roubicek, Tomas: Quasistatic Small-strain Plasticity in the Limit of Vanishing Hardening and its Numerical Approximation
491. Bebendorf, Mario; Venn, Raoul: Constructing Nested Bases Approximations from the Entries of Non-local Operators
492. Arguin, Louis-Pierre; Bovier, Anton; Kistler, Nicola: The Extremal Process of Branching Brownian Motion
493. Adler, Mark; Ferrari, Patrik L.; van Moerbeke, Pierre: Non-intersecting Random Walks in the Neighborhood of a Symmetric Tacnode
494. Bebendorf, Mario; Bollhöfer, Matthias; Bratsch, Michael: Hierarchical Matrix Approximation with Blockwise Constrains
495. Bartels, Sören: Total Variation Minimization with Finite Elements: Convergence and Iterative Solution
496. Kurzke, Matthias; Spirn, Daniel: Vortex Liquids and the Ginzburg-Landau Equation
497. Griebel, Michael; Harbrecht, Helmut: On the Construction of Sparse Tensor Product Spaces
498. Knüpfer, Hans; Kohn, Robert V.; Otto, Felix: Nucleation Barriers for the Cubic-to-tetragonal Phase Transformation
499. Frehse, Jens; Specovius-Neugebauer, Maria: Fractional Differentiability for the Stress Velocities to the Prandtl-Reuss Problem
500. McCord, Jeffrey; Otto, Felix; Schäfer, Rudolf; Steiner, Jutta; Wiecezoreck, Holm: The Formation and Coarsening of the Concertina Pattern
501. Bartels, Sören: Finite Element Approximation of Large Bending Isometries
502. Knüpfer, Hans; Muratov, Cyrill B.: On an Isoperimetric Problem with a Competing Non-local Term. I. The Planar Case.
503. Bebendorf, Mario; Kühnemund, Andreas; Rjasanow, Sergej: A Symmetric Generalization of Adaptive Cross Approximation for Higher-Order Tensors
504. Griebel, Michael; Wissel, Daniel: Fast Approximation of the Discrete Gauss Transform in Higher Dimensions

505. Ferrari, Patrik L.; Frings, René: Finite Time Corrections in KPZ Growth Models
506. Bartels, Sören: Approximation of Large Bending Isometries with Discrete Kirchhoff Triangles
507. Bartels, Sören; Dolzmann, Georg; Nochetto, Ricardo H.; Raisch, Alexander: Finite Element Methods for Director Fields on Flexible Surfaces
508. Bartels, Sören; Roubíček, Tomáš: Numerical Approaches to Thermally Coupled Perfect Plasticity
509. Bebendorf, Mario; Hardesty, Sean: Adaptive Cross Approximation of Tensors Arising in the Discretization of Boundary Integral Operator Shape Derivatives