Fast cluster techniques for BEM

Krzebek, N ; Sauter, S

Abstract: In this paper, we will present a new approach for solving boundary integral equations with panel clustering. In contrast to all former versions of panel clustering, the computational and storage complexity of the algorithm scales linearly with respect to the number of degrees of freedom without any additional logarithmic factors. The idea is to develop alternative formulations of all classical boundary integral operators for the Laplace problem where the kernel function has a reduced singular behaviour. It turns out that the application of the panel-clustering method with variable approximation order preserves the asymptotic convergence rate of the discretisation and has significantly reduced complexity.

DOI: https://doi.org/10.1016/S0955-7997(02)00155-8

Fast Cluster Techniques for BEM

Nico Krzebek, Stefan Sauter
Universität Zürich
Institut für Mathematik
Winterthurerstrasse 190
CH-Zürich

Febuary 2002

Abstract

In this paper, we will present a new approach for solving boundary integral equations with panel-clustering. In contrast to all former versions of panel clustering, the computational and storage complexity of the algorithm scales linearly with respect to the number of degrees of freedom without any additional logarithmic factors. The idea is to develop alternative formulations of all classical boundary integral operators for the Laplace problem where the kernel function has a reduced singular behaviour. It turns out that the application of the panel-clustering method with variable approximation order preserves the asymptotic convergence rate of the discretisation and has significantly reduced complexity.

Keywords: Boundary integral equations, Galerkin boundary element method, panel-clustering method, alternative representations

1 Introduction

In this paper, we will present some new methods for the numerical solution of elliptic boundary value problems on three-dimensional domains. We employ the method of integral equations to transform the elliptic differential equation into a boundary integral equation on the surface of the domain. Galerkin's method with boundary elements is used for its discretisation.

As a consequence of the non-localness of boundary integral operators, the basis representation of these operators leads to a fully populated coefficient matrix. This fact was considered as the major drawback of this approach until the panel-clustering representation and the multipole method in the mid eighties were introduced and have reduced the computational and storage complexity of the boundary integral

This work was partially supported by the Swiss National Science Foundation under grants 21-61764.00 and 21-63944.00.

equation from $O\left(N^2\right)$ to $O\left(N \log^\kappa N\right)$, where N denotes the number of unknowns and $\kappa \approx 4-7$. The asymptotic gain of the method is obvious as, for $N \to \infty$, the term $\log^\kappa N$ becomes negligible compared to N. However, in practical applications the number of unknowns, typically, ranges from 5×10^3 to 10^5 and, hence, the constant in the $O(\cdot)$ estimate and the term $\log^\kappa N$ are not negligible.

The idea of the panel-clustering method is to approximate the kernel function of the integral operator by a degenerate kernel where the coupling of the variables is factorised. The approximation is constructed piecewise on each element (called *block*) of a partitioning of the surface Γ (more precisely of $\Gamma \times \Gamma$) which is graded towards the diagonal where $x = y$. For the largest blocks, the order of the kernel approximation has to be chosen proportionally to $\log N$ to preserve the convergence order of the discretisation. In all versions of the panel-clustering method, the expansion order is chosen constant on all blocks and, since the total number of blocks is $O(N)$, the complexity estimate $O\left(N \log^\kappa N\right)$ results.

In [12], the variable order panel-clustering method was introduced. The idea is to replace the kernel function of the integral operator on small surface blocks by low-order approximations and on larger surface blocks by approximations with increasing order. By choosing the slope of the order distribution as an affine function it turns out that, for the classical double layer potential on smooth surfaces in $\mathbb{R}^3$ discretised by piecewise constant boundary elements, the computational and storage complexity of the method is $O(N)$ without any logarithmic terms.

The variable order panel-clustering method cannot be applied to more general integral operators or higher order boundary elements directly without any loss of accuracy since, in these cases, the (strong) singular behaviour of the kernel function reduces the accuracy of the approximation with variable order significantly.

In this paper, we will overcome this difficulty by developing alternative representations of all classical integral operators corresponding to the Laplace problem. The order of singularity in the kernel function is significantly reduced, and the variable order panel-clustering method can be applied to these new formulations without any loss in the convergence order.

Thus, all integral operators related to the Laplace problem which are discretised with low-order boundary elements can be solved numerically with $O(N)$ computational complexity without any logarithmic terms.

2 The Boundary Element Method

The goal of this paper is to present efficient numerical methods for solving elliptic boundary value problems on a domain in $\mathbb{R}^3$. Since the integral equation method is employed to transform these elliptic partial differential equations into integral equations on the surface of the domain, we restrict here to linear and homogeneous equations with constant coefficients. Hence, the Laplace problem serves as the adequate model problem to develop the new methods. The formulation of these boundary value problems requires the introduction of some notations.

Let $\Omega^- \subset \mathbb{R}^3$ be a bounded, piecewise smooth Lipschitz domain with boundary Γ and $\Omega^+ := \mathbb{R}^3 \setminus \overline{\Omega^-}$. The outer normal vector relative to Ω^- is denoted by n.

The space of continuous functions on Γ is denoted by $C^0(\Gamma)$ and the space of k-times continuously differentiable functions on a domain Ω by $C^k(\Omega)$.

The space of all measurable functions $f\colon \Gamma \to \mathbb{R}$ which are square integrable with respect to the surface measure ds is denoted by $L^2(\Gamma)$. For $0 < s < 3/2$ we define the Sobolev space $H^s(\Gamma)$ as the space of traces on Γ of functions in $H^{s+1/2}_{\text{loc}}(\mathbb{R}^3)$, and $H^{-s}(\Gamma)$ is the dual space of $H^s(\Gamma)$. The scalar product in $L^2(\Gamma)$ is denoted by $\langle .\,,.\rangle_{L^2(\Gamma)}$ and identified with its continuous extension to the dual pairing $\langle .\,,.\rangle_{H^s(\Gamma)\times H^{-s}(\Gamma)}$ and $\langle .\,,.\rangle_{H^{-s}(\Gamma)\times H^s(\Gamma)}$.

2.1 Model problems

In our paper we shall consider the numerical solution of boundary value problems for the Laplace equation. We start with the classical (strong) formulation of these problems.

ID: Given $g_D \in C^0(\Gamma)$, find $u \in C^2(\Omega^-) \cap C^0(\overline{\Omega^-})$ such that

$$\begin{aligned} \Delta u &= 0 && \text{in } \Omega^-, \\ u &= g_D && \text{on } \Gamma. \end{aligned} \tag{1}$$

IN: Given $g_N \in C^0(\Gamma)$, find $u \in C^2(\Omega^-) \cap C^1(\overline{\Omega^-})$ such that

$$\begin{aligned} \Delta u &= 0 && \text{in } \Omega^-. \\ \partial u/\partial n &= g_N && \text{on } \Gamma. \end{aligned} \tag{2}$$

ED: Given $g_D \in C^0(\Gamma)$, find $u \in C^2(\Omega^+) \cap C^0(\overline{\Omega^+})$ such that

$$\begin{aligned} \Delta u &= 0 && \text{in } \Omega^+, \\ u &= g_D && \text{on } \Gamma, \\ |u(x)| &\le C\|x\|^{-1} && \text{as } \|x\| \to \infty. \end{aligned} \tag{3}$$

EN: Given $g_N \in C^0(\Gamma)$, find $u \in C^2(\Omega^+) \cap C^1(\overline{\Omega^+})$ such that

$$\begin{aligned} \Delta u &= 0 && \text{in } \Omega^+. \\ \partial u/\partial n &= g_N && \text{on } \Gamma, \\ |u(x)| &\le C\|x\|^{-1} && \text{as } \|x\| \to \infty. \end{aligned} \tag{4}$$

Remark 1. *Note that, for exterior problems, the decay condition*

$$|u(x)| \le C\,\|x\|^{-1} \quad \text{as} \quad \|x\| \to \infty$$

has to be imposed in order to guarantee uniqueness. For the interior Neumann problem, the right-hand side g_N must satisfy $\int_\Gamma g_N ds = 0$ for existence and the solution $\int_{\Omega^-} u dx = 0$ for uniqueness. For details, we refer to [4].

2.2 Boundary integral equations for the model problems

These boundary value problems can be transformed into boundary integral equations (BIEs) on Γ by using either the direct or the indirect method. We will restrict here to the direct method and refer for the indirect method to [5] and [8]. For the numerical methods presented in this paper it is important to emphasise that the arising integral operators for the indirect method are contained in the set of integral operators corresponding to the direct method.

The fundamental solution of the Laplace operator in $\mathbb{R}^3$ is given by

$$G(z) = \frac{1}{4\pi\|z\|}. \tag{5}$$

The following four bilinear forms are related to this fundamental solution or certain Gâteau derivatives of it.

1. **Single layer potential.**

   The bilinear form $a_V\colon H^{-1/2}(\Gamma) \times H^{-1/2}(\Gamma) \to \mathbb{R}$ related to the classical single layer potential operator is given by

   $$a_V(u,v) := \int_{\Gamma\times\Gamma} v(x)G(x-y)u(y)\,ds_y ds_x. \tag{6}$$

2. **Double layer potential.**

   The bilinear form $a_{K_\pm}\colon L^2(\Gamma) \times L^2(\Gamma) \to \mathbb{R}$ corresponding to the classical double layer potential operator is given by

   $$a_{K_\pm}(u,v) := \pm\frac{1}{2}\int_\Gamma u(x)v(x)\,ds_x + \int_{\Gamma\times\Gamma} v(x)\frac{\partial}{\partial n_y}G(x-y)u(y)\,ds_y ds_x. \tag{7}$$

   The "+"-sign corresponds to the Laplace problem on Ω^+ and the "$-$"-sign to the interior problem.

3. **Adjoint double layer potential.**

   The bilinear form $a_{K'_\pm}\colon L^2(\Gamma) \times L^2(\Gamma) \to \mathbb{R}$ is given by

   $$a_{K'_\pm}(u,v) := \mp\frac{1}{2}\int_\Gamma u(x)v(x)\,ds_x + \int_{\Gamma\times\Gamma} v(x)\frac{\partial}{\partial n_x}G(x-y)u(y)\,ds_y ds_x. \tag{8}$$

4. **Hypersingular integral operator.**

   The bilinear form $a_W\colon H^{1/2}(\Gamma) \times H^{1/2}(\Gamma) \to \mathbb{R}$ associated with the hypersingular integral operator is given by

   $$a_W(u,v) := \int_\Gamma v(x)\frac{\partial}{\partial n_x}\int_\Gamma \frac{\partial}{\partial n_y}G(x-y)u(y)\,ds_y ds_x. \tag{9}$$

By using these operators, the boundary value problems (1)-(4) can be transformed into a variational formulation.

ID: Given $g_D \in H^{1/2}(\Gamma)$, find $u \in H^{-1/2}(\Gamma)$ such that

$$a_V(u,v) = a_{K_+}(g_D, v), \qquad \forall v \in H^{-1/2}(\Gamma).$$

IN: Given $g_N \in H^{-1/2}(\Gamma)$, find $u \in H^{1/2}(\Gamma)$ such that

$$a_W(u,v) = -a_{K'_-}(g_N, v), \qquad \forall v \in H^{1/2}(\Gamma).$$

ED: Given $g_D \in H^1(\Gamma)$, find $u \in L^2(\Gamma)$ such that

$$a_{K'_-}(u,v) = -a_W(g_D, v), \qquad \forall v \in L^2(\Gamma).$$

EN: Given $g_N \in H^{-1/2}(\Gamma)$, find $u \in L^2(\Gamma)$ such that

$$a_{K_-}(u,v) = a_V(g_N, v), \qquad \forall v \in L^2(\Gamma).$$

The transformation of boundary value problems to BIEs is by no means unique. In this paper we focus on the numerical solution of the abstract variational problem

$$a(u,v) = F(v), \qquad \forall v \in H(\Gamma), \tag{10}$$

where a is one of the four bilinear forms a_V, $a_{K_\pm}$, $a_{K'_\pm}$ or a_W. For a complete listing of the arising equations see for instance [5, Section 8.4] or [14]. For existence and uniqueness results we refer to [8], [9], [14].

2.3 Galerkin discretisation

We consider the discretisation of (10) by the Galerkin method. The infinite dimensional Sobolev space $H^s(\Gamma)$ is replaced as usual by a finite dimensional subspace. Let $\mathcal{G} = \{\tau_1, \ldots, \tau_n\}$ be a surface mesh with open and disjoint panels τ_i satisfying $\Gamma = \cup_{\tau \in \mathcal{G}} \overline{\tau}$. The panels are either (possibly curved) triangles or quadrilaterals. More precisely, we assume that, for any $\tau \in \mathcal{G}$, there exists a sufficiently smooth chart $\chi_\tau : Q \to \tau$, where Q is either the unit square in $\mathbb{R}^2$ or the unit triangle (with vertices $(0,0)$, $(1,0)$, $(1,1)$).

The boundary element spaces are defined by lifting polynomial spaces on the reference element Q to the surface mesh. Let $\mathcal{S}^0(Q)$ denote the space of all constant functions on Q and

$$\mathcal{S}^1(Q) := \begin{cases} \operatorname{span}\{1, x_1, x_2\} & \text{if } Q \text{ is the unit triangle,} \\ \operatorname{span}\{1, x_1, x_2, x_1 x_2\} & \text{if } Q \text{ is the unit square.} \end{cases}$$

Let us introduce the discontinuous, piecewise constant and the continuous, piecewise linear boundary element space

$$\begin{aligned}\mathcal{S}^0 &:= \left\{u \in L^\infty(\Gamma) \mid \forall \tau \in \mathcal{G} : u|_\tau \circ \chi_\tau \in \mathcal{S}^0(Q)\right\}, \\ \mathcal{S}^1 &:= \left\{u \in C^0(\Gamma) \mid \forall \tau \in \mathcal{G} : u|_\tau \circ \chi_\tau \in \mathcal{S}^1(Q)\right\}.\end{aligned} \tag{11}$$

We use $\mathcal{S}$ as the general notation if no confusion is possible. Throughout the paper we restrict to conforming subspaces satisfying $\mathcal{S} \subset H$. For the hypersingular operator, this implies $\mathcal{S} = \mathcal{S}^1$, while, for the single and the double layer potential, both choices $\mathcal{S} = \mathcal{S}^0$ and $\mathcal{S} = \mathcal{S}^1$ are possible.

The standard nodal basis functions on $\mathcal{S}$ are denoted by b_i, $i = 1, \ldots, N$, so that the basis representation of a function $u_G \in \mathcal{S}$ is given by

$$u_G(x) = \sum_{i=1}^N u_i b_i(x). \tag{12}$$

The coefficient vector of u_G is denoted by $\mathbf{u} = \{u_i\}_{i=1}^N$. The Galerkin discretisation of (10) is defined by seeking $u_G \in \mathcal{S}$ such that

$$a(u_G, v) = F(v), \qquad \forall v \in \mathcal{S}. \tag{13}$$

Problem (13) is equivalent to solving the system of linear equations

$$\mathbf{A}\mathbf{u} = \mathbf{f}, \tag{14}$$

where $\mathbf{A} \in \mathbb{R}^{N\times N}$ and $\mathbf{f} \in \mathbb{R}^N$ are given by

$$\mathbf{A}_{i,j} := a(b_i, b_j), \quad \mathbf{f}_i := F(b_i).$$

The solutions $\mathbf{u}$ and u_G are related via (12).

From the theoretical point of view, the Galerkin boundary element method is the method of choice since stability and convergence can be proved for a much larger class of integral equations and surfaces than, for instance, the collocation and the Nyström method.

Since the nineties, efficient techniques have been developed for the computation of the coefficients of the Galerkin system matrix and the panel-clustering method has been extended to the Galerkin discretisation [7], [3], [2], [13]. As a consequence, the computational complexity of the Galerkin method has become comparable to the collocation and Nyström method (cf. [3]).

The complexity of the Galerkin method with numerical quadrature and panel clustering is of order $N \log^\kappa N$ with $\kappa \approx 4 - 7$. More precisely, the computation of the coefficients of the *nearfield* matrix (which is related to pairs of panels with small distance) requires $O\left(\log^4 N\right)$ quadrature points per matrix entry and the cost sums up to $O\left(N \log^4 N\right)$ for the nearfield matrix.

Furthermore, the approximation of the kernel function of the integral operator by the panel-clustering method requires $O\left(N \log^\kappa N\right)$ quantities to be stored and $O\left(N \log^\kappa N\right)$ arithmetic operations for their computation.

Although this is a substantial improvement compared to the $O(N^2)$ complexity of classical matrix-oriented methods, the constant in the complexity estimate and the $\log^\kappa(N)$ terms are still large.

We address two issues in this paper. In Section 3 we will present alternative representations of the boundary integral operators ((6)-(9)) that reduce the degree of singularity of the kernel functions. As a consequence, the entries of the nearfield matrix can be computed with $O(1)$ quadrature points per coefficient and, furthermore, the integral kernel can be approximated by the panel-clustering method with $O(N)$ quantities. The resulting fully discrete Galerkin method will be presented in Section 4. Finally, in Section 5 the results of some numerical experiments are reported.

3 Alternative Representations

We will introduce alternative representations of the bilinear forms associated with the integral operators introduced in Section 2. The idea behind these reformulations is to express the kernel functions as derivatives of kernels with reduced singular behaviour and to approximate the antiderivative of these kernels. The regularity of the solution of the BIE can be employed in the error analysis by partial integration yielding an optimal convergence rate.

3.1 Double Layer Potential

Let G be as in (5). It is well known, e.g. [5, Section 8.2.4.3], that the solid angle of Γ with respect to an observation point x is given by

$$\Upsilon_\Gamma(x) := -\int_\Gamma \frac{\partial}{\partial n_y} G(x-y)\, ds_y, \qquad x \in \Gamma. \tag{15}$$

The function Υ is in $L^\infty(\Gamma)$ and equals $1/2$ almost everywhere on Γ. The bilinear form $a_{K_\pm}$ can then be rewritten as

$$\begin{aligned} a_{K_\pm}(u,v) = & \left(\pm\frac{1}{2}+\frac{1}{2}\right)\int_\Gamma u(x)v(x)\, ds_x \\ & + \int_{\Gamma\times\Gamma} v(x)(u(y)-u(x))\frac{\partial}{\partial n_y}G(x-y)\, ds_y ds_x. \end{aligned} \tag{16}$$

Remark 2. *The difference $u(y)-u(x)$ in* (16) *reduces the singular behaviour of the integrand in* (16) *if the solution u has some regularity, i.e. $u \in H^s(\Gamma)$ for some $s > 0$.*

3.2 Hypersingular Operator

In order to rewrite the bilinear form a_W associated with the hypersingular operator, we have to introduce some surface differential operators. For a more detailed description of these operators we refer to [10]. For functions $u \in H^{1/2}(\Gamma)$ and surface vector fields v having componentwise differentiable extensions $\tilde{u}$ and $\tilde{v}$, respectively, in $H^1(\mathcal{U})$, where $\mathcal{U}$ is some three-dimensional neighbourhood of Γ, we define the *tangential gradient* $\nabla_\Gamma u$ and the *surface divergence* $\operatorname{div}_\Gamma v$ as restrictions of their related operators to the surface Γ

$$\nabla_\Gamma u := \nabla \tilde{u}\big|_\Gamma, \quad \operatorname{div}_\Gamma v := \operatorname{div} \tilde{v}\big|_\Gamma.$$

This enables us to introduce the *tangential rotation* of u as

$$\overrightarrow{\operatorname{curl}}_\Gamma u := \operatorname{curl}(\tilde{u}n)\big|_\Gamma = -n \times \nabla_\Gamma u$$

and the *surface rotation* as

$$\operatorname{curl}_\Gamma v := \langle n, \operatorname{curl} \tilde{v}\rangle\big|_\Gamma,$$

where $\langle .\,,.\rangle$ denotes the Euclidian scalar product in $\mathbb{R}^3$. The composition of surface and tangential rotation leads to the Laplace-Beltrami operator

$$\Delta^\Gamma u := -\operatorname{curl}_\Gamma \overrightarrow{\operatorname{curl}}_\Gamma u = \left(\Delta \tilde{u} - \left(\frac{\partial}{\partial n}\right)^2 \tilde{u} \right). \tag{17}$$

Note that an index z in Δ^Γ_z indicates differentiation by the z-variable.

Using these notations, a_W can be rewritten as follows

$$a_W(u,v) = \tag{18}$$
$$\frac{1}{4\pi} \int_{\Gamma\times\Gamma} \left\langle \overrightarrow{\operatorname{curl}}_\Gamma v(x), \overrightarrow{\operatorname{curl}}_\Gamma u(y) \right\rangle \left(\Delta^\Gamma_y \|y-x\| - \frac{\langle n_y, y-x\rangle^2}{\|y-x\|^3} \right) ds_y ds_x,$$

cf. [1].

3.3 Single Layer Potential

To reduce technicalities, we assume that Γ is the surface of a Lipschitz polyhedron with disjoint open plane faces $\Gamma^{(i)}$, $1 \le i \le q$, and

$$\Gamma = \overline{\bigcup_{i=1}^q \Gamma^{(i)}}.$$

Then, a_V can be rewritten as

$$\begin{aligned} a_V(u,v) = & \frac{1}{4\pi} \int_{\Gamma\times\Gamma} v(x)u(y)\Delta^\Gamma_y \|y-x\|\, ds_y ds_x \\ & - \frac{1}{4\pi} \int_{\Gamma\times\Gamma} v(x)(u(y)-u(x)) \frac{\langle n_y, y-x\rangle^2}{\|y-x\|^3}\, ds_y ds_x \\ & - \sum_{i=1}^q \int_{\Gamma^{(i)}} \langle n_y, y-x\rangle \Upsilon^{(i)}_\Gamma(x) v(x) u(x)\, ds_x, \end{aligned} \tag{19}$$

where $\Upsilon^{(i)}_\Gamma$ denotes the spherical angle of $\Gamma^{(i)}$ with respect to $x \in \mathbb{R}^3$, cf. (15). Cf. [1] for details.

Remark 3. *The kernel functions of the transformed bilinear forms, cf.* (16)*,* (18) *and* (19)*, are combinations of the kernel functions*

$$\begin{gathered} k_1(x,y) := \frac{\partial}{\partial n_y} G(x-y), \quad k_2(x,y) := \langle n_y, x-y \rangle \frac{\partial}{\partial n_y} G(x-y), \\ k_3(x,y) := \frac{1}{4\pi} \Delta_y^\Gamma \|x-y\|, \end{gathered} \tag{20}$$

i.e. all kernels are a derivative either of the fundamental solution or of the Euclidian distance $\|x-y\|$*.*

4 The panel-clustering algorithm

The matrix representation (14) of the finite dimensional bilinear form in (10) leads to a coefficient matrix $\mathbf{A}$ which is fully populated so that the complexity of the resulting algorithm is at least $O(N^2)$. The idea of the panel-clustering algorithm is to employ an alternative representation of the discrete problem. A matrix-vector multiplication can be formulated in the panel-clustering representation which is the most time consuming step in any iterative solver like GMRES. In this section we will present the panel-clustering method and give an algorithm for the matrix-vector multiplication.

The procedure of the panel-clustering algorithm consists of three basic phases

1. The panels of the surface mesh $\mathcal{G}$ are clustered hierarchically to larger surface pieces (clusters) yielding the cluster tree T.
2. By employing a suitable admissibility condition, pairs of clusters will be combined to blocks which will define a covering of $\Gamma \times \Gamma$ which is graded towards the diagonal.
3. The near-diagonal part of the discrete integral operator is represented by a sparse matrix and the farfield part by replacing the kernel on blocks by a degenerate kernel and by using an alternative representation of the discrete farfield operator.

4.1 The cluster tree

The panel-clustering algorithm starts with the generation of a *cluster tree* T, which is split into two steps:

- The iterative subdivision of the minimal, axis-parallel bounding box $box(\Gamma)$ containing Γ into congruent sub-boxes resulting in the virtual cluster tree.
- Association of geometric portions (clusters) of Γ to the generated sub-boxes.

The algorithm for the construction of the cluster tree T requires some notations which will be introduced first. As usual in graph theory, the tree T consists of a set of vertices V and a set of oriented edges $E \subset V \times V$. The root of T is denoted by $\operatorname{root}(T)$ and the power set of T by $\mathcal{P}(T)$. In the context of panel clustering, the vertices of the tree are called *clusters*. For every cluster $c \in T$ we introduce the *set of sons*

$$\operatorname{sons}\colon T \to \mathcal{P}(T), \quad \operatorname{sons}(c) := \{\tilde{c} \in T \mid (c, \tilde{c}) \in E\},$$

which is generated during the construction of the cluster tree, cf. Procedure 1. The level of a cluster in T is defined as the mapping

$$\operatorname{level}\colon T \to \mathbb{N}_0, \quad \operatorname{level}(c) := \begin{cases} 0 & \text{for } c = \operatorname{root}(T), \\ \operatorname{level}(\tilde{c}) + 1 & \text{for } c \in \operatorname{sons}(\tilde{c}),\ \tilde{c} \in T. \end{cases} \tag{21}$$

We also denote the *set of leaves* of T by

$$\mathfrak{L} := \{c \in T \mid \operatorname{sons}(c) = \emptyset\}.$$

Furthermore, the minimal and maximal depth of T is

$$d_{\min} := \min\{\operatorname{level}(c) \mid c \in \mathfrak{L}\}, \qquad d_{\max} := \max\{\operatorname{level}(c) \mid c \in \mathfrak{L}\}, \tag{22}$$

and the set of all clusters of level l is given by

$$T_l := \{c \in T \mid \operatorname{level}(c) = l\}.$$

It follows from (21) that $T_0 = \{\operatorname{root}(T)\}$.

The cluster tree T is assembled by a recursive subdivision of $\operatorname{box}(\Gamma)$. As a convention we assume that $\operatorname{box}(\Gamma)$ and all its sub-boxes are closed sets. To stop the recursion, we introduce the function $\texttt{refine}\colon T \times \mathbb{N}_0 \to \{\texttt{true}, \texttt{false}\}$,

$$\texttt{refine}(c, l) := \begin{cases} \texttt{true} & \exists \tau \in \mathcal{G}\colon \\ & \operatorname{area}(\tau \cap c) > 0 \wedge \operatorname{diam}(\tau) \leq 2^{-l} \operatorname{diam}(\operatorname{box}(\Gamma)), \\ \texttt{false} & \text{otherwise.} \end{cases}$$

In other words, the stopping criterion halts the refinement if the diameter of a box is smaller than the size of the inscribed panels.

The cluster tree T is built by initialising $\operatorname{root}(T) := \operatorname{box}(\Gamma)$, $\operatorname{sons}(\operatorname{root}(T)) := \emptyset$ and calling $\texttt{build\_cluster\_tree}(0)$, cf. Procedure 1. We emphasise that Procedure 1 builds only the abstract tree structure while the link of tree nodes to the geometry will be generated afterwards.

In the next step, we link clusters in T to portions of the surface Γ and introduce, in this light, an abstract function $\sigma : T \to \Gamma$. As a short-hand, we write $c^\Gamma := \sigma(c)$.

In our concrete applications, two choices of σ will appear.

Procedure 1 `build_cluster_tree`(l)

```
for $c \in T_l$ do
  if refine($c, l$) = true then  /* Refine c or stop recursion */
    subdivide $c$ into eight congruent sub-boxes $c_i$, $i = 1, \dots, 8$
    for all $i = 1, \dots, 8$ do
      if there is a $\tau$ in $\mathcal{G}$ s.t. area($c_i^\Gamma \cap \tau$) > 0 then
        sons($c$) := sons($c$) $\cup \{c_i\}$, sons($c_i$) := $\emptyset$
    $T_{l+1} := T_{l+1} \cup$ sons($c$)  /* Assemble next level */
if $T_{l+1} \neq \emptyset$ then  /* Descend to next level */
  build_cluster_tree($l + 1$)
```

Remark 4. *Let M_ω denote the centre of mass of a subset $\omega \subset \mathbb{R}^3$. The choice*

$$\sigma_1(c) := \{\tau \in \mathcal{G} \mid M_\tau \in c\}, \tag{23}$$

will be considered in the context of a discontinuous panel-clustering approximation, while

$$\sigma_2(c) := c \cap \Gamma \tag{24}$$

will be used for the construction of a continuous panel-clustering approximation, cf. Section 4.4. We will indicate the concrete choice of σ where necessary.

The (geometric) size of a cluster is defined via the function σ.

Definition 5. *For a given mapping $\sigma \colon T \to \Gamma$, the radius ρ_c of cluster c is*

$$\rho_c := \frac{1}{2}\operatorname{diam}(\operatorname{box}(\sigma(c)))$$

and the cluster box box$\,(c)$ *is the minimal axis-parallel box containing $\sigma\,(c)$.*

4.2 Block decomposition of $\mathbf{\Gamma} \times \mathbf{\Gamma}$

The approximation of the bilinear form a requires the construction of a covering of $\Gamma \times \Gamma$. This will be done by constructing the *block cluster tree* $P \subset T \times T$. Its elements $\mathbf{b} = (c_1, c_2) \in P$ will be called *blocks*. Application of the function σ as in Definition 5 to the components c_1, c_2 of a block yields a covering of $\Gamma \times \Gamma$. We will use the notation $\mathbf{b}^\Gamma := (c_1^\Gamma, c_2^\Gamma)$.

The generation of the block cluster tree P is based on an admissibility condition controlling the relative distance of two clusters.

Definition 6. *Let $\eta \in (0, 1)$. A block $\mathbf{b} = (c_1, c_2) \in P$ is called η-*admissible *if*

$$\max(\rho_{c_1}, \rho_{c_2}) \leq \eta \operatorname{dist}(\operatorname{box}(c_1), \operatorname{box}(c_2)). \tag{25}$$

If it is clear from context, we will write "admissible" short for "η-admissible".

The set of admissible blocks in P is called farfield P_{far} *while the complement $P_{\mathrm{near}} := P \setminus P_{\mathrm{far}}$ is the* nearfield.

Using the notation $\widetilde{\mathrm{sons}}(c) := \mathrm{sons}(c)$ if $c \in T \setminus \mathfrak{L}$ and $\widetilde{\mathrm{sons}}(c) := \{c\}$ otherwise, we can define the *sons of a block* (c_1, c_2) as

$$\mathrm{sons}((c_1, c_2)) := \begin{cases} \widetilde{\mathrm{sons}}(c_1) \times \widetilde{\mathrm{sons}}(c_2) & \text{if } (c_1, c_2) \notin \mathfrak{L} \times \mathfrak{L}, \\ \emptyset & \text{otherwise.} \end{cases} \tag{26}$$

After initialising the nearfield $P_{\mathrm{near}} := \emptyset$ and the farfield $P_{\mathrm{far}} := \emptyset$, Procedure 2 is called by `divide`$((\mathrm{box}(\Gamma), \mathrm{box}(\Gamma)), P_{\mathrm{near}}, P_{\mathrm{far}})$. The surface portions $\mathbf{b}^\Gamma$, $\mathbf{b} \in P$, then form a minimal, disjoint covering of $\Gamma \times \Gamma$.

Procedure 2 `divide`$((c_1, c_2), P_{\mathrm{near}}, P_{\mathrm{far}})$

```
if $(c_1, c_2)$ is admissible then
  $P_{\mathrm{far}} := P_{\mathrm{far}} \cup \{(c_1, c_2)\}$
else if $(c_1, c_2) \in \mathfrak{L} \times \mathfrak{L}$ then
  $P_{\mathrm{near}} := P_{\mathrm{near}} \cup \{(c_1, c_2)\}$
else
  for all $(\tilde{c}_1, \tilde{c}_2) \in \mathrm{sons}((c_1, c_2))$ do
    divide$((\tilde{c}_1, \tilde{c}_2), P_{\mathrm{near}}, P_{\mathrm{far}})$
```

4.3 Abstract formulation of the panel-clustering algorithm

4.3.1 Panel-clustering representation of boundary integral operators

The approximation of a boundary integral operator K with kernel function k is based on three assumptions:

1. The approximation is *semi-separable.* On a farfield block $\mathbf{b} = (c_1, c_2)$ we assume the general form

$$k(x, y) \approx \tilde{k}_{\mathbf{b}}(x, y) = \sum_{(\nu,\mu) \in \mathcal{I}_{\mathbf{b}}} \kappa_{\nu,\mu}(\mathbf{b}) \Phi_{c_1}^{(\nu)}(x) \Psi_{c_2}^{(\mu)}(y), \tag{27}$$

   where $\mathcal{I}_{\mathbf{b}} \subset \mathbb{N}_0^d \times \mathbb{N}_0^d$ is a *finite* set of pairs of multi-indices. The parameter d depends on the chosen approximation system. For $\Gamma \subset \mathbb{R}^3$, we usually have $d = 3$, while in special cases $d = 2$ is possible.

2. The function systems $\left\{\Phi_c^{(\nu)}\right\}_{\nu \in \mathcal{L}_c}$, $\left\{\Psi_c^{(\mu)}\right\}_{\mu \in \mathcal{R}_c}$ with suitable index sets $\mathcal{L}_c, \mathcal{R}_c \subset \mathbb{N}_0^d$ have a hierarchical structure. We require the expansion functions on clusters $c \in T \setminus \mathfrak{L}$ to fulfil the *refinement relations*

$$\begin{aligned} \Phi_c^{(\nu)}\big|_{\tilde{c}^\Gamma} &= \sum_{\tilde{\nu} \in \mathcal{L}_{\tilde{c}}} \gamma_{\nu,\tilde{\nu},\tilde{c}}^{\Phi} \Phi_{\tilde{c}}^{(\tilde{\nu})}, \qquad \forall c \in T \setminus \mathfrak{L}, \quad \forall \nu \in \mathcal{L}_c, \quad \forall \tilde{c} \in \mathrm{sons}(c), \\ \Psi_c^{(\mu)}\big|_{\tilde{c}^\Gamma} &= \sum_{\tilde{\mu} \in \mathcal{R}_{\tilde{c}}} \gamma_{\mu,\tilde{\mu},\tilde{c}}^{\Psi} \Psi_{\tilde{c}}^{(\tilde{\mu})}, \qquad \forall c \in T \setminus \mathfrak{L}, \quad \forall \mu \in \mathcal{R}_c, \quad \forall \tilde{c} \in \mathrm{sons}(c), \end{aligned} \tag{28}$$

with some *shift coefficients* $\gamma^{\Phi}_{\nu,\tilde{\nu},\tilde{c}}, \gamma^{\Psi}_{\nu,\tilde{\nu},\tilde{c}} \in \mathbb{R}$. The index sets $\mathcal{L}_c, \mathcal{R}_c \subset \mathbb{N}_0^d$ will be specified in Definition 11. Via the refinement relations we are able to represent the expansion functions on larger clusters recursively by their restrictions to smaller ones.

3. The kernel approximation $\tilde{k}_{\mathbf{b}}$ possesses a *local approximation property* on each farfield block $\mathbf{b} = (c_1, c_2)$, i.e. there exist $s \in \mathbb{R}$, $\eta \in (0,1)$ and positive constants $C_1 < \infty$, $C_2 < 1$ such that, for $m \in \mathbb{N}$, the estimate
$$|k(x,y) - \tilde{k}_{\mathbf{b}}(x,y)| \leq C_1 C_2^m \operatorname{dist}^{-s}(c_1,c_2), \qquad \forall (x,y) \in \mathbf{b}$$
holds.

Remark 7. *If the kernel function of an integral operator is the sum of kernel functions, the panel-clustering approximation is applied separately to each part of the sum.*

Example 8. *Let*
$$\Theta_c := \left\{ \theta_c^{(\nu)} \in \mathbb{R}^3 \mid 0 \leq \nu_i \leq m, i = 1,2,3 \right\}$$
denote the set of $(m+1)^3$ *equidistant interpolation points on a cluster* c. *Let* $\Lambda_c^{(\nu)}$ *be the three-dimensional Lagrange polynomial normalised to* c *corresponding to point* $\theta_c^{(\nu)}$. *The interpolant* $I_{\mathbf{b}}[k]$ *of the fundamental solution of the Laplacian* $k(x,y) := G(x-y)$ *with* G *as in* (5) *on a farfield block* $\mathbf{b} = (c_1, c_2)$ *is given by*
$$I_{\mathbf{b}}[k](x,y) := \sum_{\nu,\mu \leq m(\mathbf{b})} k\left(\theta_{c_1}^{(\nu)} - \theta_{c_2}^{(\mu)}\right) \Lambda_{c_1}^{(\nu)}(x) \Lambda_{c_2}^{(\mu)}(y), \tag{29}$$
where the notation "$\nu, \mu \leq m(\mathbf{b})$*" is short for* $(\nu,\mu) \in \mathcal{I}_{\mathbf{b}}$ *and the index set* $\mathcal{I}_{\mathbf{b}}$ *is given by*
$$\mathcal{I}_{\mathbf{b}} := \left\{ (\nu,\mu) \in \mathbb{N}_0^3 \times \mathbb{N}_0^3 \mid 0 \leq \nu_i, \mu_i \leq m(\mathbf{b}), i = 1,2,3 \right\}. \tag{30}$$
Since block $\mathbf{b}$ *is admissible, the clusters* c_1 *and* c_2 *are well separated, and the expansion coefficients* $\kappa_{\nu,\mu} := k\left(\theta_{c_1}^{(\nu)} - \theta_{c_2}^{(\mu)}\right)$ *are well defined. The Lagrange polynomials fulfil the refinement relations* (28) *with shift coefficients*
$$\gamma_{\nu,\tilde{\nu},\tilde{c}} := \Lambda_c^{(\nu)}\left(\theta_{\tilde{c}}^{(\tilde{\nu})}\right). \tag{31}$$

Example 9. *Since the kernels of boundary integral operators are suitable Gâteau derivatives of the fundamental solution, we may apply these derivatives to the expansion of the fundamental solution in order to obtain an approximation to the integral kernels.*

Example 10. *For the Laplace operator, it is well known that the multipole expansion coincides with the three-dimensional Taylor series in appropriate two-dimensional coordinates. Hence, in these coordinates a two-dimensional approximation of the integral kernels corresponding to the Laplace problem can be derived. For details, we refer to [11].*

The panel-clustering approximation of integral operator K is given by

$$\begin{aligned}\langle v, K[u]\rangle_{L^2(\Gamma)} &\approx \sum_{\mathbf{b}\in P_{\text{near}}} \int_{\mathbf{b}^\Gamma} v(x)k(x,y)u(y)\,ds_x ds_y \\ &\quad + \sum_{\mathbf{b}\in P_{\text{far}}} \sum_{(\nu,\mu)\in\mathcal{I}_\mathbf{b}} \kappa_{\nu,\mu}(\mathbf{b}) \int_{c_1^\Gamma} \Phi_{c_1}^{(\nu)}(x)v(x)\,ds_x \int_{c_2^\Gamma} \Psi_{c_2}^{(\mu)}(y)u(y)\,ds_y \\ &=: a_{\text{near}}(u,v) + a_{\text{far}}(u,v).\end{aligned} \tag{32}$$

Note that the integrals are defined over the portions $\mathbf{b}^\Gamma \subset \Gamma\times\Gamma$ which are linked to the block $\mathbf{b}$ via the function σ (cf. Definition 5).

We abbreviate the farfield integrals by

$$L_c^{(\nu)}[v] := \int_{c^\Gamma} v(x)\Phi_c^{(\nu)}(x)\,ds_x, \qquad R_c^{(\mu)}[u] := \int_{c^\Gamma} u(x)\Psi_c^{(\mu)}(x)\,ds_x, \qquad \forall c\in T. \tag{33}$$

These quantities are the *farfield coefficients.*

The nearfield part a_{near} is evaluated in the usual, matrix-oriented way, yielding a sparse matrix. For an efficient evaluation of the matrix-vector multiplication we will rewrite the farfield part a_{far} in (32).

Observe that the farfield coefficients R and L can be computed independently, and the coupling in x and y takes place in the summation over the farfield blocks. To decouple the computation of the integrals, we need to introduce one-sided restrictions of the index set $\mathcal{I}_\mathbf{b}$.

Definition 11. *For $\mathbf{b}\in P$ we define one-sided restrictions of $\mathcal{I}_\mathbf{b}\subset\mathbb{N}_0^d\times\mathbb{N}_0^d$ by*

$$\begin{aligned}\mathcal{L}_\mathbf{b} &:= \Big\{\nu\in\mathbb{N}_0^d \mid \exists\mu\in\mathbb{N}_0^d\colon (\nu,\mu)\in\mathcal{I}_\mathbf{b}\Big\},\\ \mathcal{R}_\mathbf{b} &:= \Big\{\mu\in\mathbb{N}_0^d \mid \exists\nu\in\mathbb{N}_0^d\colon (\nu,\mu)\in\mathcal{I}_\mathbf{b}\Big\},\\ \mathcal{R}_\mathbf{b}(\nu) &:= \{\mu\in\mathcal{R}_\mathbf{b} \mid (\nu,\mu)\in\mathcal{I}_\mathbf{b}\}.\end{aligned}$$

The multi-index sets on a cluster $c\in T$ are then given by

$$\mathcal{L}_c := \bigcup_{c'\in T:(c,c')\in P_{\text{far}}} \mathcal{L}_{(c,c')}, \qquad \mathcal{R}_c := \bigcup_{c'\in T:(c',c)\in P_{\text{far}}} \mathcal{R}_{(c',c)}. \tag{34}$$

We also introduce the one-sided restriction of the farfield

$$P_{\text{far}}(c) := \big\{c'\in T \mid (c,c')\in P_{\text{far}}\big\}, \qquad \forall c\in T.$$

The definition implies the inclusions

$$\begin{aligned}\mathcal{I}_\mathbf{b} &\subseteq \mathcal{L}_{c_1}\times\mathcal{R}_{c_2}, \qquad \forall\mathbf{b}=(c_1,c_2)\in P,\\ \mathcal{L}_{\tilde c} &\subseteq \mathcal{L}_c, \quad \mathcal{R}_{\tilde c}\subseteq\mathcal{R}_c \qquad \forall\tilde c\in\text{sons}(c).\end{aligned} \tag{35}$$

These splittings induce a splitting of the sums (27), (32). Using this notation, (27) can be rewritten as

$$\tilde{k}_{\mathbf{b}}(x,y) = \sum_{\nu \in \mathcal{L}_{\mathbf{b}}} \Phi_{c_1}^{(\nu)}(x) \sum_{\mu \in \mathcal{R}_{\mathbf{b}}(\nu)} \kappa_{\nu,\mu}(\mathbf{b}) \Psi_{c_2}^{(\mu)}(y), \tag{36}$$

and the bilinear form a_{far} now takes the form

$$a_{\text{far}}(u,v) = \sum_{c_1 \in T} \sum_{c_2 \in P_{\text{far}}(c_1)} \sum_{\nu \in \mathcal{L}_{(c_1,c_2)}} L_{c_1}^{(\nu)}[v] \sum_{\mu \in \mathcal{R}_{(c_1,c_2)}(\nu)} \kappa_{\nu,\mu}((c_1,c_2)) R_{c_2}^{(\mu)}[u]. \tag{37}$$

To obtain a proper structuring of the matrix-vector multiplication, we introduce the intermediate quantity

$$B_{c_1}^{(\nu)}[u] := \sum_{\substack{c_2 \in P_{\text{far}}(c_1) \\ \text{s.t. } \nu \in \mathcal{L}_{(c_1,c_2)}}} \sum_{\mu \in \mathcal{R}_{(c_1,c_2)}(\nu)} \kappa_{\nu,\mu}((c_1,c_2)) R_{c_2}^{(\mu)}[u], \tag{38}$$

and derive, by interchanging the second and third summation in (37), finally

$$a_{\text{far}}(u,v) = \sum_{c \in T} \sum_{\nu \in \mathcal{L}_c} L_c^{(\nu)}[v] B_c^{(\nu)}[u]. \tag{39}$$

This representation will be used for the evaluation of the panel-clustering approximation which is needed for the matrix-vector multiplication. The only missing part of the panel-clustering algorithm is the introduction of a hierarchical procedure to compute the farfield coefficients $L_c^{(\nu)}$, $R_c^{(\mu)}$ which will be the topic of the next section.

4.3.2 Construction of the expansion system

Since the farfield coefficients (33) have to be computed for every evaluation of the panel-clustering approximation, an efficient computation of the integrals in (33) is essential for the performance of the algorithm and we will employ the hierarchical structure of both the cluster tree (built in by construction) and the expansion functions (assumption (28)).

Note that the refinement relation (28) in the case of polynomial expansion systems will in general not hold if the approximation orders are different on different levels of the cluster tree, i.e. $\Phi_c^{(\nu)}|_{\tilde{c}^\Gamma} \notin \text{span}_{\tilde{\nu} \in \mathcal{L}_{\tilde{c}}} \left\{ \Phi_{\tilde{c}}^{(\tilde{\nu})} \right\}$ and $\Psi_c^{(\nu)}|_{\tilde{c}^\Gamma} \notin \text{span}_{\tilde{\nu} \in \mathcal{R}_{\tilde{c}}} \left\{ \Psi_{\tilde{c}}^{(\tilde{\nu})} \right\}$. Thus the construction of the approximation system is performed in two steps. We start with a *reference approximation system* which satisfies the approximation property but is not necessarily nested. If, e.g., the approximation of the kernel is based on polynomial interpolation the *reference approximation system* is given by $\check{\Phi}_c^{(\nu)} = \Lambda_c^{(\nu)}$ with $\Lambda_c^{(\nu)}$ as in Example 8.

The final *variable order approximation system* is obtained by using (28) as the *definition* of the expansion functions on clusters $c \in T \setminus \mathfrak{L}$.

Definition 12. *Let $\check{\Phi}$ and $\check{\Psi}$ denote the expansion functions given by the reference approximation system. Then the expansion functions for the variable order panel-clustering algorithm are given by the recursion*

$$\begin{aligned} \Phi_c^{(\nu)} &:= \check{\Phi}_c^{(\nu)}, \qquad \forall c \in \mathfrak{L}, \quad \forall \nu \in \mathcal{L}_c, \\ \Psi_c^{(\mu)} &:= \check{\Psi}_c^{(\mu)}, \qquad \forall c \in \mathfrak{L}, \quad \forall \mu \in \mathcal{R}_c, \end{aligned}$$

and

$$\begin{aligned} \Phi_c^{(\nu)}\big|_{\tilde{c}^\Gamma} &:= \sum_{\tilde{\nu} \in \mathcal{L}_{\tilde{c}}} \gamma^\Phi_{\nu,\tilde{\nu},\tilde{c}} \Phi_{\tilde{c}}^{(\tilde{\nu})}, \qquad \forall c \in T \setminus \mathfrak{L}, \quad \forall \nu \in \mathcal{L}_c, \quad \forall \tilde{c} \in \operatorname{sons}(c), \\ \Psi_c^{(\mu)}\big|_{\tilde{c}^\Gamma} &:= \sum_{\tilde{\mu} \in \mathcal{R}_{\tilde{c}}} \gamma^\Psi_{\mu,\tilde{\mu},\tilde{c}} \Psi_{\tilde{c}}^{(\tilde{\mu})}, \qquad \forall c \in T \setminus \mathfrak{L}, \quad \forall \mu \in \mathcal{R}_c, \quad \forall \tilde{c} \in \operatorname{sons}(c), \end{aligned} \tag{40}$$

where the shift coefficients $\gamma^\Phi_{\nu,\tilde{\nu},\tilde{c}}$, $\gamma^\Psi_{\mu,\tilde{\mu},\tilde{c}}$ are defined as in (28).

The resulting approximation systems $\left\{\Phi_c^{(\nu)}\right\}_{\nu \in \mathcal{L}_c}$, $\left\{\Psi_c^{(\mu)}\right\}_{\mu \in \mathcal{R}_c}$ can be regarded as an approximation of the reference expansion system.

All standard cluster methods for BEM are using a fixed approximation order for all farfield blocks $\mathbf{b} \in P_{\text{far}}$. On the other hand it was shown in [12] for the classical double layer potential on smooth surfaces that a linear growth of the approximation order towards the root of the cluster tree suffices to preserve the optimal convergence rate while yielding an optimal time and storage complexity. The approach utilizes the fact that on small blocks lower approximation orders yield sufficient accuracy to preserve the asymptotic convergence rate. Here, the size of a block is determined by the size of the larger cluster which in turn is correlated directly with its level in the cluster tree. The functional dependence of the approximation orders on the blocks is defined below.

Definition 13. *The* order distribution function $m\colon P_{\text{far}} \to \mathbb{N}_0$ *is given by*

$$m(\mathbf{b}) := \lceil \alpha(d_{\min} - \min\{\operatorname{level}(c_1), \operatorname{level}(c_2)\})_+ + \beta \rceil, \tag{41}$$

where $\alpha, \beta \in \mathbb{R}_{\geq 0}$, $d_{\min}$ as in (22) *and*

$$(\cdot)_+ := \max\{0, \cdot\}, \qquad \lceil x \rceil := \min\{z \in \mathbb{Z} \mid z \geq x\}.$$

The extension of the order distribution function to $m\colon P_{\text{far}} \cup T \to \mathbb{N}_0$ is given by

$$m(c) := \max\{\, m(\mathbf{b}) \mid \mathbf{b} = (c_1, c_2) \in P_{\text{far}} \text{ and } c \in \{c_1, c_2\}\} \quad \forall c \in T. \tag{42}$$

Besides the chosen approximation system itself, the approximation order is the only parameter that the index sets on the blocks and their restrictions depend on. Checking back with the requirements on the approximation system and (34) and (35), we find that the order distribution function implements all the previously stated properties of the index sets.

4.3.3 Evaluation of the panel-clustering approximation/matrix-vector multiplication

To assemble the panel-clustering approximation of the discrete bilinear form, we replace the test function by the basis functions b_i, $i = 1, \ldots, N$, and use the basis representation (12) of the grid function u.

The nearfield entries are computed by

$$\mathbf{A}_{i,j}^{\text{near}} := \sum_{c_1 \in \mathfrak{L}} \int_{c_1^\Gamma} b_i(x) \sum_{c_2 \in P_{\text{near}}(c_1)} \int_{c_2^\Gamma} k(x,y) b_j(y)\, ds_y ds_x. \tag{43}$$

Note that the first sum only contains clusters with $\text{area}(c^\Gamma \cap \text{supp}(b_i)) > 0$. In [12], it was shown that, under moderate assumptions on the mesh, the nearfield matrix only contains $O(N)$ non-zero entries.

For the evaluation of the farfield part we use the representation (39). It consists of three phases:

1. *Upward path:* Calculate the farfield coefficients $R_c^{(\mu)}[u]$ for all $c \in T$ and all $\mu \in \mathcal{R}_c$ using the refinement relation (40).
2. *Block interaction:* For all clusters $c \in T$ and all $\nu \in \mathcal{L}_c$ add all associated farfield coefficients multiplied by the expansion coefficients and store the quantities $B_c^\nu[u]$, cf. (38).
3. *Downward path:* For all clusters $c \in T$ and $\nu \in \mathcal{L}_c$ multiply the farfield coefficients $L^{(\nu)}[v]$ with $B_c^\nu[u]$ and distribute them downwards the cluster tree to the leaves.

We explain each step in detail and formulate the algorithms in a pseudo programming language.

Upward path:
On the leaves of the cluster tree we have to evaluate and store the *basis farfield coefficients*

$$R_c^{(\mu)}[b_i] := \int_{c^\Gamma} \Psi_c^{(\mu)}(y) b_i(y)\, ds_y, \qquad c \in \mathfrak{L}, \quad b_i \in \mathcal{S}, \tag{44}$$

$$L_c^{(\nu)}[b_i] := \int_{c^\Gamma} \Phi_c^{(\nu)}(x)\, b_i(x)\, ds_x, \qquad c \in \mathfrak{L}, \quad b_i \in \mathcal{S}. \tag{45}$$

Note that the localness of the standard basis functions implies that for all leaves $c \in \mathfrak{L}$ only $O(1)$ basis farfield coefficients are nonzero, namely, the ones with $\text{area}(c^\Gamma \cap \text{supp}(b_i)) > 0$.

Using the basis representation (12), the farfield coefficients on a leaf $c \in \mathfrak{L}$ are

$$R_c^{(\mu)}[u] := \sum_{i=1}^{N} u_i \int_{c^\Gamma} \Psi_c^{(\mu)}(y) b_i(y)\, ds_y = \sum_{i=1}^{N} u_i R_c^{(\mu)}[b_i], \qquad c \in \mathfrak{L}. \tag{46}$$

Since the expansion functions on clusters $c \in T \setminus \mathfrak{L}$ are defined piecewise via the refinement relation (40), the farfield coefficients (33) are assembled via the relation

$$R_c^{(\mu)}[u] = \sum_{\tilde{c} \in \mathrm{sons}(c)} \sum_{\tilde{\mu} \in \mathcal{R}_{\tilde{c}}} \gamma^{\Psi}_{\mu,\tilde{\mu},\tilde{c}} R_{\tilde{c}}^{(\tilde{\mu})}[u], \qquad c \in T \setminus \mathfrak{L}. \tag{47}$$

Note that this evaluation is performed upwards the cluster tree starting from the leaves. The procedure `upward_path`, cf. Procedure 3, summarizes these steps in algorithmic form. Initialising $R_c^{(\mu)}[u] := 0$, for all $c \in T$, $\mu \in \mathcal{R}_c$, the call `upward_path`(u) computes the farfield coefficients for all $c \in T$.

Procedure 3 `upward_path`(u)

```
for all $l = d_{\max}, \dots, 0$ do  /* loop over tree from leaves to root */
  for all $c \in T_l$ do
    if $c \in \mathfrak{L}$ then  /* evaluate (46) on leaves of cluster tree */
      for all $\mu \in \mathcal{R}_c$ do
        for all $b_i$ with $\mathrm{area}(c^\Gamma \cap \mathrm{supp}(b_i)) > 0$ do
          $R_c^{(\mu)}[u] := R_c^{(\mu)}[u] + u_i R_c^{(\mu)}[b_i]$
    else  /* evaluate (47) on clusters $c \in T \setminus \mathfrak{L}$ */
      for all $\tilde{c} \in \mathrm{sons}(c)$ do
        for all $\tilde{\mu} \in \mathcal{R}_{\tilde{c}}$ do
          $R_c^{(\mu)}[u] := R_c^{(\mu)}[u] + \gamma^{\Psi}_{\mu,\tilde{\mu},\tilde{c}} R_{\tilde{c}}^{(\tilde{\mu})}[u]$
```

Block interaction:
Evaluate formula (38) for all clusters $c \in T$.

Procedure 4 `block_exchange`$()$

```
for all $c_1 \in T$ do
  for all $\nu \in \mathcal{L}_{c_1}$ do
    for all $c_2 \in P_{\mathrm{far}}(c_1)$ do
      for all $\mu \in \mathcal{R}_{(c_1,c_2)}(\nu)$ do
        $B_{c_1}^{(\nu)}[u] := B_{c_1}^{(\nu)}[u] + \kappa_{\nu,\mu}((c_1,c_2)) R_{c_2}^{(\mu)}[u]$
```

Downward path:
For the evaluation of the outer integrals (w.r.t. the x-variable) we make again use of the refinement relation (40). The idea is to distribute the quantities B from larger clusters down to the sons until the leaves are reached. As in the upward path step this can be performed within one sweep over the cluster tree.

Consider the sum (39) for a fixed cluster $c \in T \setminus \mathfrak{L}$. Since (39) is a sum over all clusters in T, it contains in particular the partial sum

$$\sum_{\nu \in \mathcal{L}_c} L_c^{(\nu)}[b_i] B_c^{(\nu)}[u] + \sum_{\tilde{c} \in \mathrm{sons}(c)} \sum_{\tilde{\nu} \in \mathcal{L}_{\tilde{c}}} L_{\tilde{c}}^{(\nu)}[b_i] B_{\tilde{c}}^{(\tilde{\nu})}[u].$$

The refinement relation (40) allows to represent the farfield coefficients on c by the ones defined on its sons, yielding

$$\sum_{\tilde{c}\in\operatorname{sons}(c)} \sum_{\tilde{\nu}\in\mathcal{L}_{\tilde{c}}} L_{\tilde{c}}^{(\tilde{\nu})}[b_i] \left\{ B_{\tilde{c}}^{(\tilde{\nu})}[u] + \sum_{\nu\in\mathcal{L}_c} \gamma_{\nu,\tilde{\nu},\tilde{c}}^{\Phi} B_c^{(\nu)}[u] \right\}.$$

We see that the data is distributed downwards the cluster tree. Hence the procedure `downward_path` starts at the root of the cluster tree and distributes the data down to the leaves. Finally we need to multiply the coefficients $B_c^{\nu}[u]$, $c \in \mathfrak{L}$, with the basis farfield coefficients $L_c^{(\nu)}[b_i]$. Thus the i-th component of the vector v is given by

$$v_i = \sum_{c\in\mathfrak{L}} \sum_{\nu\in\mathcal{L}_c} L_c^{(\nu)}[b_i] B_c^{(\nu)}[u].$$

Note again that the first sum contains only clusters c with area $\left(c^{\Gamma} \cap \operatorname{supp}(b_i)\right) > 0$.

Procedure 5 `downward_path`(v)

```
for all $l = 0, \dots, d_{\max} - 1$ do  /* distribute coefficients to the sons */
  for all $c \in T_l$ do
    for all $\tilde{c} \in \operatorname{sons}(c)$ do
      for all $\tilde{\nu} \in \mathcal{L}_{\tilde{c}}$ do  /* shift $B$s from $c$ to $\tilde{c}$ */
        for all $\nu \in \mathcal{L}_c$ do
          $B_{\tilde{c}}^{(\tilde{\nu})}[u] := B_{\tilde{c}}^{(\tilde{\nu})}[u] + \gamma_{\nu,\tilde{\nu},\tilde{c}}^{\Phi} B_c^{(\nu)}[u]$
for all $c \in \mathfrak{L}$ do  /* update $v$ on leaves */
  for all $b_i$ with area$(c^{\Gamma} \cap \operatorname{supp}(b_i)) > 0$ do
    for all $\nu \in \mathcal{L}_c$ do
      $v[i] := v[i] + L_c^{(\nu)}[b_i]\, B_c^{(\nu)}[u]$
```

Below we summarize all steps of the panel-clustering algorithm.

Input: Mesh $\mathcal{G}$, space $\mathcal{S}$, order distribution function $m : P \to \mathbb{N}$, admissibility constant η
Setup:

1. Build cluster tree T, cf. Procedure 1 and Definition 5, and compute for all c in T the minimal boxes box(c).

2. Compute the minimal covering P by using Procedure 2 and assign expansion orders to admissible blocks and afterwards to clusters, cf. Definition (41) and (42).

3. Assemble nearfield matrix, cf. (43).

4. Compute expansion coefficients, cf. Section 4.4, and basis farfield coefficients (44) and (45).

Matrix-Vector Multiplication:

- upward path, cf. Procedure 3
- block interaction, cf. Procedure 4
- downward path, cf. Procedure 5
- nearfield multiplication, cf. (43)

4.4 Panel-clustering approximation via interpolation

So far the panel-clustering algorithm was treated only in a formal fashion utilising only the three assumptions made at the beginning of Section 4.3. For a concrete realisation the function systems and the approximation method, e.g. Taylor, interpolation, projection, and the admissibility constant have to be selected. We will present here a panel-clustering approximation based on interpolation with polynomials.

This section is divided into two parts. In the first part we describe the construction of a discontinuous interpolant which will be used for the kernel functions k_1 and k_2, cf. (20). The approximation of the kernel function k_3 is more subtle since it requires a continuous interpolation and will be explained afterwards.

4.4.1 Approximation of k_1 and k_2

The kernel approximation via interpolation was already introduced in Example 8. We use it as the reference approximation system to construct a variable order panel-clustering approximation.

The order distribution function m from Definition 13 is used for the assignment of the approximation order to farfield blocks. While the index set $\mathcal{I}_{\mathbf{b}}$ is given in (30), the index sets $\mathcal{L}_{\mathbf{b}}$, $\mathcal{R}_{\mathbf{b}}$, $\mathcal{R}_{\mathbf{b}}(\nu)$, $\mathcal{L}_c$ and $\mathcal{R}_c$ are as in Definition 11.

To obtain a variable order approximation, cf. Definition 12, *approximated* Lagrange polynomials are used as expansion functions

$$\begin{aligned} \Phi_c^{(\nu)} &:= \Lambda_c^{(\nu)}, && \forall c \in \mathfrak{L}, \quad \nu \le m(c), \\ \Phi_c^{(\nu)}\big|_{\tilde{c}^\Gamma} &:= \sum_{\tilde{\nu} \le m(\tilde{c})} \gamma_{\nu,\tilde{\nu},\tilde{c}} \Phi_{\tilde{c}}^{(\tilde{\nu})}, && \forall \tilde{c} \in \mathrm{sons}(c), \quad c \in T \setminus \mathfrak{L}, \quad \nu \le m(c). \end{aligned} \tag{48}$$

where the shift coefficients $\gamma_{\nu,\tilde{\nu},\tilde{c}}$ are as in (31). The panel-clustering approximation of the kernel functions k_i, $i = 1, 2$, on farfield block $\mathbf{b} = (c_1, c_2)$ has the representation

$$\tilde{k}_{\mathbf{b}}(x, y) := \sum_{\nu,\mu \le m(\mathbf{b})} \kappa_{\nu,\mu}(\mathbf{b}) \Phi_{c_1}^{(\nu)}(x) \Phi_{c_2}^{(\mu)}(y).$$

Remark 14. *In the case of a discontinuous interpolant we use the function σ_1, defined in* (23) *for the association of a cluster $c \in T$ to portions of the surface*

Γ. *This choice affects only the computation of the basis farfield coefficients* (44) *and* (45). *The integration is carried out in the standard way over $\sigma_1(c)$, i.e. the set of all panels τ with $M_\tau \in c^\Gamma$.*

4.4.2 Approximation of k_3

The panel-clustering representation of the kernel function k_3, cf. (20), is more involved in order to preserve the optimal convergence rate of the undisturbed Galerkin method. Provided the solution u is in $H^1(\Gamma)$, the error estimate employs a partial integration, shifting the derivatives of the surface Laplacian to u.

To be more precise, we have to construct a continuous interpolation not to the kernel function k_3 itself but to its antiderivative. Hence, the construction of the panel-clustering approximation of k_3, cf. (20), starts with the function

$$\mathfrak{K}(x-y) := \frac{1}{4\pi}\|x-y\|. \tag{49}$$

The interpolant $I[\mathfrak{K}]$ of this function is well defined on $\mathbb{R}^3 \times \mathbb{R}^3$. As a consequence we replace k_3 not only in the farfield but also in the nearfield by the panel-clustering approximation. This means that all blocks in P are admissible.

The blockwise interpolant $I_{\mathbf{b}}[\mathfrak{K}]$ on $\mathbf{b} = (c_1, c_2)$ is given by (29). It will in general not be continuous across the common boundaries of neighbouring blocks with different approximation orders. The goal now is to modify the interpolant such that the global continuity in combination with the variable order approximation is retained.

This is possible since all of the expansion functions are defined piecewise as linear combinations of the expansion functions on the underlying leaves, cf. (48). The continuity across the boundaries of a block is enforced by a modification of the expansion coefficients $\kappa_{\nu,\mu}$.

We say two blocks $\mathbf{b}$ and $\tilde{\mathbf{b}}$ are *neighbours* if $\mathbf{b} \cap \tilde{\mathbf{b}} \neq \emptyset$. For two neighbouring blocks $\mathbf{b}$ and $\tilde{\mathbf{b}}$ with level$(\tilde{\mathbf{b}})$ = level$(\mathbf{b}) + 1$, we introduce the *shift operator*

$$\left(S_{\tilde{\mathbf{b}},\mathbf{b}}\kappa(\mathbf{b})\right)_{\tilde{\nu},\tilde{\mu}} := \sum_{\nu,\mu \leq m(\mathbf{b})} \kappa_{\nu,\mu}(\mathbf{b})\gamma_{\nu,\tilde{\nu},\tilde{c}_1}\gamma_{\mu,\tilde{\mu},\tilde{c}_2}. \tag{50}$$

The operator S can be extended to neighbouring blocks $\mathbf{b}$, $\tilde{\mathbf{b}}$ with level$(\tilde{\mathbf{b}}) >$ level$(\mathbf{b}) + 1$ by recursive application of S over the series of descendants connecting $\mathbf{b}$ with $\tilde{\mathbf{b}}$[†]. For a continuous interpolant, the coefficients on block $\tilde{\mathbf{b}} = (\tilde{c}_1, \tilde{c}_2) \in P$ have to be defined as follows

$$\kappa_{\tilde{\nu},\tilde{\mu}}(\tilde{\mathbf{b}}) := \begin{cases} \left(S_{\tilde{\mathbf{b}},\mathbf{b}}\kappa(\mathbf{b})\right)_{\tilde{\nu},\tilde{\mu}} & \text{if } \exists \mathbf{b} \in P\text{:} \\ & \text{level}(\mathbf{b}) < \text{level}(\tilde{\mathbf{b}}) \wedge \left(\theta_{\tilde{c}_1}^{(\tilde{\nu})}, \theta_{\tilde{c}_2}^{(\tilde{\mu})}\right) \in \mathbf{b} \cap \tilde{\mathbf{b}}, \\ \mathfrak{K}\left(\theta_{\tilde{c}_1}^{(\tilde{\nu})} - \theta_{\tilde{c}_2}^{(\tilde{\mu})}\right) & \text{otherwise.} \end{cases} \tag{51}$$

[†]Note that for $\widetilde{\text{sons}}(c) = \{c\}$ (cf. (26)), with $c \in \{c_1, c_2\}$, we shift only with respect to one variable.

Using the notation P_l for the set of blocks $\mathbf{b}$ of level l and

$$\mathcal{U}(\mathbf{b}) := \{\ \mathbf{b}^* \in P \mid \text{level}(\mathbf{b}^*) > \text{level}(\mathbf{b}) \text{ and } \mathbf{b}^* \cap \mathbf{b} \neq \emptyset\},$$

for the set of all smaller neighbours of block $\mathbf{b}$, the pseudo algorithm given in Procedure 6 computes the expansion coefficients for all blocks $\mathbf{b} \in P$. Note that the set of neighbours of a block is given as the Cartesian product of the sets of neighbours of the involved clusters.

Procedure 6 `compute_coefficients()`

```
for all $l = 0, \dots, d_{\max}$ do
  for all $\mathbf{b} \in P_l$ do
    for all $\nu, \mu \leq m(\mathbf{b})$ do
      if $\kappa_{\nu,\mu}(\mathbf{b})$ not yet computed then
        $\kappa_{\nu,\mu}(\mathbf{b}) := \mathfrak{K}\Big(\theta_{c_1}^{(\nu)} - \theta_{c_2}^{(\mu)}\Big)$
    for all $\mathbf{b}^* \in \mathcal{U}(\mathbf{b})$ do
      for all $\Big(\theta_{c_1^*}^{(\nu^*)}, \theta_{c_2^*}^{(\mu^*)}\Big) \in \mathbf{b}^\Gamma \cap (\mathbf{b}^*)^\Gamma$ do
        $\kappa_{\nu^*,\mu^*}(\mathbf{b}^*) := (S_{\mathbf{b}^*,\mathbf{b}}\kappa(\mathbf{b}))_{\nu^*,\mu^*}$
```

The panel-clustering approximation of k_3 is finally obtained by blockwise application of the surface Laplacian to the interpolant. We have employed the fact that we can write

$$\begin{aligned} \Delta_y^\Gamma \|x - y\| &= -\operatorname{curl}_{\Gamma,z} \left(n_y \times \nabla_x \|x - z\|\right)\big|_{z=y} \\ &= \left\langle n_y \frac{\partial}{\partial n_y} - \nabla_{\Gamma,y}, \nabla_{\Gamma,x} \right\rangle \|x - y\|. \end{aligned}$$

The expansion functions are thus given by applying the respective derivatives and truncation to the respective approximation order

$$\begin{aligned} \overrightarrow{\Phi}_c^{(\nu)} &:= \nabla_\Gamma \Lambda_c^{(\nu)}, && \forall c \in \mathfrak{L}, \quad \nu \leq m(c), \\ \overrightarrow{\Phi}_c^{(\nu)}\big|_{\tilde{c}^\Gamma} &:= \sum_{\tilde{\nu} \leq m(\tilde{c})} \gamma_{\nu,\tilde{\nu},\tilde{c}} \overrightarrow{\Phi}_{\tilde{c}}^{(\tilde{\nu})}, && \forall \tilde{c} \in \text{sons}(c), \quad c \in T \setminus \mathfrak{L}, \quad \nu \leq m(c), \\ \overrightarrow{\Psi}_c^{(\mu)} &:= (n\langle n, \nabla_\Gamma\rangle - \nabla_\Gamma)\Lambda_c^{(\mu)} && \forall c \in \mathfrak{L}, \quad \mu \leq m(c), \\ \overrightarrow{\Psi}_c^{(\mu)}\big|_{\tilde{c}^\Gamma} &:= \sum_{\tilde{\mu} \leq m(\tilde{c})} \gamma_{\mu,\tilde{\mu},\tilde{c}} \overrightarrow{\Psi}_{\tilde{c}}^{(\tilde{\mu})}, && \forall \tilde{c} \in \text{sons}(c), \quad c \in T \setminus \mathfrak{L}, \quad \mu \leq m(c) \end{aligned} \tag{52}$$

with shift coefficients $\gamma_{\nu,\tilde{\nu},c}$ as in (31). The final panel-clustering approximation of k_3 reads

$$\tilde{k}_{\mathbf{b}}(x, y) := \sum_{\nu,\mu \leq m(\mathbf{b})} \kappa_{\nu,\mu}(\mathbf{b}) \Big\langle \overrightarrow{\Phi}_{c_1}^{(\nu)}(x), \overrightarrow{\Psi}_{c_2}^{(\mu)}(y) \Big\rangle,$$

with $\kappa_{\nu,\mu}(\mathbf{b})$ as in (51) and expansion functions $\overrightarrow{\Phi}_c^{(\nu)}$ as in (52).

Remark 15. *In order to preserve the continuity of the interpolation operator for the approximation of the function* $\mathfrak{K}$ *in* (49)*, the integration which is involved in the definition of the basis farfield coefficients* (44)*,* (45) *has to be defined over the intersection* $c \cap \Gamma$*. Thus, the function* σ_2 *from* (24) *is employed for the association of surface pieces to clusters.*

Remark 16. *The concept of the panel-clustering algorithm can be applied to general linear differential equations of second order with constant coefficients. This is in contrast to the multipole method which is a special technique for Laplace's equation (and variants exist for the Helmholtz equation). Since the multipole method is based on two-dimensional expansions, we expect that it performs more efficiently for Laplace's equation. The development of a variable order multipole method for alternative representations and numerical comparisons are one topic of future investigations.*

5 Numerical Experiments

In order to validate the expected behaviour for the variable panel-clustering algorithm as described in this paper and in [12] we have performed a number of numerical tests for the three-dimensional screen problem

$$a_V(u,v) = f(v) \qquad \forall v \in \mathcal{S}^0 \tag{53}$$

with a_V as in (6) and piecewise constant boundary elements on a sequence of meshes $\mathcal{G}_\ell$, $\ell = 0, 1, \ldots$. We write u_ℓ for the corresponding Galerkin solution and $u_{PC,\ell}$ for the panel-clustering solution. The errors to the exact solution u are denoted by $e_\ell := u - u_\ell$ and $e_{PC,\ell} := u - u_{PC,\ell}$.

We have used a uniform panelisation of $\Gamma = [0,1]^2 \subset \mathbb{R}^3$ with triangles, where the mesh was refined by subdividing each triangle into four congruent subtriangles, i.e. the number of unknowns N on the mesh $\mathcal{G}_\ell$ is $2 \cdot 4^\ell$. The farfield approximation of the kernel function of the bilinear form a_V was realised via Taylor expansion as explained in [12]. The tests were performed on a SUN SunFire 6800.

For the first set of computations, the right-hand side was chosen such that the exact solution of (53) is $u(x) = x_1 + x_2$. The L^2-error $||e_\ell||_{L^2(\Gamma)}$ of the Galerkin solution u_ℓ over the tested series of meshes is shown in Table 1. The quotients for time and memory requirements are defined as

$$\rho_{t,\ell} := \frac{t_\ell}{t_{\ell-1}}, \qquad \rho_{\mathrm{Mem},\ell} := \frac{\mathrm{Mem}_\ell}{\mathrm{Mem}_{\ell-1}},$$

where the times comprise setup and solving of the linear system. For the memory we counted the entries of the Galerkin matrix respectively the entries of the nearfield matrix plus the farfield data for the panel-clustering method. The nearfield integration was performed by the blackbox quadrature described in [2] and the equation system was solved by GMRES without restart.

ℓ	N	$\|\|e_\ell\|\|_{L^2(\Gamma)}$	$\rho_{e_{G,\ell},\ell}$	t [sec]	$\rho_{t,\ell}$	Mem [KB]	$\rho_{\mathrm{Mem},\ell}$
3	128	0.03163	–	55.6	–	128	–
4	512	0.01641	1.93	796.4	14.32	2048	16
5	2048	0.00878	1.87	11974.2	15.04	32768	16
6	8192	0.00492	1.78	182364.7	15.23	524288	16
7	32768	0.00292	1.69	2904170.0	15.93	8388608	16

Table 1: Galerkin solution of (53) for $u(x) = x_1 + x_2$

Table 2 confirms the theoretical results which state that for sufficiently high (β, α) the panel-clustering algorithm with variable approximation order has linear complexity in terms of time and memory requirements. The quotients $\rho_{t,\ell}$ and $\rho_{\mathrm{Mem},\ell}$ tend to four with increasing refinement level ℓ. The ratio of the L^2-error of the Galerkin method to the L^2-error of the variable order panel-clustering method

$$\epsilon_\ell := \frac{||e_{PC,\ell}||_{L^2(\Gamma)}}{||e_\ell||_{L^2(\Gamma)}}$$

remains constant over the whole series of considered grids. The times include setup and solving of the system. Due to the very high memory requirements the undisturbed Galerkin solution could not be computed for level 8, so that a high order panel-clustering solution was used as reference.

ℓ	N	$\|\|e_{PC,\ell}\|\|_{L^2(\Gamma)}$	ϵ_ℓ	t [sec]	$\rho_{t,\ell}$	Mem [KB]	$\rho_{\mathrm{Mem},\ell}$
3	128	0.03162	0.9997	20.83	–	1864	–
4	512	0.01640	0.9994	101.11	4.85	13258	7.11
5	2048	0.00877	0.9982	469.52	4.64	72462	5.47
6	8192	0.00491	0.9968	2128.12	4.53	345372	4.77
7	32768	0.00291	0.9963	9342.28	4.38	1525198	4.41
8	131072	0.00184	0.9996	39704.70	4.25	6448310	4.23

Table 2: Results for the panel-clustering algorithm with $(\beta, \alpha) = (3, 1)$ and $\eta = 0.5$ for $u(x) = x_1 + x_2$.

While the first example shows that the variable order panel-clustering method behaves as theoretically predicted, the accuracy requirements in the form of ϵ_ℓ are much too restrictive for practical applications and, thus, the computing times too pessimistic. For the second test we have chosen the parameters (β, α) in the definition of the variable approximation order such that $\epsilon_\ell \leq 2$ for all tested levels ℓ. The admissibility constant was chosen as above as $\eta = 0.5$. The combination $(\beta, \alpha) = (1, 0.8)$ satisfies this requirement and is optimal with respect to the computing time.

Table 3 shows that the chosen parameter combination (β, α) yields much better computing times compared to the first example although it does not yield a constant

ℓ	N	$\|\|e_{PC,\ell}\|\|_{L^2(\Gamma)}$	ϵ_ℓ	t [sec]	$\rho_{t,\ell}$	Mem [KB]	$\rho_{\text{Mem},\ell}$
3	128	0.03233	1.0222	19.24	–	347	–
4	512	0.01872	1.1407	85.73	4.46	2270	6.54
5	2048	0.01171	1.3328	367.74	4.29	12272	5.41
6	8192	0.00804	1.6342	1542.82	4.20	58920	4.80
7	32768	0.00610	2.0870	6261.31	4.06	260443	4.42

Table 3: Results for the panel-clustering algorithm with $(\beta, \alpha) = (1, 0.8)$ and $\eta = 0.5$ for $u(x) = x_1 + x_2$.

ratio ϵ_ℓ for the tested series of grids. It appears that this holds only for sufficiently large β and α where the error is near or equal to the Galerkin error.

For the third series of tests we chose problem (53) with exact solution $u(x) = x_1^{-1/4}$. The convergence rate of the Galerkin method is about 0.84, as Table 4 shows. In addition, the table gives the optimal combination (β, α) with ϵ_ℓ close to one. The admissibility constant was again chosen as $\eta = 0.5$.

ℓ	N	$\|\|e_\ell\|\|_{L^2(\Gamma)}$	β	α	ϵ_ℓ
3	128	0.2613	1	0.2	1.0057
4	512	0.2201	1	0.2	1.0325
5	2048	0.1852	1	0.2	1.1175
6	8192	0.1556	1	0.4	1.1096
7	32768	0.1311	1	0.6	1.0033

Table 4: Comparison of Galerkin method and panel-clustering algorithm for $u(x) = x_1^{-1/4}$

As we can clearly see, the panel-clustering solution approaches the Galerkin solution for a very moderate choice of β and α, i.e. the additional error introduced by the panel-clustering approximation is not affected by the smoothness of the solution.

Acknowledgements: Thanks are due to J.-C. Nédélec for fruitful discussions concerning the regularisation of integral operators.

References

[1] S. Börm and S. Sauter. Alternative representations of classical boundary integral operators allowing cheap quadrature formulae and low order panel-clustering approximations. planned for 2002.

[2] S. Erichsen and S. Sauter. Efficient automatic quadrature in 3-d Galerkin BEM. *Comp. Meth. Appl. Mech. Eng.*, 157:215–224, 1998.

[3] I. Graham, W. Hackbusch, and S. Sauter. Hybrid Galerkin Boundary Elements: Theory and Implementation. *Numer. Math.*, 86(98-6):139–172, 2000.

[4] W. Hackbusch. *Elliptic Differential Equations.* Springer Verlag, 1992.

[5] W. Hackbusch. *Integral Equations.* ISNM. Birkhäuser, 1995.

[6] W. Hackbusch and Z. Nowak. On the Fast Matrix Multiplication in the Boundary Element Method by Panel-Clustering. *Numerische Mathematik*, 54:463–491, 1989.

[7] W. Hackbusch and S. Sauter. On the Efficient Use of the Galerkin Method to Solve Fredholm Integral Equations. *Applications of Mathematics*, 38(4-5):301–322, 1993.

[8] W. McLean. *Strongly Elliptic Systems and Boundary Integral Equations.* Cambridge, Univ. Press, 2000.

[9] J. Nédélec. Integral Equations with Non Integrable Kernels. *Integral Equations Oper. Theory*, 5:562–572, 1982.

[10] J. Nédélec. *Acoustic and Electromagnetic Equations.* Springer Verlag, 2001.

[11] V. Rokhlin. Rapid solutions of integral equations of classical potential theory. *Journal of Computational Physics*, 60(2):pp. 187–207, 1985.

[12] S. Sauter. Variable order panel clustering. *Computing*, 64:223–261, 2000.

[13] S. Sauter and C. Lage. Transformation of hypersingular integrals and black-box cubature. *Math. Comp.*, 70(97-17):223–250, 2001.

[14] S. Sauter and C. Schwab. Quadrature for hp-Galerkin BEM in $\mathbb{R}^3$. *Numer. Math.*, 78(2):211–258, 1997.