The Panel Clustering Method in 3-d BEM

Stefan A. Sauter

Abstract

In many cases, boundary value problems on a domain Ω can be rewritten as integral equations on the boundary of Ω. The discretization of this integral equation leads to a system of linear equations with a dense coefficient matrix of dimension N. In this paper, we will present the panel clustering algorithm which avoids the generation of the N^2 matrix entries by representing the integral operator on the discrete level by only $O\left(N \log^\kappa N\right)$ quantities. Thus, a matrix vector multiplication as a basic step in every iterative solver can be performed by $\mathrm{O}(N \log^\kappa N)$ operations. This method can be applied to all kinds of integral equations discretized by, e.g., the Nyström, the collocation or the Galerkin method.

1 Introduction

In this paper, we will present the panel-clustering method to solve boundary integral equations (BIE) arising from elliptic boundary value problems in three space dimensions. Elliptic boundary value problems can be rewritten as integral equations on the boundary of the domain in various ways. The common properties of all these integral equations are that they are pseudo-differential operators of integer order. The arising orders are

- 1: hypersingular equations of first kind,
- 0: second kind integral equations or
- -1: weakly singular equations of first kind.

To discretize these equations three methods are used

- the Nyström method,
- the collocation method and
- the Galerkin method.

From the theoretical point of view the Galerkin BEM applied to the hypersingular integral equation is the method of choice, because stability is ensured independent of the smoothness of the domain, convergence in weaker norms can be proven and observed, e.g., the high order convergence in interior points of the domain. In addition, fast solvers for the linear system as the multi-grid method or the CG-algorithm can be applied efficiently and robustly, independent of the smoothness of the domain. In contrast to this, it is very hard to prove stability (meaning existence and boundedness of the inverse of the discrete operator) for the collocation or the Nyström method on 3-d surfaces. Elschner and Rathsfeld proved in [2],[13], [14] stability for second kind integral equations with the double layer potential operator on polyhedral domains, provided graded meshes are used. Stability is still an open problem for more general problems as, e.g., the Lamé-equations in linear elasticity for both the Nyström method and the collocation method. The applications of these methods to hypersingular equation are not developed and analyzed. For example nodal point collocation is *not* applicable to hypersingular formulations on non-smooth domains. Therefore, in complicated situations, like e.g. equations on domains with only piecewise smooth surfaces, systems of equations, nonlinear integral equations or equations with mixed boundary conditions, the Galerkin method should be applied. On the other hand, if the problem is simpler and it is possible to apply the collocation or Nyström method, it turns out that these methods give the same accuracy as the Galerkin method with a substantially lower amount of CPU-time.

We will present here the panel clustering method which is applicable for all kinds of integral equations. The idea is to approximate the kernel function on the continuous level and then to discretize by any of the three discretization methods.

We restrict in this paper to the three dimensional case, i.e., the boundary of the domain is a piecewise smooth surface in $\mathbf{R}^3$. The application of the presented methods to 2-d problems is straightforward.

A method which is based on a similar idea is the multipole method introduced by Rokhlin in [15]. This method can be used to accelerate particle simulation problems. For boundary integral equations, however, this technique is restricted to Nyström discretizations, since the multipole technique is applied *after* the discretization of the integral equation.

A further sparse technique for integral equations is presented by Brandt and Lubrecht. Here, the multi-grid technique is used to avoid the full system matrix. The kernel function is evaluated in smooth regions on the coarse grid and then interpolated to finer grid by high-order interpolation. The application of the method to Nyström discretization is obvious, while the realization for Galerkin discretizations of general 3-d integral equations having piecewise smooth kernels on non-smooth surfaces seems to be possible but is not worked out yet. Therefore, a comparison of the complexity of this method with panel-clustering is not possible.

2 Preliminaries

2.1 Fredholm integral equations

Let Γ denote a piecewise smooth surface in $\mathbf{R}^3$. On Γ, we consider a Fredholm integral equation of the form

$$\lambda u(x) + K[u](x) = r(x), \quad \forall x \in \Gamma \tag{1}$$

with the integral operator K

$$K[u](x) = p.f. \int_\Gamma k(x,y)\, u(y)\, do_y$$

and the kernel function $k(x,y)$ having a singular behavior only for $y = x$. If $\lambda = 0$, the integral equation is of *first* kind, while, for $\lambda \not\equiv 0$, we have an integral equation of *second* kind. The kernel function may have a non-integrable singularity and the integral has to be understood in the sense of Hadamard.

Definition 1 *Let $k(x,y)$ be singular for $x = y$ and regular otherwise. Define the function $J(\epsilon, x)$ by*

$$J(\epsilon, x) := \int_{\Gamma - B_\epsilon(x)} k(x,y)\, u(y)\, do_y,$$

where $B_\epsilon(x)$ denotes a ball with radius ϵ about x. If the integral equation corresponds to an elliptic boundary value problem it can be shown (see [11], [18]) that $J(\epsilon, x)$ admits an uniquely determined expansion of the form

$$J(\epsilon, x) = A_0(x) + A_1(x) \log \epsilon + \sum_{j=2}^{m} A_j \epsilon^{-j+1} + o(1).$$

The part-fini integral is defined by

$$p.f. \int_\Gamma k(x,y)\, u(y)\, do_y := A_0(x).$$

2.2 Kernel properties of boundary integral operators

The kernel functions which arise in boundary integral equations are suitable Gâteau derivatives of the singularity function of the underlying elliptic boundary value problem. If the differential equation is elliptic of even order $2m$ in $\mathbf{R}^3$ with constant coefficients, the singularity function can be written in the form

$$s(x,y) = \frac{f(x-y)}{\|x-y\|^{s+t}}.$$

The function $f(z)$ can be expanded as a Taylor series of the form

$$f(z) = \sum_{|\alpha|=t}^{\infty} c_\alpha z^\alpha,$$

while t is even. The order of the singularity s is given by

$$s = 3 - 2m.$$

Here and in the following, we are using the conventions for multi-indices $\alpha \in \mathbf{N}_0^3$:

$$\begin{aligned}
|\alpha| &= \alpha_1 + \alpha_2 + \alpha_3, \\
z^\alpha &= z_1^{\alpha_1} z_2^{\alpha_2} z_3^{\alpha_3}, \\
\alpha! &= a_1! \cdot \alpha_2! \cdot \alpha_3!, \\
\left(\begin{array}{c} \nu \\ \mu \end{array} \right) &= \frac{\nu!}{(\nu-\mu)!\mu!}, \\
\sum_{|\nu|=0}^{m} c_\nu &= \sum_{s=0}^{m} \sum_{\nu_1=0}^{s} \sum_{\nu_2=0}^{s-\nu_1} c_{\nu_1,\nu_2,s-\nu_1-\nu_2}, \\
\sum_{\nu=\mathbf{0}}^{\mu} c_\nu &= \sum_{\nu_1=0}^{\mu_1} \sum_{\nu_2=0}^{\mu_2} \sum_{\nu_3=0}^{\mu_3} c_{\nu_1,\nu_2,\nu_3}.
\end{aligned}$$

We restrict our presentation to the case that the kernel function is given by normal derivatives of the singularity function, i.e.,

$$k(x,y) := \left(\frac{\partial}{\partial n_x} \right)^{\alpha_1} \left(\frac{\partial}{\partial n_y} \right)^{\alpha_2} s(x-y), \tag{2}$$

while more general cases can be treated analogously. The order of the arising derivatives satisfies $\alpha_1 + \alpha_2 \leq 2m$. We assumed that Γ is piecewise smooth, therefore, all kernel functions can be written in the form

$$k(x,y) = \frac{\tilde{f}(x,y,y-x)}{\|y-x\|^{s+t}}, \tag{3}$$

where $\tilde{f}$ can be expanded as a Taylor series with respect to the third variable:

$$\tilde{f}(x,y,z) = \sum_{|\alpha|=t}^{\infty} c_\alpha(x,y) z^\alpha.$$

The coefficients $c_\alpha(x,y)$ are piecewise smooth, dependent on the smoothness of the surface. If the surface is piecewise flat, i.e. a polygon, then, these coefficients are piecewise constant. The order of the singularity s in (3) satisfies

$$s \leq 3.$$

Remark 1 *We state that the panel-clustering algorithm, which will be presented below, can be applied also to kernels containing logarithmic singularities and to systems of integral equations as well.*

2.3 Discretization of boundary integral equations

Let $T := \{\Delta_1, \Delta_2, \ldots, \Delta_{NP}\}$ denote a panelization of the surface Γ which has to be regular in the sense that all Δ_j are smooth images of an open triangular or square master element $\hat{\Delta} \subset \mathbf{R}^2$ and the following conditions are fulfilled. If $i \neq j$ the intersection $\bar{\Delta}_i \cap \bar{\Delta}_j$ is either empty, an edge or a vertex and $\overline{\bigcup_{j=1}^{NP} \Delta_j} = \Gamma$.

Let $\mathcal{S}^{k,p} \subset \mathcal{C}^k(\Gamma)$ be the usual finite element space of dimension N consisting of polynomials of degree p on the master element which are lifted on the surface Γ by the local parametrization. Let $\Theta := \{x_j\}_{1 \leq j \leq N}$ be a set of unisolvent nodal points meaning that the interpolation problem:

find $u \in \mathcal{S}^{k,p}$ such that $u(x_j) = f_j$ for all $x_j \in \Theta$

has a unique solution for all $f \in \mathbf{R}^N$. Let further $\{\varphi_j\}_{1 \leq j \leq N}$ denote the local nodal basis of $\mathcal{S}^{k,p}$.

The *collocation method* to solve (1) is given by seeking $u_N \in \mathcal{S}^{k,p}$ such that

$$\lambda(x_j)\, u_N(x_j) - K[u_N](x_j) = r(x_j), \ \forall 1 \leq j \leq N. \tag{4}$$

The *Galerkin method* is defined by finding $u_N \in \mathcal{S}^{k,p}$ such that

$$(\lambda u_N, \varphi_j)_0 - (K[u_N], \varphi_j)_0 = (r, \varphi_j)_0, \quad \forall 1 \leq j \leq N. \tag{5}$$

The symbol $(\cdot,\cdot)_0$ denotes the $\mathcal{L}^2$-scalar product on the surface Γ. For the *Nyström method* the integral is replaced by a quadrature method, yielding

$$\lambda u(x_i) = \sum_{j=1}^{N} \omega_{i,j} k(x_i, x_j)\, u(x_j) = r(x_i), \ \forall x_i \in \Theta. \tag{6}$$

If the kernel function contains singularities suitable modifications are necessary.

3 The panel clustering method for Fredholm integral equations

3.1 Standard matrix technique to solve the discrete equations

For functions $u \in \mathcal{S}^{k,p}$, we introduce the basis representation

$$u(x) = \sum_{i=1}^{N} \mathbf{u}_i \varphi_i(x). \tag{7}$$

The discrete problems (4), (5) and (6) are equivalent to solve the system of linear equations

$$\mathbf{K}\mathbf{u} = \mathbf{r}$$

and identifying the solution $\mathbf{u}$ with $u(x)$ by (7). For the collocation method, the system matrix is given by

$$\mathbf{K}_{i,j} = \lambda(x_i)\,\varphi_j(x_i) + K[\varphi_j](x_i)$$

and the right hand side by

$$\mathbf{r}_i = r(x_i).$$

For the Galerkin method the coefficient matrix is defined by

$$\mathbf{K}_{i,j} := (\varphi_i, \lambda\varphi_j + K[\varphi_j])_0$$

and the right hand side by

$$\mathbf{r}_i = (r, \varphi_i)_0.$$

Finally, for the Nyström method the system matrix is given by

$$\mathbf{K}_{i,j} = \lambda(x_i)\,\delta_{i,j} + \omega_{i,j}k(x_i, x_j),$$

where suitable modifications apply if k is singular for $x_i = x_j$.

Here, we will not work out the panel clustering algorithm for the Nyström method. The application of this technique for Nyström discretization is straightforward. In this case, also the multipole algorithm of Rokhlin [15] or the interpolation technique of Brand and Lubrecht [1] could be used. As pointed out in the introduction, a drawback of Nyström's method is that the stability (invertibility) is not guaranteed, if systems of integral equations, mixed boundary conditions, non-linear integral equations or hypersingular formulations are considered.

From the numerical point of view the main properties of integral equations are:

- the system matrix is full,
- the condition number of the coefficient matrix is usually small.

For iterative solvers, the matrix $\mathbf{K}_{i,j}$ is not required explicitly, but only the result of

$$\begin{array}{ll} K[u](x_i) & (collocation) \\ (\varphi_i, K[u])_0 & (Galerkin). \end{array} \tag{8}$$

It follows that, in order to obtain a fast solver for BEM-equations, it is necessary to

- develop efficient cubature methods for singular and nearly singular surface integrals,
- adapt iterative solvers (e.g. multi-grid) to solve the linear system,

- apply sparse matrix techniques to approximate (8).

The general properties of kernel functions arising in boundary integral equations are developed in [18]. Efficient cubature techniques for singular integrals arising in collocation BEM are presented in [19], [4] and [3]. The nearly singular case is treated in [9]. For the Galerkin method in 3-d, efficient cubature techniques were developed in [17] and [10]. Iterative solvers for integral equations are presented in [5] and [6].

In our paper, we will concentrate on sparse matrix techniques for boundary element methods.

3.2 The Panel Clustering Method

3.2.1 Preliminaries

The panel clustering algorithm was developed by Hackbusch and Nowak for the collocation method (see [7]). A complete analysis of this algorithm was presented in [8]. In [16], the procedure was optimized depending on the magnitude of the considered problem. This algorithm was generalized for the Galerkin method in [17] and [10].

The idea is the following. The kernel function behaves singularly only in the nearfield meaning $\operatorname{dist}(x,y)$ is "small" and has an increasingly simpler behavior in the farfield, i.e., $\operatorname{dist}(x,y)$ is "large". Nonetheless the *formal description* of the kernel function in the farfield is still rather involved. Thus, we may approximate the kernel function by a function having a preferable numerical behavior compared to $k(x,y)$. In view of the explanations in Section 2.2, we assume that the singularity function in $\mathbf{R}^3$ of the differential operator can be written in the form

$$s(y-x) = \sum_{|\nu|\geq t} c_\nu \frac{(y-x)^\nu}{\|y-x\|^{s+t}}.$$

Let us assume that $z_0 = y_0 - x_0$ is large. Consequently, $s(z) = s(y-x)$ is smooth in a neighborhood $z \in \mathcal{U}(z_0)$. We expand $s(z)$ as a series of order m about z_0 :

$$s(z) = \sum_{|\nu|=0}^{m-1} \sum_{\mu=\mathbf{0}}^{\nu} \kappa_{\nu,\mu-\nu}(z_0)\, \gamma_\nu(x)\, \gamma_{\mu-\nu}(y) + R_m(z_0, z). \tag{9}$$

Here, $\kappa_{\nu,\mu}$ denote the expansion coefficients and $\{\gamma_\nu\}_{\nu\in\mathbf{N}_0^3}$ the expansion system. From (9), one obtains expansions of the kernel function which appears in the boundary integral operators described in Section 2.2.

$$k(x,y) = \underbrace{\sum_{|\nu|=0}^{m-1} \sum_{\mu=\mathbf{0}}^{\nu} \kappa_{\nu,\mu-\nu}(z_0) \left(\frac{\partial}{\partial n_x}\right)^{\alpha_1} \gamma_\nu(x) \left(\frac{\partial}{\partial n_y}\right)^{\alpha_2} \gamma_{\mu-\nu}(y)}_{=:k_m(x,y)} + \frac{\partial^{\alpha_1+\alpha_2}}{\partial n_x^{\alpha_1} \partial n_y^{\alpha_2}} R_m(z_0, y-x).$$

Example 2 *Let* $s(z) = \frac{1}{\|z\|}$ *which is (up to a constant factor) the singularity function of the 3-d Laplace operator. Taylor expansion about* $z_0 = y_0 - x_0$ *yields*

$$s(y-x) = s(z) = \sum_{|\nu|=0}^{m-1} \underbrace{\frac{1}{\nu!} \frac{\partial^{|\nu|}}{\partial z^\nu} \frac{1}{\|z\|}\bigg|_{z=z_0}}_{\kappa_\nu(z_0)} (y-x-z_0)^\nu + R_m(z_0, z).$$

The sum can be reordered as

$$\sum_{|\nu|=0}^{m-1} \sum_{\mu=0}^{\nu} \kappa_{\nu,\mu-\nu}(z_0)\, y^\mu x^{\nu-\mu}, \tag{10}$$

resulting in an expansion of the form (9).

The expansion for the kernel function $k(x, y-x) := \frac{\partial}{\partial n_y} s(y-x)$ *would be*

$$k(x, y-x) = \sum_{|\nu|=0}^{m-1} \sum_{\mu=0}^{\nu} \kappa_{\nu,\mu-\nu}(z_0) \langle n(y), \nu \rangle\, y^{\nu-1} x^{\mu-\nu} + \frac{\partial}{\partial n_y} R_m(z_0, y-x). \tag{11}$$

Remark 2 *(a) Expansion (10) is six-dimensional containing* $\frac{m^6}{720} + O(m^5)$ *indices. We state that the expansion order will be typically very small* $m \in \{1,2,3,4\}$. *The following table shows the number of summation terms in these cases.*

m	1	2	3	4
# indices	1	7	28	84

(b) In the case of the collocation method, the expansion can be simplified, resulting in a three-dimensional expansion of the form:

$$s(z) = \sum_{|\nu|=0}^{m-1} \kappa_\nu(x_i, z_0)\, y^\nu + R_m(z_0, y - x_i). \tag{12}$$

The number of summation terms are given by

m	1	2	3	4
# indices	1	4	10	20

.

Remark 3 *(a) If the direct method is used to transfer a boundary value problem of second order into an integral equation, the equation can be written in the following from*

$$u(x) = \int_\Gamma \gamma_1 s(x-y) \frac{\partial}{\partial n} u(y) + \gamma_2 u(y) \frac{\partial}{\partial n_y} s(x-y)\, dy, \qquad \forall x \in \Gamma$$

with some constants γ_1, γ_2. *Therefore, one has to discretize both the singularity function and the Gâteau derivative of it. In view of (10) and (11), it is obvious*

that one has to compute and store only the expansion coefficients $\kappa_{\nu,\mu-\nu}$, which correspond to the singularity function. Then, the coefficients of the Gâteau derivative can be computed with only a few multiplications. This remark applies also for the potential ansatz method and for general elliptic boundary value problems.

(b) For systems of integral equations, it is sufficient in many cases, to compute the expansion coefficients of scalar kernel functions to derive the coefficients for the arising kernel matrix. For the Lamé equations the fundamental matrix is given by

$$S(z) = \frac{\lambda + 3\mu}{8\pi(\lambda + 2\mu)} \left(\frac{1}{\|z\|} I + \frac{\lambda + \mu}{\lambda + 3\mu} \mathbf{D}_2 \|z\| \right).$$

Here, $\mathbf{D}_2$ denote the Hesse matrix: $(\mathbf{D}_2)_{i,j} = \frac{\partial^2}{\partial z_i \partial z_j}$. Therefore, it is sufficient to compute (and to store) the expansions of $\frac{1}{\|z\|}$ and $\|z\|$.

In order to explain the panel clustering technique, we have to introduce the following definitions.

Definition 3 *Let $T_h := \{\Delta_1, \Delta_2, \ldots, \Delta_{NP}\}$ denote a partitioning of the boundary Γ into panels. A "cluster" is the union of one or more panels:*

$$\tau = \bigcup_{j=1}^{q} \Delta_{n_j}.$$

The size of a cluster is given by the "cluster radius" $\rho(\tau)$ which is defined by the radius of the minimal ball which contains τ. The center z_τ of this ball is called "cluster center".

Definition 4 *Let the relative distance of a cluster from the singularity be defined by*

$$d_i(\tau) := \begin{cases} \frac{\rho(\tau)}{\operatorname{dist}(x_i, z_\tau)} & \textit{collocation}, \\ \\ \frac{\rho(\tau)}{\operatorname{dist}(\operatorname{supp}\varphi_i, z_\tau)} & \textit{Galerkin}. \end{cases}$$

Let the kernel function satisfy

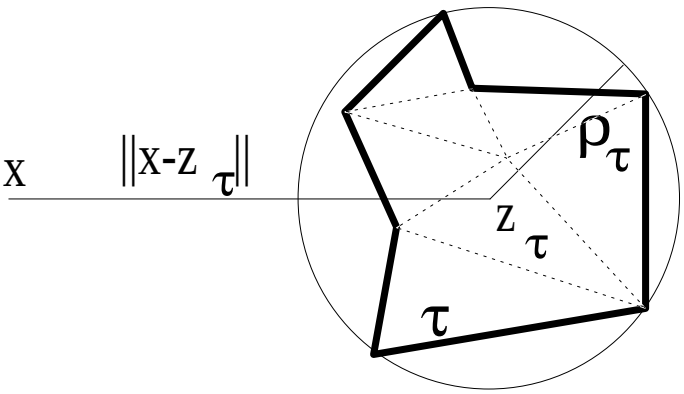
$$|k(x,y)| \leq C \|x - y\|^{-s}.$$

For given $\epsilon > 0$ and expansion order m, a cluster τ is called "admissible" with respect to an index i if

$$|k(x,y) - k_m(x,y)| \leq \epsilon \frac{1}{\|x - y\|^s}, \quad \forall y \in \tau \text{ and } x \begin{cases} = x_i & \textit{collocation}, \\ \in \operatorname{supp}\varphi_i & \textit{Galerkin}. \end{cases} \tag{13}$$

We recall here that the expansion k_m depend on τ and x_i (resp. φ_i).

The situation described above is illustrated in the following figure.



The dotted lines show the triangles which are clustered together. The cluster center is z_τ and the "singular" point x.

Assumption 5 *For given ϵ and expansion order m, there exists $0 < \eta < 1$ such that, for all $1 \leq i \leq N$, the condition*

$$d_i(\tau) \leq \eta < 1 \tag{14}$$

implies that τ is admissible with respect to i.

Definition 6 *Let ϵ and an expansion order $m \in \mathbf{N}$ be given. Let the relative size of the admissible clusters η be determined as explained above. A set of clusters $\{\tau_1, \tau_2, \ldots, \tau_k\}$ with disjoint interiors is called a covering of Γ if $\tau_1 \cup \tau_2 \cup \ldots \cup \tau_k = \Gamma$.*

A covering is called admissible with respect to an index $i \in \{1, 2, \ldots, N\}$ if either

$$d_i(\tau) < \eta \text{ (admissible cluster)}$$

or

$$\tau \text{ is a panel.}$$

The admissible covering which contains a minimal number of clusters is called minimal admissible covering C_i. The nearfield C_i^{near} and the farfield C_i^{far} are defined by

$$C_i^{near} := \{\tau \in C_i \mid \tau \text{ is non-admissible with respect to "}i\text{"}\},$$
$$C_i^{far} := \{\tau \in C_i \mid \tau \notin C_i^{near}\}.$$

3.2.2 The Algorithm

For the collocation method, a matrix vector multiplication can be written in the form

$$\sum_{j=1}^{N} \mathbf{K}_{i,j}\mathbf{u}_j = K\left[u_N\right](x_i) = \int_\Gamma k\left(x_i, y, y-x_i\right) u_N\left(y\right) d\Gamma_y. \tag{15}$$

In the following , we will use the summation convention:

$$\sum_{j\in C_i^{near}} c_j := \sum_{j:\operatorname{supp}\varphi_j \cap C_i^{near} \neq \emptyset} c_j$$

Splitting the surface in the nearfield and farfield part and using (12) shows that (15) equals

$$\begin{aligned}
& \sum_{j\in C_i^{near}} \mathbf{u}_j \underbrace{\int_{C_i^{near}} k\left(x_i, y, y-x_i\right)\varphi_j\left(y\right) dy}_{=:\mathbf{K}_{i,j}^{near}} + \\
& + \sum_{\tau\subset C_i^{far}} \int_\tau k\left(x_i, y, y-x_i\right) u_N\left(y\right) dy \\
\approx & \sum_{j\in C_i^{near}} \mathbf{K}_{i,j}^{near}\mathbf{u}_j \\
& + \sum_{\tau\subset C_i^{far}} \sum_{|\nu|=0}^{m-1} \kappa_\nu\left(x_i, z_\tau\right) \underbrace{\int_\tau \frac{\partial^{\alpha_2}\gamma_\nu\left(y\right)}{\left(\partial n_y\right)^{\alpha_2}} u_N\left(y\right) dy}_{=:J_\tau^\nu(u_N)} .
\end{aligned}$$

We summarize that a matrix vector multiplication is approximated with panel clustering by

$$\sum_{j=1}^{N} \mathbf{K}_{i,j}\mathbf{u}_j = \sum_{j\in C_i^{near}} \mathbf{K}_{i,j}^{near}\mathbf{u}_j + \sum_{\tau\subset C_i^{far}} \sum_{|\nu|=0}^{m-1} \kappa_\nu\left(x_i, z_\tau\right) J_\tau^\nu\left(u_N\right). \tag{16}$$

The matrix $\mathbf{K}^{near}$ is called "nearfield matrix" and is sparse in the sense that $\operatorname{supp}\varphi_j \cap C_i^{near} = \emptyset$ implies that $\mathbf{K}_{i,j}^{near} = 0$. The coefficients κ_ν are called expansion coefficients and $J_\tau\left(u_N\right)$ denote the "farfield coefficients".

For the Galerkin method, a matrix vector multiplication is approximated by the panel clustering method in the following way.

$$\begin{aligned}
\sum_{j=1}^{N} \mathbf{K}_{i,j}\mathbf{u}_j &= \left(K\left[u_n\right], \varphi_i\right)_0 = \int_\Gamma \varphi_i\left(x\right) p.f. \int_\Gamma k\left(x, y, y-x\right) u_N\left(y\right) dydx \\
&= \int_{\operatorname{supp}\varphi_i} \varphi_i\left(x\right) p.f. \int_{C_i^{near}} k\left(x, y, y-x\right) u_N\left(y\right) dydx \\
&\quad + \int_{\operatorname{supp}\varphi_i} \varphi_i\left(x\right) \int_{C_i^{far}} k\left(x, y, y-x\right) u_N\left(y\right) dydx
\end{aligned}$$

$$
\begin{aligned}
&\approx \sum_{j\in C_i^{near}} \mathbf{u}_j \underbrace{\int_{\operatorname{supp}\varphi_i} \varphi_i(x)\, p.f. \int_{C_i^{near}} k(x,y,y-x)\,\varphi_j(y)\,dydx}_{=:\mathbf{K}_{i,j}^{near}} \\
&\quad + \sum_{\tau\in C_i^{far}} \sum_{\nu=0}^{m-1}\sum_{\mu=0}^{\nu} \kappa_{\mu,\nu-\mu}(z_\tau - x_i) \int_{\operatorname{supp}\varphi_i} \varphi_i(x) \frac{\partial^{\alpha_1}\gamma_{\nu-\mu}(x)}{(\partial n_x)^{\alpha_1}}dx \int_\tau u_N(y) \frac{\partial^{\alpha_2}\gamma_\mu(y)}{(\partial n_y)^{\alpha_2}}dy \\
&= : \sum_{j\in C_i^{near}} \mathbf{K}_{i,j}^{near}\mathbf{u}_j + \sum_{\tau\in C_i^{far}} \sum_{\nu=0}^{m-1}\sum_{\mu=0}^{\nu} \kappa_{\mu,\nu-\mu,i} J_\tau^\mu(u_N).
\end{aligned}
$$

The form of the Galerkin version of the panel clustering method looks very similar as the collocation version. However, asymptotically, it is obvious that the Galerkin version is more expensive. The second sum contains $O\left(m^6 \cdot \#C_i^{far}\right)$ summation terms, while the sum for the collocation version only contains $O\left(m^3 \cdot \#C_i^{far}\right)$ terms.

3.2.3 Structuring of the algorithm

The algorithm consists of three phases. First, one has to choose the parameters which determine the accuracy of the panel clustering algorithm, namely, the expansion order m and the relative size of the admissible clusters η. These quantities has to be chosen such that the consistency error is adapted to the consistency requirement (approximation quality) of the whole discretization process. The details will be explained later. Phase II replaces the generation of the system matrix for the standard matrix algorithms. Here, the nearfield matrix and the expansion coefficients are computed and stored. Finally, in Phase III, a matrix vector multiplication is approximated by the panel clustering method as explained above. For the collocation method, this results in the following algorithm:

Collocation Version

Phase I:

- Choose the accuracy ϵ of the kernel approximation in (13) according to the required precision.
- Choose the expansion order m and relative size of the cluster $\eta(\epsilon, m)$ such that 5 is satisfied and the storage/computational consumptions are minimal.
- Organize the cluster in a binary tree structure.
- For each index $i \in \{1, 2, \ldots, N\}$, compute the near- and farfield of the minimal coverings.

Phase II:

- Compute the nearfield matrix $\mathbf{K}_{i,j}^{near}$
- Compute the expansion coefficients $\kappa_\nu(x_i, z_\tau)$.
- Compute the basis farfield coefficients for all $\Delta \in T$

$$J_\Delta^\nu(\varphi_j) := \int_\Delta \varphi_j(y) \frac{\partial^{\alpha_2} \gamma_\nu(y)}{(\partial n_y)^{\alpha_2}} d\Delta_y.$$

Phase III:

- Compute the farfield coefficients

$$J_\tau^\nu(u_N) = \int_\tau u_N(y) \frac{\partial^{\alpha_2} \gamma_\nu(y)}{(\partial n_y)^{\alpha_2}} d\tau_y$$

- Approximate a matrix vector multiplication by (16)

For the Galerkin version, Phase I coincides with the collocation version. Phase II and III take the following form.

Galerkin Version

Phase II:

- Compute the nearfield matrix $\mathbf{K}_{i,j}^{near}$
- Compute the basis farfield coefficients

$$J_\Delta^\nu(\varphi_j) := \int_\Delta \varphi_j(y) \frac{\partial^{\alpha_2} \gamma_\nu(y)}{(\partial n_y)^{\alpha_2}} dy$$

$$\tilde{J}_\Delta^\nu(\varphi_j) := \int_\Delta \varphi_j(x) \frac{\partial^{\alpha_1} \gamma_\nu(x)}{(\partial n_x)^{\alpha_1}} dx.$$

- Compute the expansion coefficients $\kappa_{\mu,\nu-\mu}(z_\tau - x_i)$.

Phase III:

- Compute the farfield coefficients

$$J_\tau^\nu(u_N) = \int_\tau u_N(y) \frac{\partial^{\alpha_2} \gamma_\nu(y)}{(\partial n_y)^{\alpha_2}} d\tau_y.$$

$$\kappa_{\mu,\nu-\mu,i} := \kappa_{\mu,\nu-\mu}(z_\tau - x_i) \sum_{\Delta \subset \operatorname{supp} \varphi_i} \tilde{J}_\Delta^{\nu-\mu}(\varphi_i)$$

- Approximate a matrix vector multiplication by

$$\sum_{j=1}^N \mathbf{K}_{i,j} \mathbf{u}_j \approx \sum_{j \in C_i^{near}} \mathbf{K}_{i,j}^{near} \mathbf{u}_j + \sum_{\tau \in C_i^{far}} \sum_{\nu=0}^{m-1} \sum_{\mu=0}^{\nu} \kappa_{\mu,\nu-\mu,i} J_\tau^\mu(u_N).$$

Remark 4 *If the Galerkin method is employed, we recommend to use, if possible, an integral equation with a symmetric kernel, i.e., $\alpha_1 = \alpha_2$ in (2). Then $J_\Delta^\nu(\varphi_i) = \tilde{J}_\Delta^\nu(\varphi_i)$ and the computation and storage amount is considerably smaller.*

4 Efficient algorithms for the micro steps of the Panel Clustering algorithm

In order to make the panel clustering efficient, fast micro algorithms have to be developed for the different steps in the algorithms described above. We will present here some of them exemplarily, while a detailed description could be found in [16] and [17].

4.1 Construction of the cluster tree, minimal coverings and the cluster radius

In Phase I of the algorithm, one has to compute minimal coverings. Hence, one has to organize the clusters in a tree structure which allows to choose minimal coverings efficiently out of this reservoir. From the viewpoint of the complexity, binary trees are optimal. In this case, the numbers of cluster is bounded by $2NP$, where NP denotes the number of panels. In many cases fine grids are generated by refining coarser grids, resulting in a nested hierarchy of grids. These grids can directly be used as a cluster tree. Using this strategy, every triangle (cluster), usually, has four sons. In order to obtain a binary grid, one can define intermediate grids, where always two of the four sons a clustered together.

It also possible to construct a cluster tree directly from the fine grid in an efficient way (see [8]).

We state that, for the construction of the clusters, no regularity assumptions apply, as, e.g., for the generation of triangulations. Clusters are allowed to be neither connected nor of regular shape.

After the cluster tree is generated, it is possible to construct for every matrix line "i" minimal coverings by the following procedure "Divide" (cf. [8]).

$C_i^{near} = \emptyset; C_i^{far} = \emptyset;$

$Divide\left(\Gamma, C_i^{near}, C_i^{far}\right)$; **comment** result is near- and farfield w.r.t. "i".

procedure Divide $\left(\tau, C_i^{near}, C_i^{far}\right)$

begin

if τ is admissible w.r.t "i" **then** $C_i^{far} := C_i^{far} \cup \tau$ **else**

if τ is a panel **then** $C_i^{near} := C_i^{near} \cup \tau$ **else**

begin determine sons $\{\tau_{n_i}\}_{1 \le i \le q}$ of τ

for $i := 1$ **to** q **do** $Divide\left(\tau_q, C_i^{near}, C_i^{far}\right)$

end **end**;

We readily state that, due to the assumption that the cluster tree is binary, the complexity of the algorithm is $2\left(\#C_i^{near} + \#C_i^{far}\right)$ arithmetic operations. In order to check, whether a cluster is admissible with respect to an index i, we need the radius and the center of every cluster (cf. Definition 4). The definition of the

cluster radius can be written in the form

$$\rho_\tau^2 := \min_{z \in \mathbf{R}^3} \underbrace{\max_{1 \le n \le q} \left\{ \|z - x_n\|^2 ; x_n \text{ is a vertix of the convex hull of } \tau \right\}}_{F_\infty(z)}.$$

To compute the minimum above we replace F_∞ by

$$F_p(z) := \left\{ \sum_{n=1}^{q} \|z - x_n\|^p \right\}^{2/p}$$

The minimum of $\tilde{\rho}_\tau := \min_{z \in \mathbf{R}^3} F_p(z)$ can be computed easily and robust by means of Newton's method, since F_p is convex and smooth. Note that the cluster center is computed by this method, too.

4.2 Computation of the farfield coefficients

The farfield coefficients are defined by

$$J_\tau^\nu(u_N) := \int_\tau u_N(y) \frac{\partial^\alpha \gamma_\nu(y)}{(\partial n_y)^\alpha} d\tau_y.$$

We first compute the basis farfield coefficients

$$J_\Delta^\nu(\varphi_i) := \int_\tau \varphi_i(y) \frac{\partial^\alpha \gamma_\nu(y)}{(\partial n_y)^\alpha} d\Delta_y. \tag{17}$$

There are several strategies to compute these integrals. Usually, there are recursion formulae for the function system γ_ν and $\frac{\partial^\alpha \gamma_\nu(y)}{(\partial n_y)^\alpha}$. Computing (17) for the "lower indices" by some quadrature method, the recursion could be used. Asymptotically, i.e., if the number of multi indices ν is large, this formulae are of optimal complexity, because each index ν requires $O(1)$ operations.

An alternative to this method will be explained in the following. For triangles or quadrilaterals, there are quadrature formulae which integrate polynomials up to a certain degree exactly using a minimal number of function evaluation (cf. [20]). These formulae can be used to approximate the integrals (17). It turns out that this method is much cheaper for practical problem sizes, where the expansion order m is small and far away from the asymptotically gain of the recursive method.

The derivatives of the kernel function for the computations of the expansion coefficients κ can be done recursively. The amount per multi-index is $O(m)$, with m denoting the expansion order. To avoid here too many technicalities, we refer for an explicit description to [16] and [17].

5 Error Analysis for the Panel Clustering Method

Using the panel clustering method to approximate a matrix vector multiplication, introduces an additional error in the discretization of the continuous problem. The parameters which determines the precision of the panel clustering have to be adapted to the required asymptotic accuracy of the approximation. The error analysis consists of two parts. First, one has to choose a suitable expansion system to approximate the kernel function. We have to check, how the approximation of the kernel function (13) depends on the size of the cluster and the expansion order. Then, we need an abstract error estimate ot the form that

$$|k(x,y) - k_m(x,y)| \leq \epsilon \frac{1}{\|x-y\|^2}$$

implies a consistency error estimate of the form

$$\left| K[u] - \tilde{K}[u] \right| \leq \delta |u|_\infty .$$

Here, K denotes the original integral operator and $\tilde{K}$ the integral operator, where the kernel function is replaced on every cluster by the expansion k_m.

We restrict our presentation to the case that the kernel function is the singularity function corresponding to the Laplacian (up to a constant factor), i.e.,

$$s(x,y) = \frac{1}{\|x-y\|}. \tag{18}$$

An error analysis of Taylor based expansions of general kernel functions can be found in [8], [16], [17] and [10].

Theorem 7 *Let $T_m(z_0)$ denote the 3-d Taylor expansion of $\frac{1}{\|z\|}$ about $z = z_0$ with $z_0 \neq 0$:*

$$T_m(z_0)(z) = \sum_{|\nu|=0}^{m-1} \frac{1}{\nu!} \frac{\partial^{|\nu|}}{\partial z^\nu} \frac{1}{\|z\|}\bigg|_{z=z_0} (z-z_0)^\nu .$$

Let σ denote the relative distance of z from the singularity, i.e., $\sigma = \frac{\|z-z_0\|}{\|z_0\|}$. Then,

$$\left| \frac{1}{\|z\|} - T_m(z_0) \right| \leq \frac{\sigma^m}{\|z\|}.$$

Proof. The proof is given in [16, Theorem 3.2.11]. ■

Using this Theorem, we know that, for given ϵ and expansion order m, the choice of $\eta := \sqrt[m]{\epsilon}$ has the consequence that condition (14), i.e.,

$$d_i(\tau) \leq \eta$$

with d_i from Definition 4, implies the estimate (13) for the kernel approximation.

We proceed here with an estimate of the consistency error caused by the panel clustering algorithm.

Theorem 8 *Let the kernel function $k(x,y)$ be singular of order s:*

$$|k(x,y)| \leq \frac{C}{\|x-y\|^s}, \qquad \forall x,y \in \Gamma, x \neq y.$$

We assume that k is approximated by an expansion which satisfies (13). Let all coverings be admissible and the panelization be quasi-uniform:

$$\begin{aligned} h &: = \max_{\Delta \in T} \operatorname{diam}(\Delta), \\ h &\leq C_u \rho(\Delta), \end{aligned}$$

where $\rho(\Delta)$ denotes the radius of a panel Δ (see Definition 3). Then, for the collocation method, we get the error estimate:

$$|K[u_N](x_i) - K_m[u_N](x_i)| \leq \epsilon C C_s(h) \|u_N\|_{\mathcal{L}^\infty}.$$

The operator K_m is defined by replacing the kernel function on each admissible cluster by the expansion of order m. For the Galerkin method, we obtain

$$|(\varphi_i, K[u_N] - K_m[u_N])_0| \leq \epsilon C C_s(h) \mu(\operatorname{supp} \varphi_i) \|u_N\|_{\mathcal{L}^\infty}$$

with the area measure $\mu(\omega) := \int_\omega 1 dx$. The size of the constant C_s is related to the order s of the singularity:

$$C_s(h) = C \cdot \begin{cases} 1, & if \quad s \leq 1, \\ \log(h) & if \quad s = 2, \\ h^{-1} & if \quad s = 3. \end{cases} \tag{19}$$

Proof. The proof for the collocation method is given in [8, Lemma 5.1], while the proof for the Galerkin method can be found in [17, Theorem 2.2.1]. In both cases the constant $C_s(h)$ is given by the integral

$$C_s(h) = \sup_{x \in \Gamma} \int_{\Gamma - B_{Ch}(x)} \frac{1}{\|x-y\|^s} dy,$$

where $B_r(x)$ denotes a ball with radius r about x. By introducing polar coordinates it is easy to show that this integral leads to the formula (19). ■

This theorem shows that the additional error, introduced by the panel clustering method, can be adapted to any desired accuracy, by choosing the admissible size of the clusters and the expansion order in an appropriate way. Usually, the consistency requirement is of the form

$$\epsilon \leq C h^\kappa.$$

If the kernel function k is given by (18), the condition above would imply that η and m has to be chosen such that

$$\eta^m \leq C h^\kappa. \tag{20}$$

This means that, for each expansion order $m \in \mathbf{N}$, we can find an η such that (20) is satisfied. Hence, we have the freedom to choose the pair (m, η) such that the CPU-time and the storage assumption is minimal. In [8] it is shown that the choice of

$$\begin{aligned} m &= \left\lfloor \frac{\kappa}{4} \log N \right\rfloor \\ \eta &= C^{1/m} h^{\kappa/m} = const \end{aligned} \tag{21}$$

is asymptotically optimal. In the following Section we will use these Theorems to derive estimates for the complexity of the panel clustering algorithm.

6 Asymptotic Complexity of the Panel Clustering Algorithm

The estimate of the complexity of the algorithm consists of two steps. First, we have to estimate the number of clusters in the admissible coverings, dependent on the choice of the precision ϵ and the expansion order m. Furthermore, we need estimates for the computational work of all micro steps in the algorithm, e.g., the number of operations to compute the size of a cluster, the expansion coefficients and the farfield coefficients of a cluster, etc.

In this Section we will use the notation introduced in the previous one. First, we will consider the asymptotic complexity. As explained in (21), we choose

$$\begin{aligned} m &= \left\lfloor \frac{\kappa}{4} \log N \right\rfloor \\ \eta &= const. \end{aligned}$$

It turns out that, under natural conditions on the surface Γ, which are described in detail in [8], the number of farfield panels per matrix line can be bounded by

$$\#C_i^{far} \leq CN \log N,$$

while the number of nearfield panels does not increase with smaller values of h

$$\#C_i^{near} \leq C.$$

This implies that the number of non-zero entries per line of the nearfield matrix does not increase with increasing number of unknowns.

A thoroughly investigation of the micro steps of the panel clustering algorithm shows that each farfield coefficient can be computed in $O(1)$ arithmetical operations and each expansion coefficient by $O(m)$ operations. Furthermore, each nearfield matrix elements can be computed for all integral operators which arise

for elliptic boundary value problems by $O\left(\log^3 N\right)$operations. This can be found in detail in [16, Chapter 4], [17, Chapter 4] and [9].

For the collocation method, a matrix vector multiplication was approximated by

$$\sum_{j\in C_i^{near}} \mathbf{K}_{i,j}^{near}\mathbf{u}_j + \sum_{\tau\in C_i^{far}} \sum_{|\nu|=0}^{m-1} \kappa_\nu\left(x_i, z_\tau\right) J_\tau^\nu\left(u_N\right).$$

The number of expansion coefficients per matrix line is bounded by

$$\#\left\{\kappa_\nu\left(x_i, z_\tau\right), 0 \le |\nu| \le m-1, \tau \in C_i^{far}\right\} \le C\log^4 N.$$

Here, we made use of $m = O\left(\log N\right)$. Thus,we obtain that the amount of work to compute these coefficients behaves asymptotically as $O\left(\log^5 N\right)$. The number of farfield coefficient is bounded by

$$\#\left\{J_\tau\left(u_N\right), \tau \text{ is a cluster in the cluster tree}\right\} = O\left(N\log^3 N\right),$$

thus the computational amount and the storage requirements are of order $N\log^3 N$. Since the number of non-zero entries per matrix line of $\mathbf{K}^{near}$ is independent of N, the computing time can be estimated by

$$O\left(\log^3 N\right)$$

per matrix line.

The only difference between the Galerkin method and the collocation method from the view point of the complexity is that representation of a matrix vector multiplication takes the form

$$\sum_{j\in C_i^{near}} \mathbf{K}_{i,j}^{near}\mathbf{u}_j + \sum_{\tau\in C_i^{far}} \sum_{|\nu|=0}^{m-1} \sum_{\mu=\mathbf{0}}^{\nu} \kappa_{\mu,\nu-\mu,i} J_\tau^\mu\left(u_N\right)$$

and consequently the number of expansion coefficients $\kappa_{\mu,\nu-\mu,i}$ per matrix line is of order $O\left(\log^7 N\right)$. It is too expensive to store all of these coefficients. The computations of these coefficients can be split in two parts. First, one computes and stores $O\left(\log N\right)$auxiliary quantities, say $\tilde{\kappa}_\tau$ which are costly to compute, since the evaluation the (complicated) kernel function is involved. The coefficients $\kappa_{\mu,\nu-\mu,i}$ can be computed from $\tilde{\kappa}_\tau$ very cheaply, because only a few additions and multiplications are necessary per coefficient. Obviously, this reduces the storage amount of the algorithm but increase the CPU-time of a matrix vector multiplication. The following table shows the asymptotic complexity of the panel clustering algorithm for the Galerkin and the collocation method.

	computing time		storage amount	
	Collocation	Galerkin	Collocation	Galerkin
Generation of system	$N\log^5 N$	$N\log^3 N$	$N\log^4 N$	$N\log^3 N$
Matrix vector mult.	$N\log^4 N$	$N\log^7 N$	0	0

The asymptotical gain of the panel-clustering method is obvious, compared to the standard matrix technique, where both the CPU-time and the storage consumption behaves like $O(N^2)$. Nevertheless, it is by no means clear, whether the asymptotic gain is visible for practical problem sizes. The constants in the $O(\cdot)$-estimates plays here an important role. To study this theoretically, we have counted in detail the arithmetic operations which are necessary for the panel clustering dependent on the parameters m and η. The result is depicted in Figures 2 and 3.

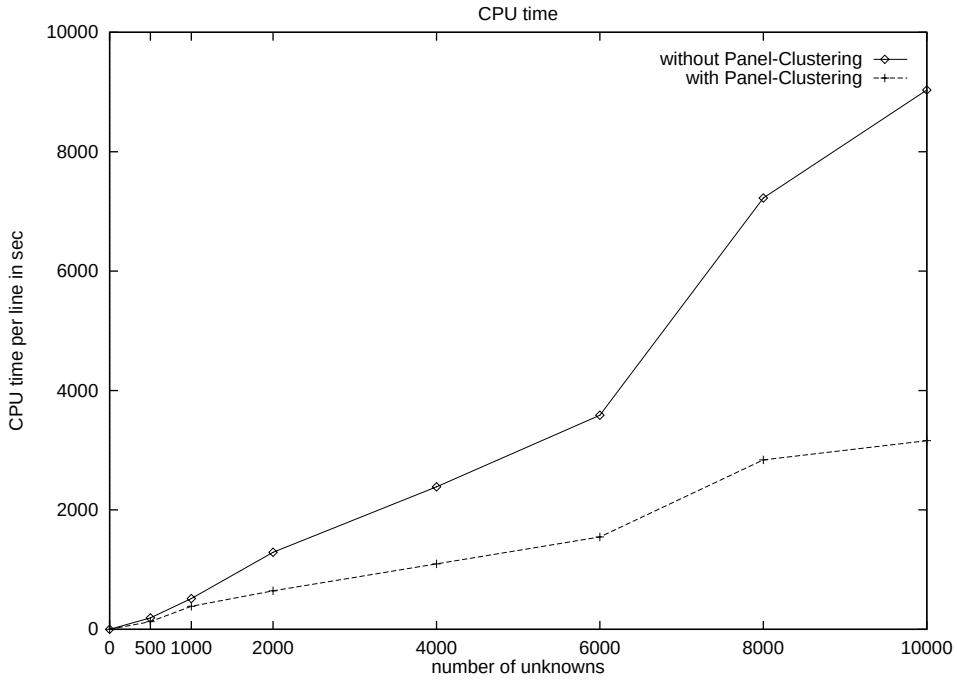


Figure 2: Comparision of the CPU time per matrix line with and without panel clustering

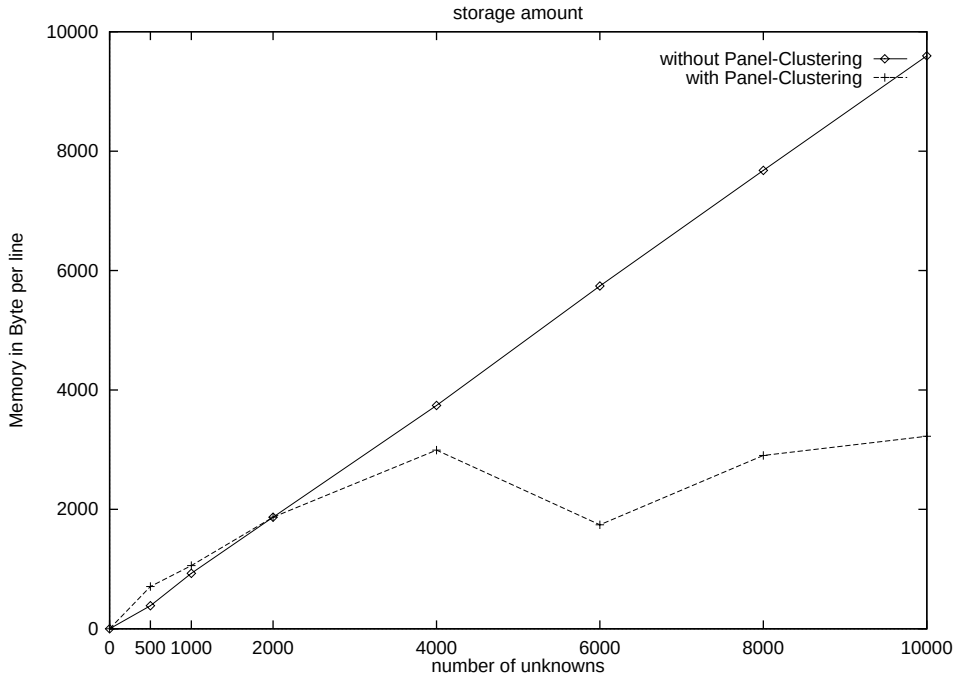


Figure 3: Comparision of the storage consumption per matrix line with and without panel clustering

The pictures underpins the efficiency of the presented method, if the dimension of the problem is large. On the other hand, it turns out that, if the number of unknowns is smaller than 1000, the standard matrix technique should be used. In complicated engineering application the number of unknowns ranges between 2000 and 10000. Sometimes, even more unknowns would be adequate for the considered problem. In that cases, the panel clustering technique could reduce the CPU-time and the storage requirement considerably.

7 Numerical Examples

Let $B_1 := \{x \in \mathbf{R}^3 \mid \|x\| = 1\}$ be the unit ball. As a test problem we consider

$$\begin{array}{lll} \Delta u = 0 & in & \mathbf{R}^3 \backslash B_1, \\ \dfrac{\partial u}{\partial n} = \dfrac{\partial}{\partial n} \dfrac{1}{\|x - x_0\|} =: g(y) & on & \partial B_1, \\ \lim\limits_{\|x\| \to \infty} u(x) = 0 & & \end{array}$$

with $x_0 = (0.9, 0, 0)^T$. The solution to this problem is given by $u(x) = \frac{1}{\|x - x_0\|}$. This function is smooth on the surface but "nearly" singular near the surface point $x = (1, 0, 0)^T$. The solution is depicted in Figure 4.

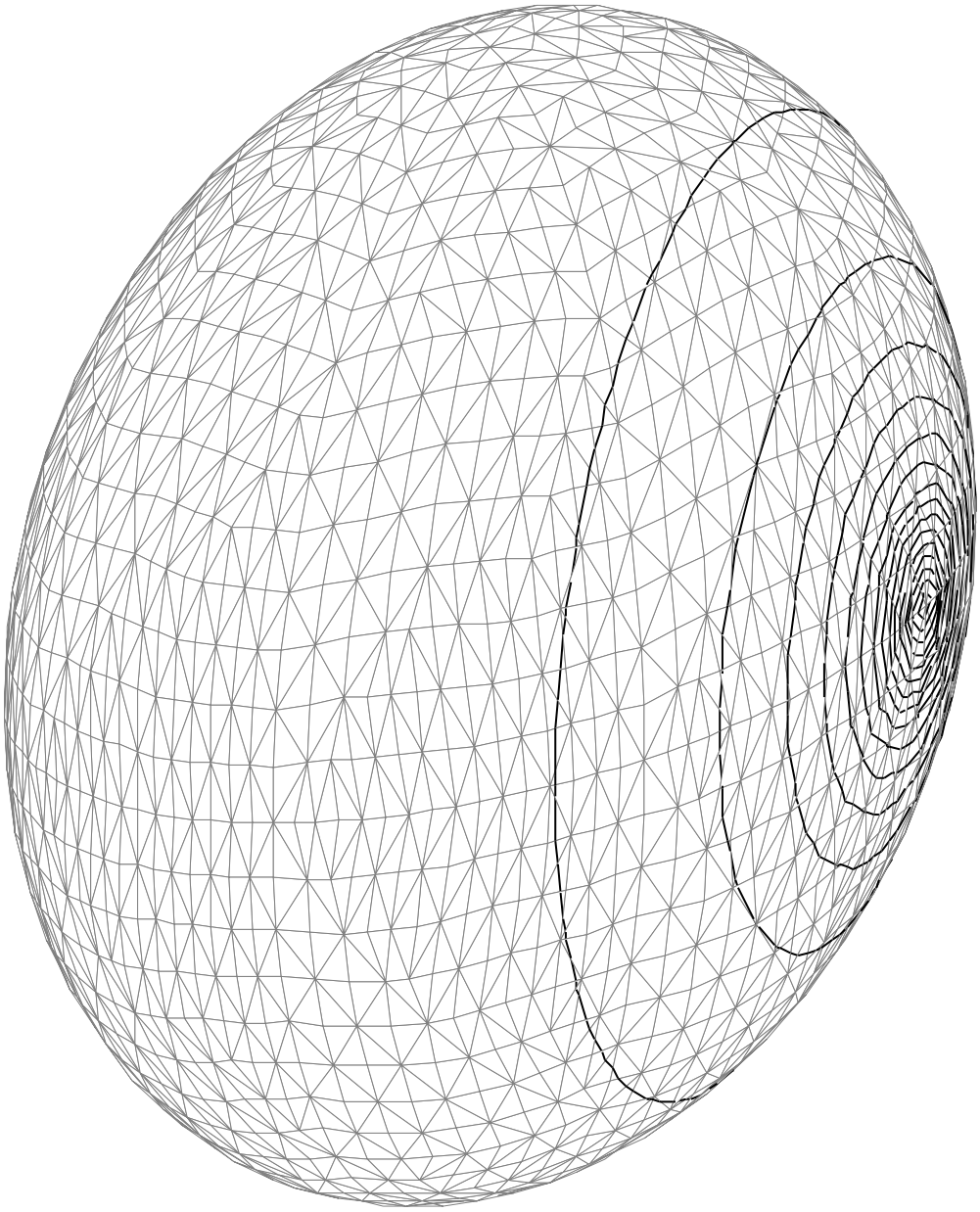


Figure 4: Solution u of the numerical test problem. It behaves nearly singular at the surface point $x = (1, 0, 0)^T$

We write this equation as a Fredholm integral equation of second kind using the direct method

$$-2\pi u(x) + \int_\Gamma \frac{\partial}{\partial n_y} \frac{u(y)}{\|x-y\|} d\Gamma_y = \int_\Gamma \frac{g(y)}{\|x-y\|} d\Gamma_y.$$

For the discretization we have approximated the surface by a polygon consisting of plane triangles which interpolates the surface in the vertices. We have used a hierarchy of five grids. The details can be found in the following table.

Level	# of triangles	# of vertices
1	16	10
2	64	34
3	256	130
4	1024	514
5	4096	2050

We have measured the error in the $\mathcal{L}^2-$norm. Let u_h denote the BEM-solution without panel clustering and $e_h := \|u - u_h\|_0$ the corresponding error. Since we are using piecewise linear elements we expect that the method converges quadratically in the $\mathcal{L}^2-$norm. This means that

$$e_h \approx \frac{e_{2h}}{4}.$$

The panel clustering method introduces an additional error in the discretization process. The parameters m and η which controls the size of this error should be tuned in such a way that the solution u_h^{PC} with panel-clustering differs only by a constant factor from e_h. This factor has to be smaller than 4, otherwise we would get the same accuracy with the standard matrix method using the grid of the coarser level. Let the ratio f be defined by

$$f = \frac{e_h^{PC}}{e_h}.$$

We have considered the cases that m and η have been chosen such that $f \in \{1, 1.5, 2, 3\}$. The panel clustering method which corresponds to $f = 1$ results in the standard matrix method, i.e., $m = 0$ and $\eta = 0$. The following tables shows the storage amount and CPU-time for the different values of the ratio f. Table 1 corresponds to the Galerkin method, while Table 2 corresponds to the collocation method. We see that, using the panel-clustering method, the CPU time for this problem size can be reduced up to factor $4 - 7.5$ and the storage amount up to a factor 7. We expect that for larger problem sizes this effect increases. This problem was chosen such that the exact solution is nearly singular. Compared to practical engineering problems, the solution is still quite unrealistic smooth. It is clear that, if the solution has a less smooth behavior, the discretization error for this problem size (about 2000 unknowns) will be substantially larger. Then, one can choose larger clusters or lower expansion orders in order to get the same additional error. The reason is that the panel clustering method replaces the kernel function by a 3-d expansion, which does not depend on the smoothness of the surface and the smoothness of the solution.

		no PC	**f = 1.5**	**f = 2**	**f = 3**
Level 2	e_h	3.38	3.21	3.21	3.21
	e_h^{PC}/e_h	1.0	0.95	0.95	0.95
	CPU-time	0.61	0.20	0.20	0.20
	CPU(1)/CPU(f)	1.0	3.0	3.0	3.0
	MEM	0.038	0.087	0.087	0.087
	MEM(1)/MEM(f)	1.0	0.43	0.43	0.43
Level 3	e_h	1.14	1.2	1.2	1.2
	e_h^{PC}/e_h	1.0	1.05	1.05	1.05
	CPU-time	8.42	1.09	1.09	1.09
	CPU(1)/CPU(f)	1.0	7.71	7.71	7.71
	MEM	0.414	0.40	0.40	0.40
	MEM(1)/MEM(f)	1.0	1.04	1.04	1.04
Level 4	e_h	1.63 E-1	2.38 E-1	3.27 E-1	3.50 E-1
	e_h^{PC}/e_h	1.0	1.46	2.0	2.14
	CPU-time	162.8	19.7	15.73	15.21
	CPU(1)/CPU(f)	1.0	8.26	9.16	10.7
	MEM	5.85	2.94	2.70	2.67
	MEM(1)/MEM(f)	1.0	2.0	2.2	2.2
Level 5	e_h	1.1 E-2	1.64 E-2	2.3 E-2	3.28 E-2
	e_h^{PC}/e_h	1.0	1.49	2.09	2.98
	CPU-time	3030	962.7	643	407.7
	CPU(1)/CPU(f)	1.0	3.15	4.71	7.43
	MEM	90.5	35.7	26.5	22.6
	MEM(1)/MEM(f)	1.0	2.53	3.41	4.0

Table 1: Consumptions of the Galerkin method with and without panel clustering. CPU(f) denotes the CPU-time, where the panel clustering parameters are chosen such that f is the relative error ratio. MEM stands for "memory amont"

		no PC	**PC**$_{f=1.5}$	**PC**$_{f=2}$	**PC**$_{f=3}$
Level 2	e_h	1.97	2.72	2.72	2.72
	e_h^{PC}/e_h	1.0	1.38	1.38	1.38
	CPU-time	0.038	0.038	0.038	0.038
	CPU(1)/CPU(f)	1.0	1.01	1.01	1.01
	MEM	0.038	0.069	0.069	0.069
	MEM(1)/MEM(f)	1.0	0.55	0.55	0.55
Level 3	e_h	0.883	1.3	1.3	1.3
	e_h^{PC}/e_h	1.0	1.47	1.47	1.47
	CPU-time	0.34	0.14	0.14	0.14
	CPU(1)/CPU(f)	1.0	2.51	2.51	2.51
	MEM	0.414	0.307	0.307	0.307
	MEM(1)/MEM(f)	1.0	1.35	1.35	1.35
Level 4	e_h	1.15 E-1	1.75 E-1	2.26 E-1	3.44 E-1
	e_h^{PC}/e_h	1.0	1.52	1.97	2.99
	CPU-time	4.63	1.87	1.53	1.29
	CPU(1)/CPU(f)	1.0	2.47	3.02	3.59
	MEM	5.85	2.48	2.26	2.14
	MEM(1)/MEM(f)	1.0	2.36	2.59	2.73
Level 5	e_h	1.51 E-2	2.25 E-2	2.96 E-2	4.48 E-2
	e_h^{PC}/e_h	1.0	1.5	1.96	2.97
	CPU-time	72.0	42.7	25.3	18.1
	CPU(1)/CPU(f)	1.0	1.69	2.85	3.97
	MEM	90.5	20.8	16.2	13.3
	MEM(1)/MEM(f)	1.0	4.35	5.59	6.8

Table 2: Consumptions of the collocation method with and without panel clustering. CPU(f) denotes the CPU-time, where the panel clustering parameters are chosen such that f is the relative error ratio. MEM stands for "memory amont'

For this test problem, both, the Galerkin and the collocation method can be applied. Then it is clear that the collocation method is the faster method. The situation would change, if we would consider negative norms or evaluate the potential in interior grid points. As pointed out in the previous sections, for more complicated equations and domains one should use the hypersingular formulation together with the Galerkin method. A numerical comparison of the efficiency of different formulations of integral equations will be addressed in a forthcoming paper (see [12]).

References

[1] A. Brandt. Multilevel computations of integral transforms and particle interactions with oscillatory kernels. In *IMACS 1st Int. Conf. On Comp. Phys., Boulder, Colorado*, June 1990.

[2] J. Elschner. The Double Layer Potential Operator over Polyhedral Domains I: Solvability in Weighted Sobolev Spaces. *Appl. Anal.*, 45:117–134, 1992.

[3] M. Guiggiani. Direct Evaluation of Hypersingular Integrals in 2D BEM. In W. Hackbusch, editor, *Proc. Of 7th GAMM Seminar on Numerical Techniques for BEM, Kiel 1991*, pages 23–34, Braunschweig, 1991. Vieweg.

[4] M. Guiggiani and A. Gigante. A General Algorithm for Multidimensional Cauchy Principal Value Integrals in the Boundary Element Method. *ASME J. Appl. Mech.*, 57:907–915.

[5] W. Hackbusch. *Integralgleichungen*. Teubner, Stuttgart, 1989.

[6] W. Hackbusch. The Solution of Large Systems of BEM Equations by the Multi-Grid and Panel Clustering Technique. *Rend. Sem. Mat. Uni. Pol. Torino, Fasc. Spec.: Numerical Methods*, pages 163–187, 1991.

[7] W. Hackbusch and Z.P. Nowak. On the Complexity of the Panel Method (in russ.). In *Proc. Of the Conference: Modern Problems in Numerical Analysis*, Moscow, Sept 1986.

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

[9] W. Hackbusch and S. A. Sauter. On Numerical Cubatures of Nearly Singular Surface Integrals arising in BEM Collocation. *Computing*, 52:139–159, 1994.

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

[11] R. Kieser, C. Schwab, and W. L. Wendland. Numerical Evaluation of Singular and Finite-Part Integrals on Curved Surfaces Using Symbolic Manipulation. *Computing*, 49:279–301, 1992.

[12] C. Lage. *A Parallel Implementation of 3-d BEM with Focus on Panel Clustering.* PhD thesis, Institut für Informatik und Praktische Mathematik, University of Kiel, 24118 Kiel, Germany, to appear in 1995.

[13] A. Rathsfeld. On Quadrature Methods for the Double Layer Potential Equation over the Boundary of a Polyhedron. *Numer. Math.*, 66:67–95, 1993.

[14] A. Rathsfeld. Nyström's Method and Iterative Solvers for the Solution of the Double Layer Potential Equation over Polyhedral Boundaries. *SIAM J. Numer. Anal.*, to appear.

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

[16] S. A. Sauter. Der Aufwand der Panel-Clustering-Methode für Integralgleichungen. Technical Report 9115, Institut für Praktische Mathematik, University of Kiel, 24118 Kiel, Germany, 1991.

[17] S. A. Sauter. *Über die effiziente Verwendung des Galerkinverfahrens zur Lösung Fredholmscher Integralgleichungen.* PhD thesis, Institut für Praktische Mathematik, University of Kiel, 24118 Kiel,Germany, 1992.

[18] C. Schwab and W.L. Wendland. Kernel Properties and Representations of Boundary Integral Operators. *Math. Nachr.*, 156:187–218, 1992.

[19] C. Schwab and W.L. Wendland. On Numerical Cubatures of Singular Surface Integrals in Boundary Element Methods. *Numerische Mathematik*, pages 343–369, 1992.

[20] A. H. Stroud. *Approximate Calculations of Multiple Integrals.* Prentice Hall, Englewood Cliffs, 1973.