Coarsening of Finite Element and Boundary Element Spaces

W. Hackbusch (Leipzig), M. Löhndorf (Reading, GB), and S.A. Sauter (Zürich)

Abstract

In this paper, we will present composite boundary elements (CBE) for classical Fredholm boundary integral equations. These new boundary elements allow the low-dimensional discretisation of boundary integral equation where the minimal number of degrees of freedom is *independent* of the, possibly, huge number of charts which are necessary to describe a complicated surface.

The applications are three-fold: (a) The coarse grid discretisation by composite boundary elements allow the use of multi-grid algorithms for solving the fine grid discretisation independently of the number of patches which are necessary to describe the surface. (b) If the accuracy requirements are moderate the composite boundary elements allow the low-dimensional discretisation of the integral equation. (c) A posteriori error indicators can be applied already to a low dimensional discretisation, which do not resolve the domain, to obtain a problem-adapted discretisation.

1 Introduction

In this paper, we will introduce composite boundary elements (CBE) for solving classical boundary integral equations in $\mathbb{R}^3$. To be specific, let $\Omega \subset \mathbb{R}^3$ be a bounded Lipschitz domain with boundary $\Gamma := \partial\Omega$. The Sobolev spaces on Γ are denoted by $H^\mu(\Gamma)$, where the range of $\mu \in [-\kappa, \kappa] \subset \mathbb{R}$ may be restricted due to the smoothness of the surface. Our goal is to solve boundary integral equations which are given in a variational formulation: Find $u \in H^\mu(\Gamma)$ such that

$$(v, \lambda u)_0 + (v, Ku)_0 = (f, v)_0 ,$$

where $(\cdot,\cdot)_0$ denotes the continuous extension of the $L^2(\Gamma)$ scalar product to the dual pairing $H^\mu(\Gamma) \times H^{-\mu}(\Gamma)$. The data are the right-hand side $f \in H^{-\mu}(\Gamma)$, the piecewise constant function $\lambda : \Gamma \to \mathbb{R}$, and the boundary integral operator

$$Ku := \int_\Gamma k(x,y)\, u(y)\, ds_y,$$

where $k : \Gamma \times \Gamma \to \mathbb{C}$ is the kernel function. We assume that $\lambda I + K : H^\mu(\Gamma) \to H^{-\mu}(\Gamma)$ is an isomorphism.

Remark 1.1 *In some cases as, e.g., for the hypersingular integral equation of the Laplace equation, the null-space* $\ker(\lambda I + K)$ *is non-trivial. This problem can be cured easily, by restricting* $H^\mu(\Gamma)$ *to an appropriate quotient space or by extending the system using Lagrange multipliers. Here, we do not discuss these technical details.*

Example 1.2 *The classical single layer operator for the Laplace equation is characterised by* $k(x,y) := (4\pi \|x-y\|)^{-1}$ *and* $\lambda = 0$*. For given* $f \in H^{1/2}(\Gamma)$*, we are seeking* $u \in H^{-1/2}(\Gamma)$ *such that*

$$\int_\Gamma \int_\Gamma \frac{v(x)\, u(y)}{4\pi \|x-y\|} ds_y ds_x = f(v) \qquad \forall v \in H^{-1/2}(\Gamma).$$

In this paper, we will consider such types of boundary integral equations and focus to the case that the surface Γ is very complicated, i.e., the parametric description of Γ requires a very large number of surface patches along corresponding charts to local parameter domains.

We consider two applications for this class of problems.

1. The number of patches is so large that the resolution by standard boundary elements exceeds the computing and memory capacities of the computers at hand. On the other hand, the accuracy requirements are only moderate that, from the viewpoint of the approximation property, a low dimensional boundary element space would be appropriate. Only the geometric details *enforce* a very fine boundary element mesh which results in a finite element space of very large dimension.

2. The number of patches in the surface description of Γ again is very large while they are resolved by a fine boundary element mesh. Here, the goal is to solve the corresponding linear system on the fine mesh efficiently, e.g., by multigrid methods. Since the efficiency of fast solvers such as multigrid methods and wavelets are based on a hierarchical discretisation of the underlying problem, they loose their efficiency since standard boundary elements do not allow the low dimensional discretisation of the underlying equation due to the very large number of geometric details in the surface.

We will introduce *Composite Boundary Elements (CBE)* which can be regarded as a coarsening strategy for boundary element spaces allowing the low dimensional discretisation of boundary integral operators on complicated surfaces. The minimal dimension of the space is independent of the number and size of the geometric details in Γ. The construction will be based on a graph coarsening algorithm and can be applied to (unstructured) finite element meshes for the discretisation of partial differential equations as well.

In the literature, various methods exist for coarsening finite element spaces. Some of the them are based on coarsening of simplicial meshes in the Euclidean space (see, e.g., [13], [12], [21], [18], [9], [3], [2], [8]) some of them are based on the construction of auxiliary meshes (see, e.g., [26], [10], [27]) or reduced meshes ([14]).

Also, algebraic multigrid methods involve in many cases a graph coarsening (see [20], [25], [4], [17], [1], [19]).

For boundary element methods, the literature is not as vast as for finite element methods. Multigrid methods for non-local operators (including also operators of negative order) are described, e.g., in [5], [6], and [7]. A combination of algebraic multigrid methods with data-sparse representation of boundary integral operators has been described in [15] and [16].

Coarsening strategies for wavelet discretisation have been presented for boundary integral equations, e.g., in [24], [22].

2 Finite and Boundary Element Methods

We assume in an abstract way that a bilinear form $a : H^\mu(\Omega) \times H^\mu(\Omega) \to \mathbb{R}$ is given which is continuous and H^μ-coercive and injective: There exist constants $C_1, c_2 > 0$ and $C_2 \geq 0$ such that, for all $u, v \in H^\mu(\Omega)$, there holds

$$|a(u,v)| \leq C_1 \|u\|_{H^\mu(\Omega)} \|v\|_{H^\mu(\Omega)} \tag{2.1a}$$

$$\operatorname{Re} a(u,u) \geq c_2 \|u\|^2_{H^\mu(\Omega)} - C_2 \|u\|^2_{H^\nu(\Omega)} \quad \text{for some } \nu < \mu. \tag{2.1b}$$

$$u \neq 0 \implies \sup_{w \in H^\mu(\Gamma)} a(u,w) > 0. \tag{2.1c}$$

For given right-hand side $f \in (H^\mu(\Omega))'$, we are seeking $u \in H^\mu(\Omega)$ such that

$$a(u,v) = f(v) \qquad \forall v \in H^\mu(\Omega). \tag{2.2}$$

It is well known that conditions (2.1a-c) ensure existence, uniqueness and well-posedness of problem (2.2).

Let $S \subset H^\mu(\Omega)$ be a finite dimensional subspace of $H^\mu(\Omega)$. Then, the Galerkin discretisation is given by seeking $u_S \in S$ such that

$$a(u_S, v) = f(v) \qquad \forall v \in S.$$

We are left with the problem of constructing (a nested sequence of) appropriate (finite element) spaces $S \subset H^\mu(\Omega)$. Let $\mathcal{G}_0$ denote a conforming (in the sense of Ciarlet) triangulation of a bounded domain $\Omega \subset \mathbb{R}^d$ resp. of a d-dimensional hypersurface $\Omega \subset \mathbb{R}^{d+1}$. Let

$$\Theta_0 : \text{set of vertices in } \mathcal{G}_0, \qquad \mathcal{E}_0 : \text{set of edges in } \mathcal{G}_0. \tag{2.3}$$

We assume that – due to the possibly very complicated shape of Ω – the numbers $\sharp\mathcal{G}_0$, $\sharp\Theta_0$, $\sharp\mathcal{E}_0$ are very large accordingly and the goal of this paper is to define an algorithm for coarsening the corresponding finite element space. Note that we will use throughout the paper the terminology *finite element space* for both, *boundary element space* on surfaces and classical *finite element space* on a domain. As the "standard" finite element space we consider the space of continuous, piecewise linear functions

$$S_0 := \left\{ u \in C^0(\overline{\Omega}) \mid \forall \tau \in \mathcal{G}_0 : u|_\tau \in \mathbb{P}_1 \right\}. \tag{2.4}$$

(Note that, for curved elements, the pullbacks $\chi_\tau : \hat{\tau} \to \tau$ to the unit element $\hat{\tau}$ have to satisfy $u \circ \chi_\tau \in \mathbb{P}_1$ instead of $u \in \mathbb{P}_1$).

Remark 2.1 *The coarsening algorithm is not restricted to the case that S_0 is the space of continuous, piecewise linear functions. The interpolation operators (cf. Subsection 3.2.2) can be modified in a canonical way, so that the coarsening strategies can be applied to piecewise constant discontinuous boundary element spaces or higher order elements as well.*

3 Coarsening of Finite Element and Boundary Element Spaces

In this section, we will present an algorithm which generates a sequence of coarsened finite element spaces, i.e., lower dimensional subspaces of S_0. We will use negative indices, e.g., $\mathcal{E}_{-1}, \mathcal{E}_{-2}, \ldots$ for quantities on coarser levels.

The construction of the coarsened finite element spaces is recursive and starts with the "standard" finite element space S_0. In this light, we assume that a set of vertices $\Theta_{-\ell}$, a set of edges $\mathcal{E}_{-\ell} \subset \Theta_{-\ell} \times \Theta_{-\ell}$, and the basis functions $(\varphi_{-\ell,x})_{x\in\Theta_{-\ell}}$ of the space $S_{-\ell}$ are already generated. In the sequel, we will explain how to generate the next coarser level $(\Theta_{-\ell-1}, \mathcal{E}_{-\ell-1})$ based on this input.

3.1 Graph Coarsening

First we will introduce some notations.

Notation 3.1 *Let $\mathcal{G}$ denote a graph, embedded in $\mathbb{R}^d$, which consists of a set of vertices $\Theta \subset \mathbb{R}^d$ and set of edges $\mathcal{E} \subset \Theta \times \Theta$. Layers about a vertex $x \in \Theta$ of width $n \in \mathbb{N}_0$ are defined by the recursion*

$$\begin{aligned} \mathcal{L}^{(0)}(x) &:= \{x\}, \\ \mathcal{L}^{(i+1)}(x) &:= \{y \in \Theta \mid \exists z \in \mathcal{L}_i(x) : (y,z) \in \mathcal{E}\} \cup \mathcal{L}^{(i)}(x). \end{aligned}$$

The distance $\operatorname{dist}_{\mathcal{G}}(x,y)$ of two points $x, y \in \Theta$ in the graph $\mathcal{G}$ is the minimal number of edges in $\mathcal{E}$ which are necessary to connect x and y, i.e.,

$$\operatorname{dist}_{\mathcal{G}}(x,y) := \min\left\{ n : y \in \mathcal{L}^{(n)}(x) \right\}.$$

For a subset $\omega \subset \Theta$ and $x \in \Theta$, we define

$$\operatorname{dist}_{\mathcal{G}}(x,\omega) := \min_{y\in\omega} \operatorname{dist}_{\mathcal{G}}(x,y).$$

If $\mathcal{G}$, Θ, $\mathcal{E}$ are supplied with a level index ℓ, e.g., $\mathcal{G}_\ell$, the corresponding layers are denoted by $\mathcal{L}_\ell^{(i)}(x)$ and the distance by $\operatorname{dist}_\ell(x,y)$.

The algorithm consists of two steps: 1: Geometric coarsening, 2: Definition of the basis functions.

The geometric coarsening starts with the selection of coarse nodal points. Once, these points have been selected, the set of edges for the coarse graph will be generated. We employ heuristic criteria in such a way that for a regularly refined simplicial mesh the coarse meshes satisfy these criteria.

For a graph $(\Theta_{-\ell}, \mathcal{E}_{-\ell})$, we first define the set of coarse nodal points $\Theta_{-\ell-1}$. The definition of the set of edges, then, depends on a control parameter $\eta \in \{2,3\}$ and is given by

$$\mathcal{E}_{-\ell-1} := \{(x,y) \in \Theta_{-\ell-1} \times \Theta_{-\ell-1} : \operatorname{dist}_\ell(x,y) \le \eta\}. \tag{3.1}$$

Algorithm 3.2
procedure choice_of_coarse_grid_points:

Initialisation: $\Theta_{temp} := \Theta_{-\ell}$; $\Theta_{-\ell-1} := \emptyset$;

begin
1: *Pick some* $x \in \Theta_{temp}$; (3.2)
$\Theta_{-\ell-1} := \Theta_{-\ell-1} \cup \{x\}$;
$\Theta_{temp} := \Theta_{temp} \backslash \mathcal{L}_{-\ell}^{(1)}(x)$;
if $\Theta_{temp} \neq \emptyset$ ***goto 1;***
Generate $\mathcal{E}_{-\ell-1} := \{(x,y) \in \Theta_{-\ell-1} \times \Theta_{-\ell-1} : \mathrm{dist}_{\ell}(x,y) \leq \eta\}$
end;

This algorithm selects coarse grid nodes out of the finer meshes $\Theta_{-\ell}$ in an unstructured (random) way. An advanced version of this algorithm is based on the following idea. If a sequence of simplicial meshes is generated by standard mesh refinement strategies, i.e., by connecting midpoints of edges, then, any two coarse mesh points have a distance of two edges in the *finer* mesh. In this light, we define the modification of the selection algorithm as follows. Only the first coarse grid point is chosen randomly. After that, the neighbours of second order of the already chosen coarse grid points are collected in a list and will be picked first in the next step.

Algorithm 3.3
procedure advanced_choice_of_coarse_grid_points:

Initialisation: $\Theta_{temp} := \Theta_{-\ell}$; $\Theta_{-\ell-1} := \emptyset$; $\mathcal{N}_{temp} := \emptyset$;

begin
1: ***if*** $\mathcal{N}_{temp} = \emptyset$ *choose some* $x \in \Theta_{temp}$ ***else*** *choose some* $x \in \mathcal{N}_{temp}$;
$\Theta_{-\ell-1} := \Theta_{-\ell-1} \cup \{x\}$;
$\Theta_{temp} := \Theta_{temp} \backslash \mathcal{L}_{-\ell}^{(1)}(x)$;
$\mathcal{N}_{temp} := \{y \in \Theta_{-\ell} : \mathrm{dist}_{-\ell}(y, \Theta_{\mathrm{temp}}) = 2\}$;
if $\Theta_{temp} \neq \emptyset$ ***goto 1;***
Generate $\mathcal{E}_{-\ell-1} := \{(x,y) \in \Theta_{-\ell-1} \times \Theta_{-\ell-1} : \mathrm{dist}_{\ell}(x,y) \leq \eta\}$
end;

Remark 3.4 *The update of the set* $\mathcal{N}_{temp}$ *and the computation of the set of edges* $\mathcal{E}_{-\ell-1}$ *can be easily realised locally. We do not present the details here.*

3.2 Space Coarsening

In the next step, we will define composite finite elements on these thinned-out graphs in a recursive, hierarchical way. Since $\mathcal{G}_0$ is a standard finite element mesh the definition of the corresponding finite element space S_0 is as usual (see (2.4)).

For the finite element *basis* function corresponding to the level $-\ell-1$ and a nodal point $y \in \Theta_{-\ell-1}$, we employ the ansatz

$$\varphi_{-\ell-1,y} = \sum_{x \in \Theta_{-\ell}} \alpha_{x,y}^{(-\ell-1)} \varphi_{-\ell,x}, \tag{3.3}$$

where the interpolation weights will be specified later. This ansatz automatically guarantees nestedness of the corresponding composite finite element spaces

$$S_{-\ell} := \mathrm{span}\{\varphi_{-\ell,x} : x \in \Theta_{-\ell}\},$$
$$S_0 \supset S_{-1} \supset S_{-2} \supset \ldots \supset S_{-\ell} \supset S_{-\ell-1} \supset \ldots.$$

The choice of the coefficients $\alpha_{x,y}^{(-\ell-1)}$ in (3.3) will be related to the standard case, where $(S_{-\ell})_{\ell}$ is a sequence of nested finite element spaces on regular and nested simplicial finite element meshes, by means of the following conditions.

1. (Lagrange property)
$$\forall x, y \in \Theta_{-\ell,x} : \varphi_{-\ell,x}(y) = \begin{cases} 1 & x = y, \\ 0 & \text{otherwise.} \end{cases}$$

2. (Interpolation property) For a simplex τ and a vertex y of τ, let $\lambda_{\tau,y}$ denote the affine function with values 1 at y and 0 at all other vertices of τ. For $x \in \Theta_{-\ell}$, $y \in \Theta_{-\ell-1}$, we define[1] (cf. Figure 1)
$$\alpha_{x,y}^{(-\ell-1)} := \begin{cases} \lambda_{\tau,y}(x) & \text{if } y \text{ is a vertex of a simplex } \tau \in \mathcal{G}_{-\ell-1} \text{ with } x \in \overline{\tau}, \\ 0 & \text{otherwise.} \end{cases} \tag{3.4}$$

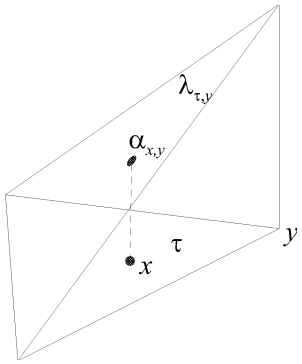


Figure 1: Construction of the interpolation weights $\alpha_{x,y}$. The piecewise affine function $\lambda_{\tau,y}$ is evaluated at the fine grid point x.

The definition of the coefficients $\alpha_{x,y}^{(-\ell-1)}$ consists of two steps:

(a) For any $y \in \Theta_{-\ell}$, construct a simplex τ with vertices in $\Theta_{-\ell-1}$ such that $x \in \overline{\tau}$ (resp. $\operatorname{dist}(x,\tau)$ is small). This step is in analogy to the first case in (3.4).

(b) Compute the weights $\alpha_{x,y}^{(-\ell-1)}$.

Remark 3.5 *In the special case of a regular, nested sequence of triangulations, there holds, for any $u_{-\ell-1} \in S_{-\ell-1}$,*
$$u_{-\ell-1} = \sum_{x \in \Theta_\ell} u_{-\ell-1}(x)\, \varphi_{\ell,x} \qquad \text{on } \Omega.$$
Choosing $u_{-\ell-1}$ as the basis function $\varphi_{-\ell-1,y}$ and using the Lagrange property, we obtain the relation
$$\alpha_{x,y}^{(-\ell-1)} = \varphi_{-\ell-1,y}(x)$$
for all $x \in \Theta_{-\ell}$ and $y \in \Theta_{-\ell-1}$. Hence, in this standard case, the definition of the weights $\alpha_{x,y}^{(-\ell-1)}$ in (3.4) coincides with the standard definition.

3.2.1 Selection of Nodal Points for the Interpolation Operator

Below, both steps (a,b) will be described in an algorithmic way. Procedure **choice_of_interpolation_points** depends on a parameter $\gamma \geq d+1$ which controls the number of coarse grid points which are examined for the choice of the interpolation points. Furthermore, an abstract function **check_quality**$(x, \{z_0, z_1, \ldots, z_d\}) \in \mathbb{R}_{>0}$ is employed to measure the interpolation quality of nodal points z_i, $0 \leq i \leq d$, for an evaluation point x. The interpolation quality is considered to be optimal if **check_quality** is minimal. The details are described later. Here, we proceed with the algorithm for selecting the set of coarse grid points for the definition of the interpolation operator.

[1] This definition will be refined if Ω is d-dimensional manifold (cf. Section 3.2.2).

Algorithm 3.6

procedure choice_of_interpolation_points $(\gamma : \text{integer})$***;***
begin
  for all $x \in \Theta_{-\ell}$ ***do begin***
    if $x \in \Theta_{-\ell-1}$ ***then*** $\Theta_{\text{int}}(x) := \{x\}$ ***else***
    begin
      initialise $\Theta_{\text{temp}} := \emptyset$; $i := 0$***;***
      while $\sharp\Theta_{\text{temp}} < \gamma$ ***do begin***
        $\Theta_{\text{temp}} := \Theta_{\text{temp}} \cup \left(\mathcal{L}^{(i)}_{-\ell}(x) \cap \Theta_{-\ell-1}\right)$; $i := i + 1$
      end;
      quality $:= +\infty$***;***
      for all subsets $A \subset \Theta_{\text{temp}}$ ***with*** $\sharp A = d + 1$ ***do begin***
        $q :=$ **check_quality** (x, A);
        if $q <$ quality ***then begin*** quality $:= q$; $\Theta_{\text{int}}(x) := A$ ***end***
      end
    end
  end
end;

3.2.2 Definition of Interpolation Weights

It remains to define the coefficients $\alpha^{(-\ell-1)}_{x,y}$ by using the set $\Theta_{\text{int}}(x)$ of interpolation points as defined in Algorithm 3.6. For $x = y \in \Theta_{-\ell-1}$ we simply put

$$\alpha^{(-\ell-1)}_{x,x} := 1 \tag{3.5a}$$

and, for $y \notin \Theta_{\text{int}}(x)$, we set

$$\alpha^{(-\ell-1)}_{x,y} := 0. \tag{3.5b}$$

For the remaining cases, $y \in \Theta_{\text{int}}(x) \setminus \{x\}$, we distinguish two cases.

1. $\Omega \subset \mathbb{R}^d$ is a d-dimensional domain.

   Let $x \in \Theta_{-\ell} \backslash \Theta_{-\ell-1}$ and $\Theta_{\text{int}}(x)$ be constructed as in procedure **choice_of_interpolation_points**. For any $y \in \Theta_{\text{int}}(x)$, an affine function $p \in \mathbb{P}_1$, $p : \mathbb{R}^d \to \mathbb{R}$ is uniquely defined by the condition

   $$\forall z \in \Theta_{\text{int}}(x) : p(z) = \begin{cases} 1 & z = y, \\ 0 & z \neq y. \end{cases}$$

   Then,

   $$\alpha^{(-\ell-1)}_{x,y} := p(x). \tag{3.5c}$$

2. $\Omega \subset \mathbb{R}^{d+1}$ is a d-dimensional manifold.

   Then, the points in $\Theta_{\text{int}}(x)$ define a d-dimensional hyperplane E_x in $\mathbb{R}^{d+1}$ in a unique way. However, it may happen that $x \notin E_x$. Let $P^{\perp}_x : \mathbb{R}^{d+1} \to E_x$ be the orthogonal projection of $\mathbb{R}^{d+1}$ onto E_x. For any $y \in \Theta_{\text{int}}(x)$, an affine function $p \in \mathbb{P}_1$ (in the d-dimensional variables of E_x) is uniquely determined by

   $$\forall z \in \Theta_{\text{int}}(x) : p(z) = \begin{cases} 1 & z = y, \\ 0 & z \neq y. \end{cases}$$

   In this case, we set

   $$\alpha^{(-\ell-1)}_{x,y} := p\left(P^{\perp}_x(x)\right). \tag{3.5d}$$

We have now all ingredients at hand for the definition of composite finite element spaces based on graph coarsening.

We briefly recall the setting.

i. A sequence of mesh points $(\Theta_\ell)_{\ell=-L}^{0}$ is generated by one of the algorithms in Subsection 3.1.

ii. The coefficients $\left(\alpha^{(\ell)}\right)_{\ell=-L}^{0}$ are given as in (3.5a), (3.5b), and (3.5c) resp. (3.5d).

iii. For any fine grid point $x \in \Theta_{-\ell}$, let $\Theta_{\text{int}}(x)$ by defined by algorithm **choice_of_interpolation_points**. Vice versa, for any coarse grid point $x \in \Theta_{-\ell-1}$, the dual set $\Theta_{\text{int}}^{\star}(x) \subset \Theta_{-\ell}$ is defined by

$$\Theta_{\text{int}}^{\star}(x) := \{y \in \Theta_{-\ell} : x \in \Theta_{\text{int}}(y)\}.$$

iv. The standard nodal basis functions on the given mesh $\mathcal{G}_0$ are denoted by $(\varphi_{0,x})_{x\in\Theta_0}$.

Definition 3.7 *Let (i)-(iv) be satisfied. The composite finite element basis functions are given by*

$$\forall y \in \Theta_{-\ell-1}: \quad \varphi_{-\ell-1,y} := \sum_{x\in\Theta_{\text{int}}^{\star}(y)} \alpha_{x,y}^{(-\ell-1)} \varphi_{-\ell,x}$$

and the corresponding composite finite element spaces based on graph coarsening by

$$S_{-\ell} := \operatorname{span}\{\varphi_{-\ell,x} : x \in \Theta_{-\ell}\}.$$

Remark 3.8 *The spaces $(S_\ell)_{\ell=-L}^{0}$ are nested. The representation of the trivial injection $\iota : S_{-\ell-1} \hookrightarrow S_{-\ell}$ with respect to the basis $(\varphi_{-\ell-1,x})_{x\in\Theta_{-\ell-1}}$ and $(\varphi_{-\ell,x})_{x\in\Theta_{-\ell}}$ is the matrix*

$$\mathbf{p}_{-\ell,-\ell-1} : \mathbb{R}^{\Theta_{-\ell-1}} \to \mathbb{R}^{\Theta_{-\ell}},$$
$$\mathbf{p}_{-\ell,-\ell-1}(x,y) := \alpha_{x,y}^{(-\ell-1)}.$$

3.2.3 Realisation of the Function check_quality

In this section, we will describe some (heuristic) strategies to measure the quality of a set of interpolation nodes with respect to a point evaluation in x. Let $x \in \mathbb{R}^d$ and $A = \{z_i : 0 \le i \le d\}$ be a set of points. The function **check_quality** will be a combination of the following three criteria.

Criterion 1: The distortion of the set A is measured by the function $c_2(A)$. For its definition we distinguish between two cases.

1. Ω is a d-dimensional Euclidean domain. Then,

$$c_1(x,A) := \frac{(\operatorname{diam} A)^d}{\operatorname{vol}(A)},$$

where $\operatorname{vol}(A)$ is the volume of $\operatorname{conv}(A)$.

2. If Ω is the surface of a $(d+1)$-dimensional bounded Euclidean domain, then

$$c_1(x,A) := \frac{(\operatorname{diam} A)^d}{\operatorname{vol}(A)},$$

where $\operatorname{vol}(A)$ is the d-dimensional surface volume of $(\operatorname{conv} A) \cap \Omega$ and, in this case,

$$\operatorname{diam} A := \max\{\operatorname{dist}_{\text{geo}}(z_i, z_j) : 0 \le i < j \le d\},$$

where $\operatorname{dist}_{\text{geo}}$ is the geodetic distance of two surface points.

Criterion 2:

1. If Ω is a d-dimensional Euclidean domain, we set $c_2(x,A) := 0$.

2. If Ω is a d-dimensional manifold and E_A is the hyperplane through the points in A, then

$$c_2(x, A) := \frac{\left\| x - P_A^{\perp} x \right\|}{(\operatorname{diam} A)^2},$$

where $P_A^{\perp}$ is as in (3.5d).

Criterion 3:
Finally, we measure the distance of the point x from the simplex which is spanned by the vertices of A :

$$c_3(x, A) := \frac{\operatorname{dist}(x, \operatorname{conv} A)}{\operatorname{diam} A}.$$

4 Numerical Tests for Choosing the Control Parameters in the Algorithms

The algorithms for coarsening finite element spaces depend on various heuristic criteria and related control parameters and we have performed numerical experiments in order to find appropriate choices of such parameters.

Problem 4.1 *Let $\mathbb{S}$ be the unit sphere in $\mathbb{R}^3$ and $f : \mathbb{S} \to \mathbb{R}$ be the function $f(x) := \log \|x - x_0\|$ with $x_0 := (2, 2, 0)^{\intercal}$. Let S_ℓ be the (abstract) finite element space and let f_ℓ denote the L^2-projection of f onto S_ℓ. Then, we study the error*

$$\|f - f_\ell\|_{L^2(\mathbb{S})}$$

depending on the control parameters for the definition of S_ℓ.

4.1 Influence of the Constants c_1, c_2, c_3

The goal of the first experiment is to find an appropriate weighting of the constants c_1, c_2, c_3 for the definition of the function **check_quality**. We have used Algorithm 3.2 to select the coarse nodal points. The given fine mesh consists of 65538 nodal points. In the following table "$\sharp$ dof" denotes the number of nodal points. The values in the table denote the L^2-error of the best approximation for Problem 4.1. The notation x_{-m} is short for $x \times 10^{-m}$.

$\sharp$ dof	c_1	c_2	c_3	$c_1 + c_2$	$c_1 + c_3$	$c_2 + c_3$	$c_1 + c_2 + c_3$
65538	2.58_{-6}	2.58_{-6}	2.58_{-6}	2.58_{-6}	2.58_{-6}	2.58_{-6}	2.58_{-6}
15203	2.08_{-5}	2.23_{-5}	2.01_{-5}	2.09_{-5}	2.01_{-5}	2.31_{-5}	2.01_{-5}
3044	1.23_{-4}	1.40_{-4}	1.32_{-4}	1.24_{-4}	1.20_{-4}	1.26_{-4}	1.21_{-4}
560	7.67_{-4}	8.41_{-4}	8.51_{-4}	7.71_{-4}	7.40_{-4}	7.34_{-4}	7.41_{-4}
94	5.26_{-3}	5.40_{-3}	4.95_{-3}	5.23_{-3}	5.11_{-3}	4.69_{-3}	5.12_{-3}

Based on these results, we have used, in the sequel, for all experiments the choice

$$\textbf{check\_quality}(x, A) := c_2(x, A) + c_3(x, A).$$

4.2 Comparison of the Coarsening Strategies

The goal of the next experiment is the comparison of the basic Algorithm 3.2 with the advanced version, i.e., Algorithm 3.3. The quantity "rel_sparsity" is the average number of non-zero entries per matrix line, more precisely, if N_{tot} denotes the total number of non-zero elements in the system matrix, then rel_sparsity:= $N_{\text{tot}}/\sharp$ dof. The quantity "error" denotes the L^2-error for Problem 4.1. In order to compare the convergence rates we compute the exponent s in $C_1 (\sharp \text{dof})^{-s}$ by using the errors of two consecutive levels

$$\text{rate} := \frac{\log \frac{e_{-\ell}}{e_{-\ell+1}}}{\log \frac{n_{-\ell+1}}{n_{-\ell}}}.$$

Remark 4.2 *We have chosen the two-dimensional (very regular) surface of the unit ball as our test example because we expect, for the standard case of regularly refined meshes and linear finite elements, optimal convergence rates, i.e., $s = 1$. Since our coarsening algorithm does not ensure that locally linear functions are always represented on the coarse mesh exactly, the comparison with these optimal convergence rates is the hardest test for our coarsening algorithm.*

Strategy 1				Strategy 2			
# dof	rel_spars	error	rate	# dof	rel_spars	error	rate
65538	7.0	2.58_{-6}		65538	7.0	2.58_{-6}	
15203	14.9	2.31_{-5}	1.50	21846	13.0	9.28_{-6}	1.16
3044	21.9	1.26_{-4}	1.06	5333	18.3	5.80_{-5}	1.30
560	25.6	7.34_{-4}	1.03	634	24.1	5.08_{-4}	1.02
94	26.1	4.69_{-3}	1.03	89	25.6	3.75_{-3}	1.02

We clearly see the improvement of the advanced selection strategy 2. The L^2-error is smaller compared to the error of the simple (random) selection strategy. Note that the convergence rates are close to the linear convergence rates for regularly refined simplicial meshes (cf. Remark 4.2).

4.3 Selection of Coarse Graph Edges

The definition of the set of edges in the coarse graph (cf. (3.1)) depends on the parameter η which determines the distance of coarse graph points which will be connected. The following table compares the choices $\eta = 2, 3$. We have used the advanced algorithm for the selection of coarse graph points.

$\eta = 2$				$\eta = 3$			
♯dof	rel_sparsity	error	rate	♯dof	rel_sparsity	error	
65538	7.0	2.58_{-6}	–	65538	7.0	2.58_{-6}	–
21846	13.0	9.28_{-6}	1.16	21846	13.0	9.28_{-6}	1.16
7186	19.0	2.53_{-5}	0.90	5333	18.3	5.80_{-5}	1.30
2329	19.6	8.15_{-5}	1.04	634	24.1	5.08_{-4}	1.02
721	21.1	3.10_{-4}	1.11	89	25.6	3.75_{-3}	1.02

The errors in both cases are of similar size. By taking into account the different number of nodal points it turns out that the errors are slightly better in the case $\eta = 2$. Since also the sparsity of the matrix is slightly better compared to $\eta = 3$, we have fixed the choice $\eta = 2$ in the sequel.

4.4 Sparsity of the System Matrix

We have performed a numerical experiment to get insight how the sparsity of the mass matrix behaves on increasingly coarser levels. Let

$$M_{-\ell} := \Big((\varphi_{-\ell,x}, \varphi_{-\ell,y})_{L^2(\Omega)} \Big)_{x,y \in \Theta_{-\ell}} .$$

For this experiment, we have chosen a given fine mesh which contains 262146 nodal points.

♯dof	rel_sparsity
262146	7.0
60765	15.1
12208	22.1
2146	25.6
363	27.0
57	24.5

The numbers of the relative sparsity indicates that the average number of non-zero entries per row is about 25 and stays bounded with increasing number of coarsening steps. This is in accordance with the relative sparsity for the mesh which contained 65538 nodal points (cf. Subsections 4.2, 4.3). For related situations, the sparsity of coarsened system matrices has been investigated in [11, Note 3.7.1, Exercise 3.9.6] and [12, Theorem 6].

5 Application to a Hypersingular Integral Equation

Let $\Omega \subset \mathbb{R}^3$ be a bounded Lipschitz domain with boundary Γ. The normal field n exists almost everywhere on Γ and is oriented toward the unbounded exterior domain. Consider the interior Neumann problem: Find u such that

$$\begin{array}{cc} \Delta u = 0 & \text{in } \Omega, \\ \partial u/\partial n = g_N & \text{on } \Gamma, \\ |u(x)| \leq C \|x\|^{-1} & \text{as } \|x\| \to \infty. \end{array}$$

The unknown Dirichlet data $u_D := u|_\Gamma$ satisfy the boundary integral equation

$$Wu_D = \left(\frac{1}{2}I - K'\right) g_N, \tag{5.1}$$

where W is the hypersingular integral operator

$$W\varphi(x) = -f.p. \int_\Gamma \varphi(y) \frac{\partial^2}{\partial n_y \partial n_x} \frac{1}{4\pi \|x-y\|} ds_y \qquad \text{for } x \in \Gamma \text{ a.e.}$$

Here, $\partial/\partial n_y$ (resp. $\partial/\partial n_x$) denotes the normal derivative with respect to y (resp. x). This integral exists only in the sense of a finite part integral which we abbreviate by $f.p. \int$. The operator K' is the adjoint of the classical double layer operator and given by

$$K'\varphi(x) = \frac{\partial}{\partial n_x} \int_\Gamma \frac{\varphi(y)}{4\pi \|x-y\|} ds_y \qquad \text{for } x \in \Gamma \text{ a.e.}$$

It is well known that the variational form of equation (5.1) is well posed in the quotient space $H^{1/2}(\Gamma)/\mathbb{R}$.

5.1 Convergence of Galerkin Boundary Element Solution

For the discretisation we employ the Galerkin Boundary Element Method by using the composite boundary element spaces $(S_\ell)_{\ell=-L}^0$. In order to avoid the discretisation of the quotient space we introduce a Lagrange multiplier λ and consider the system of linear equation

$$\begin{pmatrix} \mathbf{W}_{-\ell} & \mathbf{1}_{-\ell} \\ \mathbf{1}_{-\ell}^\intercal & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{u}_{-\ell} \\ \lambda \end{pmatrix} = \begin{pmatrix} \left(\frac{1}{2}\mathbf{M}_{-\ell} - \mathbf{K}'_{-\ell}\right) \mathbf{g}_N \\ \mu \end{pmatrix} \tag{5.2}$$

for some $\mu \in \mathbb{R}$. The matrices $\mathbf{W}_{-\ell}$, $\mathbf{M}_{-\ell}$, $\mathbf{K}'_{-\ell}$ are the basis representations of the operators W, I, K' in their variational form with respect to the basis of $S_{-\ell}$. The components of the vector $\mathbf{1}_{-\ell}$ have the constant value 1. We have chosen $\mu = 0$. The vector $\mathbf{g}_N$ is the basis representation of the L^2-orthogonal projection of g_N onto $S_{-\ell}$.

We have considered Γ to be the surface of the unit sphere in $\mathbb{R}^3$ and chosen g_N such that

$$u_D(x) = \frac{1}{\|x - x_0\|}$$

is the exact solution of (5.1).

We have chosen $x_0 = (1.1, 1.1, 0)^\intercal$ as an example where u_D has a near-singularity at $(1,1,0)^\intercal$. The other choice, $x_0 = (2,2,0)^\intercal$, corresponds to a very regular solution.

In the following tables, we have depicted the discrete energy error

$$e_{H^{1/2}}(-\ell) := \left\| u_{-\ell}^{\text{int}} - \tilde{u}_{-\ell} \right\|_{H^{1/2}},$$

where $u_{-\ell}^{\text{int}}$ is the nodal interpolant of the exact solution u_D. The function $\tilde{u}_{-\ell}$ is given by $\tilde{u}_{-\ell} = u_{-\ell} + c$, where $u_{-\ell} = \sum_{x \in \Theta_{-\ell}} (\mathbf{u}_{-\ell})_x \varphi_{-\ell,x}$ is the composite boundary element function corresponding to the solution vector $\mathbf{u}_{-\ell}$ in (5.2) and the constant c is characterised by

$$\left\| u_{-\ell}^{\text{int}} - u_{-\ell} - c \right\|_{L^2} = \min_{\mu \in \mathbb{R}} \left\| u_{-\ell}^{\text{int}} - u_{-\ell} - \mu \right\|_{L^2}.$$

The corresponding L^2-error is given by

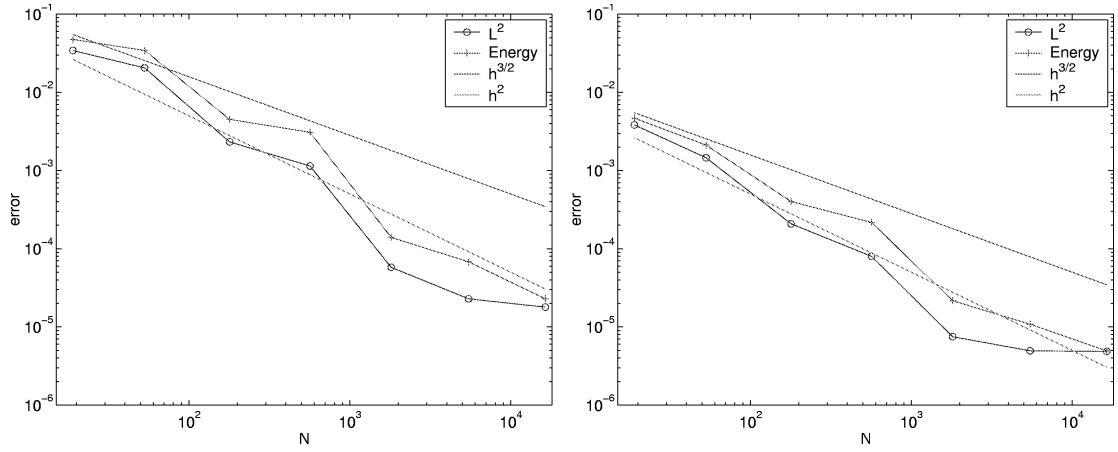
$$e_{L^2}(-\ell) := \left\| u_{-\ell}^{\text{int}} - \tilde{u}_{-\ell} \right\|_{L^2}.$$

The control parameters for our algorithm are chosen as explained in Section 4.
The following table contains the errors on different coarsening levels.

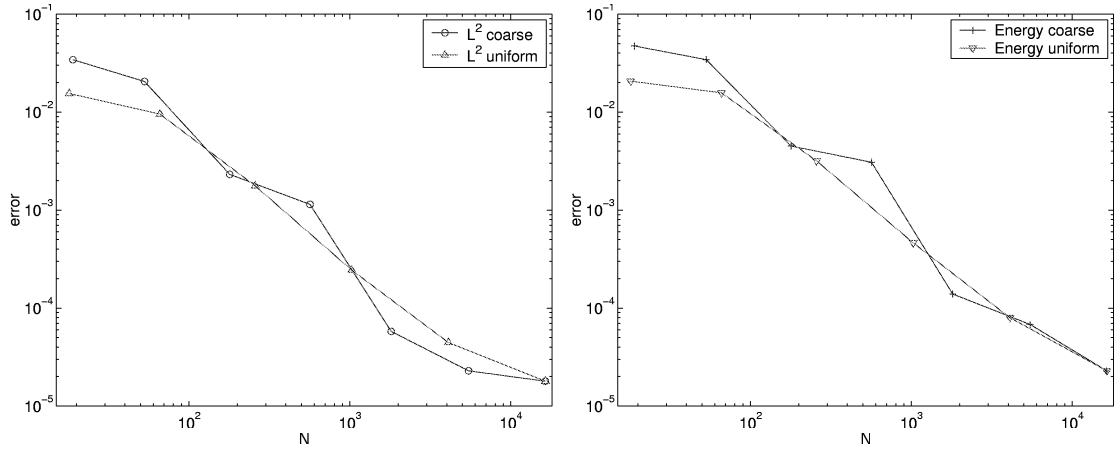
$\sharp$dof	$x_0 = (1.1, 1.1, 0)^\intercal$				$x_0 = (2, 2, 0)^\intercal$			
	e_{L^2}	rate_{L^2}	$e_{H^{1/2}}$	$\text{rate}_{H^{1/2}}$	e_{L^2}	rate_{L^2}	$e_{H^{1/2}}$	$\text{rate}_{H^{1/2}}$
16386	1.79_{-5}		2.29_{-5}		4.88_{-6}		4.97_{-6}	
5462	2.29_{-5}	0.22	6.81_{-5}	0.99	4.97_{-6}	0.016	1.09_{-5}	0.96
1805	5.84_{-5}	0.84	1.40_{-4}	0.65	8.55_{-6}	0.49	2.54_{-5}	0.79
565	1.82_{-3}	2.96	5.01_{-3}	3.08	1.21_{-4}	2.28	3.33_{-4}	2.21
179	3.15_{-3}	0.48	6.00_{-3}	0.16	2.95_{-4}	0.78	5.71_{-4}	0.47
53	2.27_{-2}	1.62	3.84_{-2}	1.53	1.53_{-3}	1.35	2.25_{-3}	1.13
19	4.52_{-2}	0.67	6.30_{-2}	0.48	5.98_{-3}	1.32	7.97_{-3}	1.23

In all cases, the errors increase with increasing coarsening levels in a reasonable way. As expected the L^2-errors are smaller than the $H^{1/2}$-errors and the errors for the problem with the nearly singular solution are larger compared to the very regular solution. Note that *small* values of "rate(ℓ)" indicate that the error is not increased due to the coarsening process and, hence, small values are better than larger values.

The following figures illustrate the convergence behaviour with respect to the L^2 and $H^{1/2}/\mathbb{R}$ norm. The left figure corresponds to the less regular case $x_0 = (1.1, 1.1, 0)^\intercal$ and the right figure to the very regular case $x_0 = (2, 2, 0)^\intercal$. In the case of linear boundary elements on regularly refined simplicial meshes, we expect a convergence rate of order 2 for the L^2-norm and a convergence rate of 3/2 for the $H^{1/2}/\mathbb{R}$ norm. Hence, we have included in the figures the functions h^2 and $h^{3/2}$ for a comparison.



Finally, we compare the convergence rates for our graph coarsening algorithm with the convergence rates on a standard hierarchy of regularly refined simplicial boundary element meshes for the less regular case of $x_0 = (1.1, 1.1, 0)^\intercal$. We see that the convergence behaviour is quite comparable with the convergence rates obtained by regularly refined meshes.



6 Convergence Rate of a Multigrid Method for the Hypersingular Equation

The sequence of coarsened boundary element spaces can be used straightforwardly for solving the Galerkin discretisation on the finest mesh via multigrid methods. We assume here that the reader is familiar with multigrid methods and refer for a general reference, e.g., to [11]. We have chosen the ingredients for the multigrid algorithm as follows:

1. **Smoother:** Damped Richardson iteration.
2. **Prolongation:** Representation of trivial injection $\iota_{-\ell,-\ell-1} : S_{-\ell-1} \hookrightarrow S_{-\ell}$ with respect to the basis $(\varphi_{-\ell-1,x})_{x\in\Theta_{-\ell-1}}$. The matrix representation of $\iota_{-\ell,-\ell-1}$ is the (sparse) prolongation matrix $\mathbf{p}_{-\ell,-\ell-1}$ as in (3.8).
3. **Coarse level system matrix:** The coarse level system matrix is obtained by means of the Galerkin product via the recursion
$$\mathbf{W}_{-\ell-1} := \mathbf{p}^{\mathrm{T}}_{-\ell,-\ell-1}\mathbf{W}_{-\ell}\,\mathbf{p}_{-\ell-\ell-1}.$$
4. **Choice of multigrid cycle:** V- and W-cycle multigrid. We have always chosen the same number of pre- and post-smoothing steps. (In the tables below, this number is denoted by "♯ smooth. steps".)
5. **Stopping criterion:** The discrete L^2-norm of the residual is smaller than 10^{-8}. (The residual after the i-th iteration is denoted by R_i.)

We have applied the multigrid method to the discretisation of the hypersingular integral equation on different surfaces.

Example 6.1 *In this case, the hypersingular boundary integral equation is considered on the surface of the three-dimensional unit sphere. We have chosen the V-cycle multigrid method with one pre- and one post-smoothing step.*

The following table shows the small (about 0.15*) and stable convergence rates which, practically, do not deteriorate as the number of coarsening levels increases.*

	$N = 258$, *4 levels*		$N = 1026$, *5 levels*		$N = 16386$, *7 levels*	
♯ *it.*	$\|R_i\|$	$\frac{\|R_{i+1}\|}{\|R_i\|}$	$\|R_i\|$	$\frac{\|R_{i+1}\|}{\|R_i\|}$	$\|R_i\|$	$\frac{\|R_{i+1}\|}{\|R_i\|}$
0	1.870_{-1}		9.696_{-2}		2.452_{-2}	
1	4.657_{-3}	0.02	1.945_{-3}	0.02	3.032_{-4}	0.01
2	4.248_{-4}	0.09	2.089_{-4}	0.11	1.995_{-5}	0.07
3	4.570_{-5}	0.11	2.847_{-5}	0.14	3.020_{-6}	0.15
4	5.438_{-6}	0.12	4.040_{-6}	0.14	4.868_{-7}	0.16
5	6.928_{-7}	0.13	5.889_{-7}	0.15	7.992_{-8}	0.16
6	9.207_{-8}	0.13	8.801_{-8}	0.15	1.326_{-8}	0.17
7	1.254_{-8}	0.14	1.350_{-8}	0.15	2.218_{-9}	0.17
8	1.732_{-9}	0.14	2.125_{-9}	0.16		

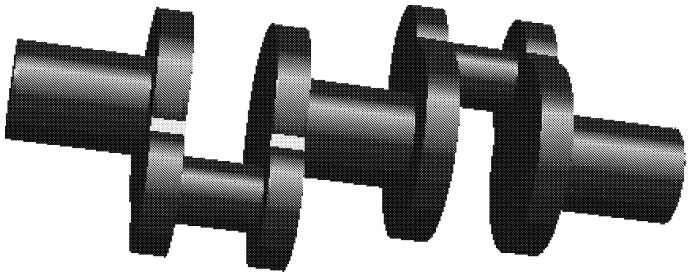


Figure 2: Crankshaft

Example 6.2 *In this case, the hypersingular boundary integral equation is solved on the surface of the crankshaft which is depicted in Figure*[2] *2. The finest mesh for the multigrid method contains 14146 nodes. The coarsening is applied four times leading to graphs with 3347, 693, 130, and 19 nodes. We have tested different numbers of pre- and post-smoothing steps and the V- and the W-cycle multigrid method.*

	# smooth. steps 1		# smooth. steps 3		# smooth. steps 5	
# it	$\|R_i\|$	$\frac{\|R_{i+1}\|}{\|R_i\|}$	$\|R_i\|$	$\frac{\|R_{i+1}\|}{\|R_i\|}$	$\|R_i\|$	$\frac{\|R_{i+1}\|}{\|R_i\|}$
1	1.170279e-03	0.18	3.871218e-04	0.06	2.114952e-04	0.03
5	6.158399e-05	0.53	5.911402e-06	0.40	2.014413e-06	0.32
10	4.682441e-06	0.63	6.136114e-08	0.41	6.824922e-09	0.32
13	1.269176e-06	0.66	4.103323e-09	0.41		
15	5.552864e-07	0.66				
26	7.753820e-09	0.69				

Table 1: Multigrid on the crankshaft: V-cycle.

	# smooth. steps 1		# smooth. steps 2		# smooth. steps 3	
# it	$\|R_i\|$	$\frac{\|R_{i+1}\|}{\|R_i\|}$	$\|R_i\|$	$\frac{\|R_{i+1}\|}{\|R_i\|}$	$\|R_i\|$	$\frac{\|R_{i+1}\|}{\|R_i\|}$
1	2.201703e-04	0.03	1.480443e-04	0.02	1.130168e-04	0.02
5	6.456771e-06	0.50	7.205130e-07	0.33	9.140126e-08	0.21
7	1.867704e-06	0.55	8.619274e-08	0.35	4.683665e-09	0.23
10	3.437266e-07	0.57	4.297848e-09	0.38		
17	9.115666e-09	0.60				

Table 2: Multigrid on the crankshaft: W-cycle.

We see that the convergence rates are worse for complicated geometries but, still, bounded properly away from 1. *Based on these results, we recommend to use three pre- and post-smoothing steps (or an advanced smoothing iteration) within a V-cycle multigrid method.*

Example 6.3 *As an example of a very complicated geometry, we considered to a brick of size* $10 \times 10 \times 1$ *with* 100 *holes as depicted in Figure 3.*

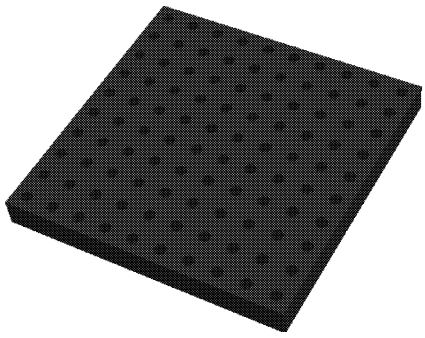


Figure 3: Brick with 100 holes

The finest mesh contains 21430 nodes and the coarsening algorithm has been applied four times, leading to graphs with 5129, 997, 151, and 13 nodes. Again, we have tested the multigrid algorithm with different numbers of pre- and post-smoothing steps and the V- and W-cycle multigrid method. The results are depicted in the Tables 3 and 4.

This example shows that for this very complicated geometry the reduction of the highly oscillatory parts of the iteration error require an increased number of smoothing steps. However, the convergence rates are still

[2]The figures of the crankshaft and the brick with many holes (Example 6.3) have been generated by the program *NETGEN* by J. Schöberl (see [23]).

	# smooth.steps=5		# smooth.steps=8		# smooth.steps=11	
# it	$\|R_i\|$	$\frac{\|R_{i+1}\|}{\|R_i\|}$	$\|R_i\|$	$\frac{\|R_{i+1}\|}{\|R_i\|}$	$\|R_i\|$	$\frac{\|R_{i+1}\|}{\|R_i\|}$
5	2.469671e-03	0.84	1.584542e-03	0.78	1.076976e-03	0.74
10	1.148291e-03	0.87	5.582200e-04	0.82	2.949363e-04	0.78
20	3.298951e-04	0.90	9.512725e-05	0.84	2.938303e-05	0.80
50	1.288089e-05	0.90	6.722382e-07	0.85	3.711182e-08	0.80
56	6.867950e-06	0.90	2.522530e-07	0.85	9.821014e-09	0.80
76	8.538130e-07	0.90	9.649817e-09	0.85		
100	7.062925e-08	0.90				

Table 3: Multigrid on the surface of the perforated brick: V-cycle

	# smooth. steps 1		# smooth. steps 4		# smooth. steps 7	
# it	$\|R_i\|$	$\frac{\|R_{i+1}\|}{\|R_i\|}$	$\|R_i\|$	$\frac{\|R_{i+1}\|}{\|R_i\|}$	$\|R_i\|$	$\frac{\|R_{i+1}\|}{\|R_i\|}$
5	4.889144e-03	0.85	1.540067e-03	0.78	7.022789e-04	0.70
10	2.883596e-03	0.92	5.187998e-04	0.82	1.361742e-04	0.73
20	1.479629e-03	0.94	8.037822e-05	0.84	7.014665e-06	0.75
43	4.174018e-04	0.95	1.578873e-06	0.85	9.459614e-09	0.75
50	2.937840e-04	0.95	4.884850e-07	0.85		
74	9.398491e-05	0.95	8.928938e-09	0.84		
100	2.904106e-05	0.96				

Table 4: Multigrid on the surface of the perforated brick: W-cycle

bounded away from one. For example, 50 V-cycle iterations are necessary (with eight pre- and post-smoothing steps) to reduce the discrete L^2-norm of the residual below 10^{-6}.

References

[1] R. Bank and R. Smith. An algebraic multilevel multigraph algorithm. *SIAM J. Sci. Comp.*, 23:1572–1592, 2002.

[2] R. Bank and J. Xu. An algorithm for coarsening unstructured meshes. *Numer. Math.*, 73:1–36, 1996.

[3] R. E. Bank and J. Xu. A hierarchical basis multi-grid method for unstructured grids. In W. Hackbusch and G. Wittum, editors, *Fast Solvers for Flow Problems, Proceedings of the Tenth GAMM-Seminar, Kiel.* Verlag Vieweg, 1995.

[4] D. Braess. Towards algebraic multigrid for elliptic problems of second order. *Computing*, 55:379–393, 1995.

[5] J. Bramble. *Multigrid Methods.* Pitman Research Notes in Mathematics. Longman Scientific & Technical, 1993.

[6] J. Bramble, Z. Leyk, and J. Pasciak. The analysis of multigrid algorithms for pseudo-differential operators of order minus one. *Math. Comp.*, 63:461–478, 1994.

[7] J. Bramble, J. Pasciak, and P. Vassilevksi. Computational scales of Sobolev norms with applications to preconditioning. *Math. Comp.*, 69:462–480, 1999.

[8] T. Chan, J. Xu, and L. Zikatanov. An agglomeration multigrid method for unstructured grids. *Contemp. Math.*, 218:67–81, 1998.

[9] T. F. Chan and B. F. Smith. Domain decomposition and multigrid algorithms for elliptic problems on unstructured meshes. *ETNA*, 2:171–182, 1994.

[10] D. Feuchter, I. Heppner, S. Sauter, and G. Wittum. Bridging the gap between geometric and algebraic multi-grid methods. *Comp. and Vis. in Sci.*, 6(1):1–13, 2003.

[11] W. Hackbusch. *Multi-Grid Methods and Applications*. Springer-Verlag, Berlin, 1985, 2nd edition 2003.

[12] W. Hackbusch and S. Sauter. Composite finite elements for problems containing small geometric details. Part II: Implementation and Numerical Results. *Comput. Visual. Sci.*, 1:15–25, 1997.

[13] W. Hackbusch and S. Sauter. Composite finite elements for the approximation of pdes on domains with complicated micro-structures. *Numer. Math.*, 75:447–472, 1997.

[14] R. Kornhuber and H. Yserentant. Multilevel methods for elliptic problems on domains not resolved by the coarse grid. *Contemporay Mathematics*, 180:49–60, 1994.

[15] U. Langer and D. Pusch. Data-sparse algebraic multigrid methods for large scale boundary element equation. *Appl. Numer. Math.*, 2004 (online version).

[16] U. Langer, D. Pusch, and S. Reitzinger. Efficient preconditioners for boundary element matrices based on grey-box algebraic multigrid methods. *Int. J. Numer. Meth. Eng.*, 58:1937–1953, 2003.

[17] J. Mandel, M. Brezina, and P. Vaněk. Energy optimization of algebraic multigrid bases. *Computing*, 62:205–228, 1999.

[18] M. Rech, S. Sauter, and A. Smolianski. Two-scale composite finite element method for the Dirichlet problem on complicated domains). Technical report no. 17-2003, Universität Zürich, 2003.

[19] A. Reusken. A multigrid method based on incomplete Gaussian elimination. *Num. Lin. Alg. Appl.*, 3:369–390, 1996.

[20] J. Ruge and K. Stüben. Algebraic multigrid. In S. McCormick, editor, *Multigrid Methods*, pp 73–130, SIAM Philadelphia, 1987.

[21] S. Sauter. *Vergröberung von Finite-Elemente-Räumen*, Habilitationsschrift. Universität Kiel, Germany, 1997.

[22] G. Schmidlin and C. Schwab. Wavelet Agglomeration on unstructured meshes. In T. Barth, T. Chan, and R. Haimes, editors, *Lecture Notes in Computational Science and Engineering*, volume 20, pp 359–378. Springer-Verlag, Heidelberg, 2002.

[23] J. Schöberl. NETGEN - An advancing front 2D/3D-mesh generator based on abstract rules. *Comput. Visual. Sci.*, 1:41–52, 1997.

[24] J. Tausch and J. White. Multiscale bases for the sparse representation of boundary integral operators on complex geometry. *SIAM J. Sci Comp.*, 25:1610–1629, 2003.

[25] P. Vaněk, J. Mandel, and M. Brezina. Algebraic multigrid by smoothed aggregation for second and fourth order elliptic problems. *Computing*, 56:179–196, 1996.

[26] J. Xu. The auxiliary space method and optimal multigrid preconditioning techniques for unstructured grids. *Computing*, 56:215–235, 1996.

[27] H. Yserentant. Coarse grid spaces for domains with a complicated boundary. *Numerical Algorithm*, 21:387–392, 1999.