INSTITUT FÜR INFORMATIK

Low-rank approximation of integral operators by using the Green formula and quadrature

Steffen Börm and Jessica Gördes

Bericht Nr. 1205
May 30, 2012
ISSN 2192-6247

CHRISTIAN-ALBRECHTS-UNIVERSITÄT

ZU KIEL

Institut für Informatik der
Christian-Albrechts-Universität zu Kiel
Olshausenstr. 40
D – 24098 Kiel

Low-rank approximation of integral operators by using the Green formula and quadrature

Steffen Börm and Jessica Gördes

Bericht Nr. 1205
May 30, 2012
ISSN 2192-6247

e-mail: sb@informatik.uni-kiel.de, jeg@informatik.uni-kiel.de

Part of this research was funded by the
Deutsche Forschungsgemeinschaft
in project BO 3289/2-1.

Low-rank approximation of integral operators by using the Green formula and quadrature

Steffen Börm and Jessica Gördes

May 30, 2012

Approximating integral operators by a standard Galerkin discretisation typically leads to dense matrices. To avoid the quadratic complexity it takes to compute and store a dense matrix, several approaches have been introduced including $\mathcal{H}$-matrices. The kernel function is approximated by a separable function, this leads to a low rank matrix. Interpolation is a robust and popular scheme, but requires us to interpolate in each spatial dimension, which leads to a complexity of m^d for m-th order. Instead of interpolation we propose using quadrature on the kernel function represented with Green's formula. Due to the fact that we are integrating only over the boundary, we save one spatial dimension compared to the interpolation method and get a complexity of m^{d-1}.

Part of this research was funded by the
Deutsche Forschungsgemeinschaft
in project BO 3289/2-1.

1 Introduction

1.1 Model problem

We consider a Fredholm integral operator of the form

$$\mathcal{G}[u](x) = \int_\Omega g(x,y)u(y)\mathrm{d}y$$

on a subdomain or submanifold Ω of $\mathbb{R}^d$.

This kind of operator occurs, e.g., in the integral equation formulation of the Poisson problem in $\mathbb{R}^2$, where g is the singularity function $g(x,y) = -\frac{1}{2\pi}\log\|x-y\|$. Discretising $\mathcal{G}$ by a standard Galerkin approach for a basis $(\varphi_i)_{i\in I}$, yields a matrix G with entries

$$G_{i,j} := \int_\Omega \int_\Omega \varphi_i(x) g(x,y) \varphi_j(y)\,\mathrm{d}x\,\mathrm{d}y\,. \tag{1}$$

In the standard case G can be expected to be a dense matrix.

Different approaches have been introduced to avoid the quadratic complexity for computing and storing dense matrices: for translation-invariant kernel functions and simple geometries, the matrix G has Toeplitz structure, which can be exploited by algorithms based on the fast Fourier transformation. Wavelet techniques can be used in order to compress the resulting dense matrix, if the underlying geometry can be described by a small number of smooth maps [9]. Techniques like the fast multipole method [14, 15, 25] and the panel clustering approach [21] use a degenerate approximation of the kernel function and can handle complicated geometries particularly well. Multipole methods without multipoles [1, 22] employ points on a curve surrounding the relevant domain to construct an efficient and accurate approximation without the need of higher derivatives.

Algebraic methods like cross approximation [24, 10, 11, 2, 4] and HSS-matrices [23] use the entries of the matrix to construct factorized approximations that are the counterpart of degenerate expansions of the kernel function.

Our approach relies on similar principles as the techniques used in [8, 13, 23], but we replace algebraic approximation by quadrature. This allows us to derive a flexible and elegant algorithm for which rigorous convergence proofs are possible. Compared to the other approaches we do not require higher derivatives or special expansions. We present the resulting method in the framework of hierarchical matrices [7, 17, 19], which translate the degenerate approximations into low rank matrices.

We stress that we are not only presenting an algorithm, but also a detailed error analysis, which is to the best of our knowledge the first one for the method.

1.2 Degenerate kernel function

To approximate the matrix G by a hierachical matrix, we need to find low rank approximations for certain sub-matrices. Because there is a close connection between the matrix G and the kernel function g, we can do this by approximating the kernel.

Let $t \times s \subseteq I \times I$ be a sub-block of the product index set. In order to get a low rank approximation for the corresponding sub-matrix $G|_{t\times s}$ we are constructing a degenerated kernel function $\tilde{g}$, i.e., functions $a_\nu : \mathbb{R}^d \to \mathbb{R}$ and $b_\nu : \mathbb{R}^d \to \mathbb{R}$ for $\nu \in \{1,\ldots,k\}$ so that

$$g(x,y) \approx \tilde{g}(x,y) = \sum_{\nu=1}^{k} a_\nu(x)\, b_\nu(y) \tag{2}$$

holds for all $x, y \in \mathbb{R}^d$. The essential characteristic of a degenerated kernel is the separation of the variables, which implies

$$\widetilde{G}_{i,j} := \sum_{\nu=1}^{k} \int \varphi_i(x)a_\nu(x)\mathrm{d}x \int \varphi_j(y)\, b_\nu(y)\, \mathrm{d}y$$

for all $i, j \in I$. The low rank matrix is therefore given by $G|_{t\times s} \approx AB^\top$ with matrices $A \in \mathbb{R}^{t\times k}$ and $B \in \mathbb{R}^{s\times k}$ as follows:

$$A_{i\nu} := \int \varphi_i(x)a_\nu(x)\,\mathrm{d}x \quad \text{and} \quad B_{j\nu} := \int \varphi_j(y)b_\nu(y)\,\mathrm{d}y\,.$$

If k is small, this is a very efficient way of storing a low rank matrix because we have $k(\#t + \#s)$ matrix entries instead of $\#t \cdot \#s$.

Typically we can find a degenerated kernel as in (2) only in an adequate distance to the singularity. So we need a criterion which tells us when we are far enough from the singularity and thereby which sub-matrices can be approximated with low rank matrices. This criterion is the so-called *admissibility condition* which is defined according to the nature of the considered problem and the kernel approximation.

1.3 Interpolation

One of the standard techniques to get a degenerated kernel is via Lagrange interpolation as described in [6]. The interpolation operator on the reference interval $[-1, 1]$ is given by

$$\mathcal{I} : C([-1,1]) \to \Pi_{m-1}, \qquad u \mapsto \sum_{\nu=1}^{m} u(\xi_\nu)\mathcal{L}_\nu \tag{3}$$

where Π_{m-1} is the space of $(m-1)$-th order polynomials and $(\xi_\nu)_{\nu=1}^m$ are the interpolation points with associated Lagrange polynomials $(\mathcal{L}_\nu)_{\nu=1}^m$.

If $[a, b]$ is an arbitrary closed interval, the transformed interpolation operator is given by $\mathcal{I}_{[a,b]}[u] := (\mathcal{I}[u\circ\Phi_{[a,b]}])\circ\Phi_{[a,b]}^{-1}$, where $\Phi_{[a,b]} : [-1, 1] \to [a, b],\ x \mapsto ((b-a)x+(b+a))/2$ is the affine mapping from the reference interval to $[a, b]$.

For an axially parallel box $B \subset \mathbb{R}^d$ with

$$B = [a_1, b_1] \times \cdots \times [a_d, b_d]$$

the multidimensional interpolation operator is given by

$$\mathcal{I}_B := \mathcal{I}_{[a_1,b_1]} \otimes \cdots \otimes \mathcal{I}_{[a_d,b_d]}\ . \tag{4}$$

To apply interpolation to the d-dimensional model problem, we need the kernel function to be smooth in the considered domain, i.e., B in this case. We ensure this with an admissibility condition. For $t \times s \subseteq I \times I$ we define the corresponding domains

$$\tau := \bigcup_{i\in r} \operatorname{supp}(\varphi_i), \quad \sigma := \bigcup_{i\in s} \operatorname{supp}(\varphi_i)$$

and (minimal) axially parallel boxes B_τ, B_σ containing τ and σ.

For the interpolation we use the standard admissibility condition

$$\min\{\operatorname{diam}(B_\tau), \operatorname{diam}(B_\sigma)\} \le \operatorname{dist}(B_\tau, B_\sigma) \tag{5}$$

and define the kernel approximation depending on the diameters of B_τ and B_σ

$$\tilde{g}(x,y) := \begin{cases} \mathcal{I}_{B_\tau}[g(\cdot,y)](x) & \text{if } \operatorname{diam}(B_\tau) \le \operatorname{diam}(B_\sigma) \\ \mathcal{I}_{B_\sigma}[g(x,\cdot)](y) & \text{otherwise} \end{cases} .$$

In the case $\operatorname{diam}(B_\tau) \le \operatorname{diam}(B_\sigma)$, we have the degenerated kernel approximation $\tilde{g}(x,y) = \mathcal{I}_{B_\tau}[g(\cdot,y)](x)$, i.e.,

$$\tilde{g}(x,y) = \sum_{\nu\in K} g(\xi_{\tau,\nu}, y)\mathcal{L}_{\tau,\nu}(x)$$

due to (3) with $K := \{1,\dots,m\}^d$ and

$$\xi_{\tau,\nu} := (\xi_{[a_1,b_1],\nu_1}, \dots, \xi_{[a_d,b_d],\nu_d}) \quad \text{and} \quad \mathcal{L}_{\tau,\nu} := \mathcal{L}_{[a_1,b_1],\nu_1} \otimes \cdots \otimes \mathcal{L}_{[a_d,b_d],\nu_d}$$

where

$$\xi_{[a,b],\nu} := \Phi_{[a,b]}(\xi_\nu) \quad \text{and} \quad \mathcal{L}_{[a,b],\nu}(x) := \prod_{\mu=0,\mu\neq\nu}^{m} \frac{x - \xi_{[a,b],\mu}}{\xi_{[a,b],\nu} - \xi_{[a,b],\mu}}$$

are the transformed interpolation points and Lagrange polynomials.

We define the entries of the matrix $\widetilde{G}$ by

$$\widetilde{G}_{ij} := \int_\tau \int_\sigma \varphi_i(x)\tilde{g}(x,y)\varphi_j(y)\,\mathrm{d}x\,\mathrm{d}y$$

and get the representation $\widetilde{G} = AB^\top$ with

$$A_{i\nu} = \int_\tau \varphi_i(x)\mathcal{L}_{\tau,\nu}(x)\,\mathrm{d}x \quad \text{and} \quad B_{j\nu} = \int_\sigma \varphi_j(y) g(\xi_{\tau,\nu}, y)\,\mathrm{d}y \tag{6}$$

for $i \in t$, $j \in s$ and $\nu \in K$, which implies $\operatorname{rank}\widetilde{G} \le \#K = m^d$. By the same arguments $\operatorname{rank}\widetilde{G} \le m^d$ holds for the second case $\operatorname{diam}(B_\tau) \ge \operatorname{diam}(B_\sigma)$, too.

In this paper we can get to $\operatorname{rank}\widetilde{G} = 8\,m$ for the 2-dimensional model problem by using the second Green formula and quadrature instead of $\operatorname{rank}\widetilde{G} = m^2$ with interpolation.

We are confident that this approach will lead to $\operatorname{rank}\widetilde{G} \le 4\,d\,m^{d-1}$ or even $\operatorname{rank}\widetilde{G} \le 2\,d\,m^{d-1}$ for the d-dimensional problem.

That means a reduction in rank, and therefore in complexity, of approximately a factor m compared to interpolation.

2 Approximation by Green formula

We will use the Green formula to get a representation of the two-dimensional kernel function that can easily be approximated by using quadrature and then leads to a low rank matrix approximation.

2.1 Green formula for the kernel function

The kernel function for the two-dimensional problem is given by

$$g(x,y) = -\frac{1}{2\pi} \log \|x-y\|$$

and we denote the normal derivative in the first or in the second variable by

$$\frac{\partial g}{\partial n_x}(x,y) = -\frac{\langle n(x), x-y\rangle}{2\pi\|x-y\|^2} \quad \text{and} \quad \frac{\partial g}{\partial n_y}(x,y) = -\frac{\langle n(y), y-x\rangle}{2\pi\|x-y\|^2}\,.$$

The following theorem converts the second Green identity as in [12] into an equation referred to as the *second Green formula* below.

Theorem 2.1 (Second Green formula) *Let $\Theta \subseteq \mathbb{R}^d$ be a normal domain, $\Gamma = \partial\Theta$ and let $n : \Gamma \to \mathbb{R}^d$ be the outer normal direction. If $u \in C^2(\overline{\Theta})$ satisfies the potential equation $\Delta u = 0$, we have*

$$u(x) = \int_\Gamma g(x,z)\frac{\partial u}{\partial n}(z)\,\mathrm{d}z - \int_\Gamma \frac{\partial g}{\partial n_z}(x,z)u(z)\,\mathrm{d}z \tag{7}$$

for all $x \in \Theta$.

Proof. [16, Theorem 2.2.2.] ■

Lemma 2.2 *For fixed $y \in \mathbb{R}^2$ the potential equation in $\mathbb{R}^2\backslash\{y\}$ is solved by $g(x,y)$.*

Proof. Let $x, y \in \mathbb{R}^2$ and $x \neq y$. Then we have

$$\begin{aligned}
\Delta_x g(x,y) &= -\frac{1}{4\pi}\Delta_x \log \|x-y\|^2 \\
&= -\frac{1}{4\pi}\left(\frac{\partial}{\partial x_1}\frac{2(x_1-y_1)}{\|x-y\|^2} + \frac{\partial}{\partial x_2}\frac{2(x_2-y_2)}{\|x-y\|^2}\right) \\
&= -\frac{1}{4\pi}\left(\frac{2}{\|x-y\|^2} - \frac{4(x_1-y_1)^2}{\|x-y\|^4} + \frac{2}{\|x-y\|^2} - \frac{4(x_2-y_2)^2}{\|x-y\|^4}\right) \\
&= -\frac{1}{2\pi\|x-y\|^2}\left(2 - 2\frac{\|x-y\|^2}{\|x-y\|^2}\right) \qquad = 0\,.
\end{aligned}$$

■

So we can apply the second Green formula (7) to $u(x) := g(x,y)$ and get

$$g(x,y) = \int_\Gamma g(x,z)\frac{\partial g}{\partial n_x}(z,y)\,\mathrm{d}z - \int_\Gamma \frac{\partial g}{\partial n_y}(x,z)g(z,y)\,\mathrm{d}z\,. \tag{8}$$

In this representation of the kernel function we have separated variables and therefore *almost* a degenerated kernel function as in (2) just with integrals instead of a sum. So far this representation is exact, but by approximating the path integrals over Γ by quadrature we obtain a practical algorithm.

2.2 Approximation by quadrature

To use the Green formula (7) for the kernel function we have to choose Θ as a normal domain around one of the bounding boxes, so the obvious choice would be a circle or a rectangle, e.g., with $B_\tau \subset \Theta$. In order to apply quadrature techniques to the kernel function represented with the Green formula (8), we need a parametrisation for the boundary Γ of Θ. But first we need an admissibility condition to ensure the smoothness of the kernel on the domains we are about to use our technique on.

For the two-dimensional problem the bounding boxes B_τ and B_σ can be written as $B_\tau = [a_1,b_1]\times[a_2,b_2]$ with $a_1 < b_1, a_2 < b_2$ and $B_\sigma = [c_1,d_1]\times[c_2,d_2]$ with $c_1 < d_1, c_2 < d_2$. We use the maximal diameter and distance of the bounding boxes in coordinate direction, i.e.,

$$\begin{aligned}\mathrm{diam}_{\max}(B_\tau) &:= \sup\{\|x-y\|_\infty \;:\; x,y\in B_\tau\}\\ &= \max\{|b_1-a_1|,|b_2-a_2|\}\end{aligned}$$

and

$$\begin{aligned}\mathrm{dist}_{\max}(B_\tau,B_\sigma) &:= \inf\{\|x-y\|_\infty \;:\; x\in B_\tau, y\in B_\sigma\}\\ &= \max\{\mathrm{dist}([a_1,b_1],[c_1,d_1]),\mathrm{dist}([a_2,b_2],[c_2,d_2])\}\,.\end{aligned}$$

to define the admissibility condition

$$\mathrm{diam}_{\max}(B_\tau) \;\le\; \mathrm{dist}_{\max}(B_\tau,B_\sigma)\,. \tag{9}$$

For the construction of a parametrisation we define

$$\delta \;:=\; \frac{1}{2}\,\mathrm{diam}_{\max}(B_\tau)\,. \tag{10}$$

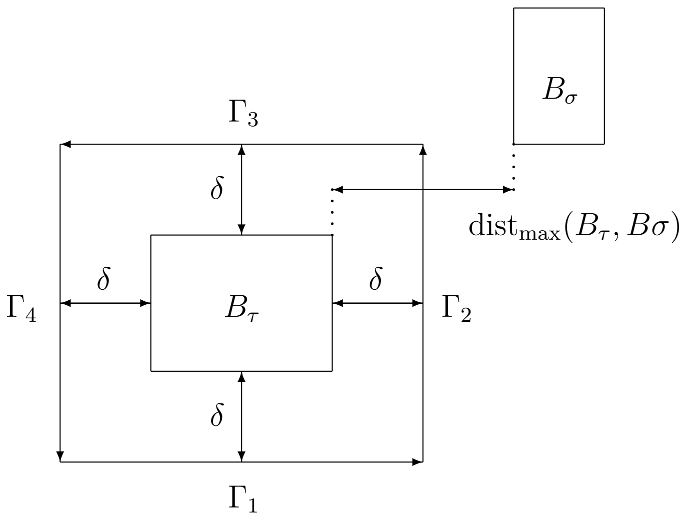
In the following let $\Gamma = \bigcup_{\iota=1}^4 \Gamma_\iota$ be a rectangle with the four-part parametrisation

$$\gamma_1 : [-1,1] \to \Gamma_1 \subseteq \mathbb{R}^2, \quad \gamma_1(t) = \begin{pmatrix} \frac{b_1+a_1}{2} + \frac{b_1-a_1+2\delta}{2}\,t \\ a_2-\delta \end{pmatrix},$$

$$\gamma_2 : [-1,1] \to \Gamma_2 \subseteq \mathbb{R}^2, \quad \gamma_2(t) = \begin{pmatrix} b_1+\delta \\ \frac{b_2+a_2}{2} + \frac{b_2-a_2+2\delta}{2}\,t \end{pmatrix},$$

$$\gamma_3 : [-1,1] \to \Gamma_3 \subseteq \mathbb{R}^2, \quad \gamma_3(t) = \begin{pmatrix} \frac{b_1+a_1}{2} - \frac{b_1-a_1+2\delta}{2}\, t \\ b_2 + \delta \end{pmatrix},$$

$$\gamma_4 : [-1,1] \to \Gamma_4 \subseteq \mathbb{R}^2, \quad \gamma_4(t) = \begin{pmatrix} a_1 - \delta \\ \frac{b_2+a_2}{2} - \frac{b_2-a_2+2\delta}{2}\, t \end{pmatrix}.$$



For error estimates we need the following two lemmas which are consequences of the construction of Γ with δ-distance around the bounding box B_τ.

Lemma 2.3 *For all $\iota \in \{1,2,3,4\}$ holds*

$$\|\gamma'_\iota(t)\| \;\leq\; \mathrm{diam}_{\max}(B_\tau)\,. \tag{11}$$

Proof. For $\iota = 1$ we have

$$\|\gamma'_\iota(t)\| = \frac{b_1 - a_1 + 2\delta}{2} = \frac{b_1 - a_1 + \mathrm{diam}_{\max}(B_\tau)}{2} \leq \frac{2\,\mathrm{diam}_{\max}(B_\tau)}{2}\,.$$

For $\iota = 2,3,4$ the proof is similar. ■

Lemma 2.4 *Let the admissibility condition* (9) *hold. Then we have*

$$\delta \;\leq\; \min\{\mathrm{dist}(x,\Gamma), \mathrm{dist}(\Gamma,y) \;:\; x \in B_\tau, y \in B_\sigma\}\,. \tag{12}$$

Proof. Let $y \in B_\sigma$ and choose $z \in \Gamma$ with $\|y-z\|_\infty$ minimal. Then we choose $x \in B_\tau$ with $\|x-z\|_\infty$ minimal. By construction this implies $\|x-z\|_\infty = \delta$. Since the admissibility condition (9) holds, we have

$$\begin{aligned} \|y-z\|_2 \;&\geq\; \|y-z\|_\infty \;\geq\; \|y-x\|_\infty - \|x-z\|_\infty \;\geq\; \mathrm{dist}_{\max}(B_\tau, B_\sigma) - \delta \\ &=\; \mathrm{dist}_{\max}(B_\tau, B_\sigma) - \frac{1}{2}\mathrm{diam}_{\max}(B_\tau) \;\geq\; \frac{1}{2}\mathrm{diam}_{\max}(B_\tau) \;=\; \delta \end{aligned}$$

and therefore the proof is complete. ■

We use composite quadrature, e.g., Gaussian quadrature, with ℓ subsections with m quadrature points each. Let $t_1, \dots, t_{m\ell} \in [-1,1]$ and $w_1, \dots, w_{m\ell} \in \mathbb{R}$ be the whole of the quadrature points and weights, then we have with (8)

$$\begin{aligned} g(x,y) \approx \tilde{g}(x,y) = \sum_{\iota=1}^{4} \sum_{\nu=1}^{m\ell} (\|\gamma_\iota'(t_\nu)\| w_\nu \, g(x, \gamma_\iota(t_\nu)) \, \frac{\partial g}{\partial n_x}(\gamma_\iota(t_\nu), y) - \\ \|\gamma_\iota'(t_\nu)\| w_\nu \, \frac{\partial g}{\partial n_y}(x, \gamma_\iota(t_\nu)) \, g(\gamma_\iota(t_\nu), y) \; . \end{aligned} \tag{13}$$

With this degenerate representation of the kernel, we can get a low rank approximation for the corresponding matrix block of the admissible index sub-blocks $t \times s \subseteq I \times I$.

2.3 Low rank approximation of the matrix block

We seek a low rank matrix $\widetilde{G}|_{t\times s}$ for the corresponding sub-matrix $G|_{t\times s}$. We can achieve this by defining the entries of the matrix $\widetilde{G}|_{t\times s}$ by

$$\widetilde{G}_{ij} = \int_\Omega \int_\Omega \varphi_i(x) \tilde{g}(x,y) \varphi_j(y) \, \mathrm{d}x \, \mathrm{d}y \qquad \text{for } \; i \in t, \; j \in s.$$

Inserting the quadrature approximation $\tilde{g}(x,y)$ from (13) we have the representation

$$\widetilde{G}|_{t\times s} = \sum_{\iota=1}^{4} A_\iota B_\iota^\top - \widehat{A}_\iota \widehat{B}_\iota^\top$$

with

$$\begin{aligned} (A_\iota)_{i\nu} &= \|\gamma_\iota'(t_\nu)\| w_\nu \int_\Omega g(x, \gamma_\iota(t_\nu)) \varphi_i(x) \, \mathrm{d}x \\ (\widehat{A}_\iota)_{i\nu} &= \|\gamma_\iota'(t_\nu)\| w_\nu \int_\Omega \frac{\partial g}{\partial n_y}(x, \gamma_\iota(t_\nu)) \varphi_i(x) \, \mathrm{d}x \\ (B_\iota)_{j\nu} &= \int_\Omega \frac{\partial g}{\partial n_x}(\gamma_\iota(t_\nu), y) \varphi_j(y) \, \mathrm{d}y \\ (\widehat{B}_\iota)_{j\nu} &= \int_\Omega g(\gamma_\iota(t_\nu), y) \varphi_j(y) \, \mathrm{d}y \end{aligned} \tag{14}$$

for $i \in t$, $j \in s$ and $\nu \in \{1, \dots, m\ell\}$, which implies $\operatorname{rank} \widetilde{G}|_{t\times s} \leq 8m\ell$, i.e., $\operatorname{rank} m\ell$ for each $A_\iota B_\iota^\top$ and each $\widehat{A}_\iota \widehat{B}_\iota^\top$. The integrals (14) are the same that appear in collocation methods, but here the kernel is integrated only in a smooth region, i.e., standard quadrature formulae apply.

3 Error analysis

Before starting the error analysis, we need to take a look at one particular property of our kernel function.

Definition 3.1 (Asymptotically smooth kernel function) *A kernel function $g : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is called* asymptotically smooth*, if there exist constants $C_{\mathrm{as}} > 0$ and $c_0 \geq 1$ and a singularity degree $\sigma \in \mathbb{N}$ such that for all $n \in \mathbb{N}_0$, $x, y \in \mathbb{R}^d$ with $x \neq y$ and all directions $p \in \mathbb{R}^d \times \mathbb{R}^d$ the inequality*

$$|\partial_p^n g(x,y)| \leq C_{\mathrm{as}} \frac{(\sigma - 1 + n)! c_0^n \|p\|^n}{\|x-y\|^{\sigma+n}} \tag{15}$$

holds.

The function is also called asymptotically smooth*, if* (15) *holds for $\sigma = 0$ and all $n \in \mathbb{N}$, $x, y \in \mathbb{R}^d$ with $x \neq y$ and all directions $p \in \mathbb{R}^d \times \mathbb{R}^d$. $\sigma = 0$ corresponds to logarithmic singularities.*

We observe that our kernel function $g(x,y) = -\frac{1}{2\pi} \log \|x-y\|$ is asymptotically smooth with $\sigma = 0$ according to Definition 3.1 [18, Korollar E.1.2].

The approximation of our method takes place in the quadrature, therefore the error is closely related to the approximation error of the composite quadrature.

We recall that a quadrature rule for the integral

$$\mathcal{I} \colon C[-1,1] \to \mathbb{R}, \quad f \mapsto \int_{-1}^{1} f(x) \, \mathrm{d}x$$

is given by

$$\mathcal{Q} \colon C[-1,1] \to \mathbb{R}, \quad f \mapsto \sum_{\nu=1}^{m\ell} w_\nu f(x_\nu)$$

and the error estimate for composite quadrature of n-th degree with ℓ subsections reads

$$|\mathcal{I}(f) - \mathcal{Q}(f)| \ \leq \ 2(2 + C_{\mathrm{Q}}) \left(\frac{1}{2\ell}\right)^{n+1} \frac{\|f^{(n+1)}\|_\infty}{(n+1)!} \tag{16}$$

where $C_{\mathrm{Q}} := \sum_{\nu=1}^{m} |w_\nu|$ is the stability constant. In the case of Gauss quadrature we have $C_{\mathrm{Q}} = 2$ and $n = 2m - 1$.

In the following theorem we examine the case $\sigma > 0$ although the proof is very similar for the case $\sigma = 0$. We assume that $g(x,y)$ is asymptotically smooth with singularity degree σ and $\nabla_x g(x,y)$ and $\nabla_y g(x,y)$ are asymptotically smooth in every component with $\sigma + 1$.

Theorem 3.2 (Quadrature error) *Let $g(x,y)$ be asymptotically smooth with singularity degree $\sigma \neq 0$ and $\nabla_x g(x,y)$ and $\nabla_y g(x,y)$ be asymptotically smooth with $\sigma + 1$.*

For the approximation error for composite quadrature of n-th degree with ℓ subsections with m quadrature points each as in formula (13) *we have*

$$|g(x,y) - \tilde{g}(x,y)| \;\leq\; 8\, C\, p(n+1) \left(\frac{c_0}{\ell}\right)^{n+1}$$

for all $x \in B_\tau$ and $y \in B_\sigma$ with $c_0 \geq 1$, $C > 0$, and a polynomial p.

Proof. Let $\iota \in \{1, \ldots, 4\}$ be fixed. We denote the first inner integral term of (8) as f, more precisely

$$f(t) \;:=\; g(x, \gamma_\iota(t)) \frac{\partial g}{\partial n_x}(\gamma_\iota(t), y)\,. \tag{17}$$

To determine the error $|g(x,y) - \tilde{g}(x,y)|$ we use the well known error estimate (16) for composite quadrature formulae $\mathcal{Q}$ of n-th degree

$$\mathcal{Q}(f) \;=\; \sum_{\nu=1}^{m\ell} w_\nu\, g(x, \gamma_\iota(t_\nu))\, \frac{\partial g}{\partial n_x}(\gamma_\iota(t_\nu), y)$$

on the function f with the exact integral

$$\mathcal{I}(f) \;=\; \int_{-1}^{1} g(x, \gamma_\iota(t))\, \frac{\partial g}{\partial n_x}(\gamma_\iota(t), y)\; \mathrm{d}t\,.$$

In order to use the Leibniz formula to calculate $f^{(n+1)}$ we take a look at each of the two factors in f. By defining

$$h(z,y) := \frac{\partial g}{\partial n_x}(z,y)$$

we get with $n_\iota := n(\gamma_\iota(t))$, $|n_\iota| = 1$ and the Cauchy-Schwarz-inequality

$$\begin{aligned} \left|\partial^n_{\gamma'_\iota(t)} h(\gamma_\iota(t), y)\right| &= \left|\partial^n_{\gamma'_\iota(t)} \langle n_\iota, \nabla_x\, g(\gamma_\iota(t), y)\rangle\right| \\ &= \left|\langle n_\iota, \partial^n_{\gamma'_\iota(t)} \nabla_x\, g(\gamma_\iota(t), y)\rangle\right| \\ &\leq \left|\partial^n_{\gamma'_\iota(t)} \nabla_x\, g(\gamma_\iota(t), y)\right| \end{aligned}$$

and therefore a function to which Definition 3.1 applies with singularity degree $\sigma + 1$ and a constant $C_{\mathrm{as},h}$ as we assumed $\nabla_x\, g(x,y)$ to be asymptotically smooth in every component with $\sigma + 1$.

Since $\frac{d^2}{dt^2}\gamma_\iota(t) \equiv 0$ for all $\iota \in \{1,2,3,4\}$ the n-th derivative of $g(x, \gamma_\iota(t))$ equals the n-th directional derivative in $\gamma'_\iota(t)$ direction:

$$\frac{d^n}{dt^n}\, g(x, \gamma_\iota(t)) = \partial^n_{\gamma'_\iota(t)}\, g(x, \gamma_\iota(t))\,. \tag{18}$$

This also holds for the n-th derivative of $h(\gamma_\iota(t), y)$:

$$\frac{d^n}{dt^n}\, h(\gamma_\iota(t), y) = \partial^n_{\gamma'_\iota(t)}\, h(\gamma_\iota(t), y)\,. \tag{19}$$

With (18) and (19) we can use the Leibniz formula on $|f^{(n+1)}|$ as follows

$$|f^{(n+1)}(t)| = \left| \frac{d^{n+1}}{dt^{n+1}} \left(g(x,\gamma_\iota(t)) \frac{\partial g}{\partial n_x}(\gamma_\iota(t), y) \right) \right|$$
$$\leq \sum_{k=0}^{n+1} \binom{n+1}{k} \left|\partial^k_{\gamma'_\iota(t)}\, g(x,\gamma_\iota(t))\right| \left|\partial^{n+1-k}_{\gamma'_\iota(t)}\, h(\gamma_\iota(t), y)\right| . \tag{20}$$

Now we can use the asymptotical smoothness (15) of both the kernel and the function h and get

$$|f^{(n+1)}(t)| \;\leq\; \sum_{k=0}^{n+1} \binom{n+1}{k} C_{\mathrm{as},g}\, \frac{(\sigma-1+k)!\, c_{0,g}^k \|\gamma'_\iota(t)\|^k}{\|x-\gamma_\iota(t)\|^{\sigma+k}}$$
$$C_{\mathrm{as},h}\, \frac{(\sigma+1+n-k)!\, c_{0,h}^{n+1-k} \|\gamma'_\iota(t)\|^{n+1-k}}{\|\gamma_\iota(t)-y\|^{\sigma+n+2-k}} .$$

Since $\|x-\gamma_\iota(t)\| \geq \mathrm{dist}(x,\Gamma)$ and $\|\gamma_\iota(t)-y\| \geq \mathrm{dist}(\Gamma, y)$ holds for all ι and with

$$\delta \leq \min\{\mathrm{dist}(x,\Gamma), \mathrm{dist}(\Gamma,y)\}$$

according to Lemma 2.4, and by defining $C_{\mathrm{as}} := C_{\mathrm{as},g} C_{\mathrm{as},h}$ and $c_0 := \max\{c_{0,g}, c_{0,h}\}$ we obtain

$$|f^{(n+1)}(t)| \;\leq\; C_{\mathrm{as}} \sum_{k=0}^{n+1} \binom{n+1}{k} \frac{(\sigma-1+k)!(\sigma+1+n-k)!\, c_0^{n+1} \|\gamma'_\iota(t)\|^{n+1}}{\delta^{\,2\sigma+n+2}}$$
$$\leq\; \frac{C_{\mathrm{as}}}{\delta^{\,2\sigma+1}} \left(\frac{c_0 \|\gamma'_\iota(t)\|}{\delta} \right)^{n+1} (n+1)! \sum_{k=0}^{n+1} \frac{(\sigma-1+k)!(\sigma+1+n-k)!}{(n+1-k)!k!}$$
$$=\; \frac{C_{\mathrm{as}}}{\delta^{\,2\sigma+1}} \left(\frac{c_0 \|\gamma'_\iota(t)\|}{\delta} \right)^{n+1} (n+1)! \;\; p(n+1)$$

where p is a polynomial of degree 2σ with

$$p(n+1) \;:=\; \sum_{k=0}^{n+1} \prod_{r=1}^{\sigma-1} (k+r) \prod_{s=1}^{\sigma} (n+1-k+s) .$$

With equation (11) we get

$$|f^{(n+1)}(t)| \;\leq\; \frac{C_{\mathrm{as}}}{\delta^{\,2\sigma+1}} \left(\frac{c_0\, \mathrm{diam}_{\max}(B_\tau)}{\delta} \right)^{n+1} (n+1)! \;\; p(n+1)$$
$$=\; \frac{C_{\mathrm{as}}}{\delta^{\,2\sigma+1}} (2c_0)^{n+1} \;\; (n+1)! \;\; p(n+1)$$

and inserted into the quadrature error estimate (16) we have

$$|\mathcal{I}(f) - \mathcal{Q}(f)| \;\leq\; 2(2+C_{\mathrm{Q}}) \left(\frac{1}{2\ell} \right)^{n+1} \frac{C_{\mathrm{as}}}{\delta^{\,2\sigma+1}} (2c_0)^{n+1} \;\; p(n+1)$$

$$= C \left(\frac{c_0}{\ell}\right)^{n+1} p(n+1) \tag{21}$$

where $C := \frac{2(2+C_{\mathrm{Q}})C_{\mathrm{as}}}{\delta^{2\sigma+1}}$.

We get a similar result if we choose the second inner integral term of (8) as f in (17) so that we have twice the error (21) for each Γ_ι and therefore for all four Γ_ι it follows

$$|g(x,y) - \tilde{g}(x,y)| \;\leq\; 4 \cdot 2 \cdot C\, p(n+1) \left(\frac{c_0}{\ell}\right)^{n+1} .$$

■

Theorem 3.2 states that the choice of ℓ should be $\ell > c_0$ to get exponential decay to counteract the polynomial growth of p with a larger n.

Remark 3.3 (Logarithmic singularities) *Since our kernel function in two dimensions $g(x,y) = -\frac{1}{2\pi}\log\|x-y\|$ is asymptotically smooth with $\sigma = 0$, the proof is slightly different in that case: we have to treat the first summand of* (20) *separately and get a logarithmic term. The final result, however, is not changed significantly except for the constants and we get*

$$|g(x,y) - \tilde{g}(x,y)| \;\leq\; 8\, C\, (\,|\log\delta| + \log(n+1)\,) \left(\frac{c_0}{\ell}\right)^{n+1} .$$

4 Numerical experiments

The question is how our approach compares in numerical tests to other analytical techniques for the approximation of the kernel function like the interpolation method as described in [6]. With the second order tensor product interpolation operator (4) there are m interpolation points for each direction, that is a local rank of $k = m^2$.

Using the Green formula and quadrature we save one direction because we integrate over the boundary Γ instead of the whole domain Θ. Hence we get a local rank of $k = 8m\ell$ with the four part parametrisation we described in section 2.2.

Another way to get a parametrisation to use with the Green formula (8) is to lay a circle around one of the bounding boxes. Here we are not using Gauss quadrature but spreading the quadrature points equally distanced across the circle. This is practicable only in 2D, in higher dimensions this construction will be more complex, because we are confronted with the task of arranging quadrature points on the sphere as "equally distanced" as possible. In case of a circle we have just one parametrisation instead of four and get a local rank of $k = 2m\ell$. This advantage in rank shows in the right picture of figure 1, where the left picture shows the error in relation to the quadrature order.

To control the quality of the approximation, the number of subintervals ℓ takes the role of the parameter η which occurs in the admissibility condition for the interpolation, and we can see that larger values of ℓ lead to faster convergence.

But before we go deeper into comparing our algorithm to interpolation, we will look at the behaviour of the error and the computing time for various degrees of freedom,

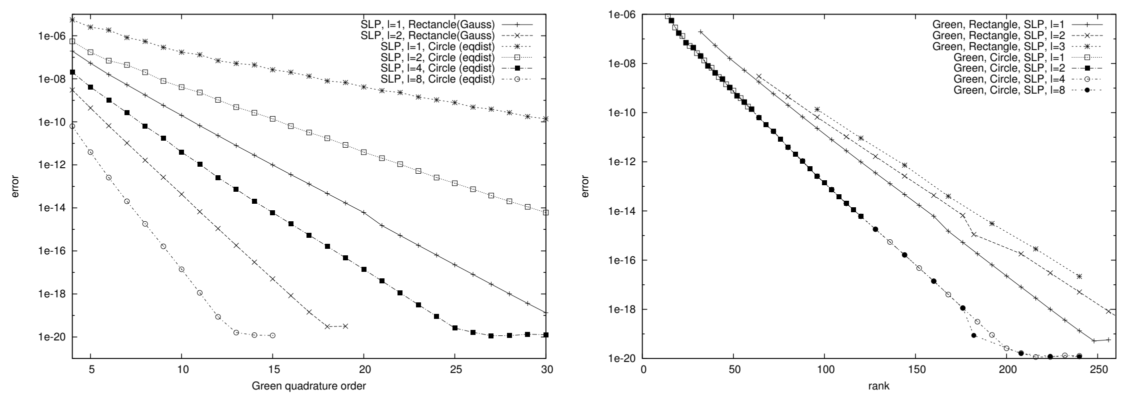


Figure 1: Error for $n = 32768$: different quadrature orders (left) and ranks (right)

quadrature orders and different numbers of subintervals for the composite quadrature for the rectangle parametrisation. We will concentrate on this parametrisation, because we have a rigorous error estimate in that case.

The underlying domain Ω is a circle in these *prove of concept* experiments.

n	$m = 4$	$m = 5$	$m = 6$	$m = 7$	$m = 8$	$m = 9$	$m = 10$	factor
1024	5.50_{-06}	1.64_{-06}	5.14_{-07}	1.65_{-07}	–	–	–	3.22
2048	3.12_{-06}	8.55_{-07}	2.38_{-07}	7.59_{-08}	2.60_{-08}	8.52_{-09}	2.83_{-09}	3.16
4096	1.55_{-06}	4.26_{-07}	1.18_{-07}	3.93_{-08}	1.31_{-08}	4.34_{-09}	1.40_{-09}	3.17
8192	7.68_{-07}	2.14_{-07}	5.91_{-08}	2.00_{-08}	6.70_{-09}	2.15_{-09}	7.18_{-10}	3.16
16384	4.50_{-07}	1.29_{-07}	3.80_{-08}	1.27_{-08}	4.27_{-09}	1.32_{-09}	4.59_{-10}	3.12
32768	1.94_{-07}	5.37_{-08}	1.59_{-08}	5.29_{-09}	1.76_{-09}	5.81_{-10}	2.00_{-10}	3.11
65536	9.87_{-08}	2.72_{-08}	7.80_{-09}	2.70_{-09}	9.14_{-10}	3.02_{-10}	1.04_{-10}	3.09

Table 1: Matrix approximation error: $\|G - \widetilde{G}\|_2$ for $\ell = 1$ (SLP)

Table 1 shows the matrix approximation errors $\|G - \widetilde{G}\|_2$ for different choices of the quadrature order m with $\ell = 1$, and different degrees of freedom n. The error behaves like $5 \cdot 10^{-5} \cdot q^m$ with $q \approx e^{-1.2} \approx 0.3$.

The same setting only with two subintervals, i.e. $\ell = 2$, leads to similar errors as shown in table 2. Here the error behaves like $10^{-5} \cdot q^m$ with $q \approx e^{-1.8} \approx 0.17$ and for $\ell = 3$ table 3 shows a behaviour like $5 \cdot 10^{-6} \cdot q^m$ with $q \approx e^{-2.5} \approx 0.08$.

The total time needed to build an $\mathcal{H}$-matrix can be seen in table 4 for $\ell = 1$ and in table 5 for $\ell = 3$. We observe a behaviour like $\mathcal{O}(nk \log n)$ as can also be seen in figure 3.

We can use our technique not only for the single layer potential (SLP), but also for the double layer potential (DLP). Instead of computing the matrix G_{ij} like in (1) for the SLP, we build the matrix with

$$G_{ij} := \int_\Omega \int_\Omega \varphi_i(x) \frac{\partial g}{\partial n_y}(x, y)\, \varphi_j(y)\, \mathrm{d}x\, \mathrm{d}y$$

n	$m=4$	$m=5$	$m=6$	$m=7$	$m=8$	$m=9$	$m=10$	factor
1024	–	–	–	–	–	–	–	–
2048	4.12_{-08}	6.28_{-09}	9.50_{-10}	1.44_{-10}	–	–	–	6.60
4096	2.16_{-08}	3.19_{-09}	4.73_{-10}	7.43_{-11}	1.18_{-11}	1.87_{-12}	3.08_{-13}	6.36
8192	1.11_{-08}	1.64_{-09}	2.41_{-10}	3.81_{-11}	6.07_{-12}	9.73_{-13}	1.56_{-13}	6.37
16384	6.87_{-09}	1.02_{-09}	1.51_{-10}	2.36_{-11}	3.75_{-12}	6.01_{-13}	9.76_{-14}	6.37
32768	3.01_{-09}	4.40_{-10}	6.55_{-11}	1.04_{-11}	1.65_{-12}	2.63_{-13}	4.32_{-14}	6.38
65536	1.49_{-09}	2.20_{-10}	3.31_{-11}	5.30_{-12}	8.44_{-13}	1.36_{-13}	2.22_{-14}	6.33

Table 2: Matrix approximation error: $\|G - \widetilde{G}\|_2$ for $\ell = 2$ (SLP)

for the DLP. Since $\frac{\partial g}{\partial n_y}(x, y)$ also solves the potential equation, we can apply the Green formula as above and get

$$\frac{\partial g}{\partial n_y}(x,y) = \int_\Gamma g(x,z)\frac{\partial^2 g}{\partial n_z n_y}(z,y)\,\mathrm{d}z - \int_\Gamma \frac{\partial g}{\partial n_z}(x,z)\frac{\partial g}{\partial n_y}(z,y)\,\mathrm{d}z\,. \tag{22}$$

For (22) we can use the same parametrisation and quadrature as we used for the SLP in Section 2.2.

Table 6 shows the behaviour of the error $\|G - \widetilde{G}\|_2$ for the DLP and $\ell = 1$. We can see that the error is slightly larger than it was under the same circumstances for the SLP but decays for growing m and n in the same way as before.

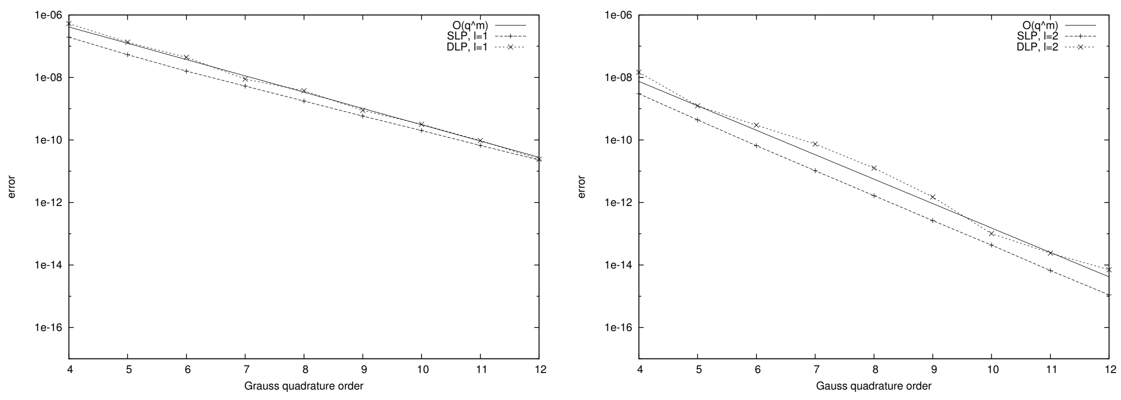


Figure 2: Error for SLP and DLP: $n = 32768$, $\ell = 1$ (left), $\ell = 2$ (right)

The errors for SLP and DLP for different quadrature orders are presented in figure 2. The left picture shows the errors for just one interval $\ell = 1$ and the right picture shows the errors for two subintervals $\ell = 2$. In both pictures it is visible that the error behaves like $\mathcal{O}(q^m)$ as seen before in tables 1, 2 and 6.

The total time needed to build a $\mathcal{H}$-matrix for the double layer potential and $\ell = 1$ can be seen in table 7. It does not differ much from the total time for the SLP build.

Figure 3 shows the relative time needed for the SLP build with $\ell = 1$ and $\ell = 3$ (in the first picture) compared to DLP build with $\ell = 1$ and $\ell = 3$ (in the second picture) with

n	$m=4$	$m=5$	$m=6$	$m=7$	$m=8$	$m=9$	$m=10$	factor
1024	–	–	–	–	–	–	–	–
2048	2.03_{-09}	1.26_{-10}	–	–	–	–	–	16.0
4096	9.77_{-10}	6.20_{-11}	4.90_{-12}	2.88_{-13}	2.27_{-14}	2.04_{-15}	1.54_{-16}	13.8
8192	4.97_{-10}	3.32_{-11}	2.60_{-12}	1.49_{-13}	1.15_{-14}	1.02_{-15}	7.84_{-17}	13.5
16384	3.03_{-10}	2.04_{-11}	1.58_{-12}	8.62_{-14}	7.31_{-15}	6.62_{-16}	4.97_{-17}	13.5
32768	1.36_{-10}	9.19_{-12}	7.30_{-13}	3.98_{-14}	3.08_{-15}	2.87_{-16}	2.18_{-17}	13.5
65536	6.52_{-11}	4.74_{-12}	3.82_{-13}	2.34_{-14}	1.95_{-15}	1.75_{-16}	1.42_{-17}	12.9

Table 3: Matrix approximation error: $\|G - \widetilde{G}\|_2$ for $\ell = 3$ (SLP)

n	$m=4$	$m=5$	$m=6$	$m=7$	$m=8$	$m=9$	$m=10$
1024	0.2	0.2	0.2	0.2	0.3	0.3	0.3
2048	0.5	0.5	0.6	0.6	0.7	0.7	0.7
4096	1.1	1.3	1.7	1.7	1.8	1.9	2.0
8192	2.4	2.8	3.6	3.9	4.2	4.6	5.1
16384	5.4	6.3	7.9	8.6	9.4	10.2	11.3
32768	12.3	14.2	17.2	19.1	20.9	23.1	25.4
65536	26.9	31.2	37.5	42.0	46.3	51.2	55.8

Table 4: Building the $\mathcal{H}$-matrix in seconds for $\ell = 1$ (SLP)

n	$m=4$	$m=5$	$m=6$	$m=7$	$m=8$	$m=9$	$m=10$
1024	0.3	0.3	0.3	0.3	0.3	0.3	0.3
2048	0.9	0.9	1.2	1.2	1.2	1.2	1.2
4096	2.5	2.6	2.7	2.9	3.6	3.7	3.8
8192	7.1	7.3	7.7	8.3	10.3	10.6	10.9
16384	15.2	16.2	18.1	22.5	26.7	27.6	28.5
32768	33.0	36.0	41.1	53.2	60.6	63.9	66.8
65536	74.1	81.1	93.3	111.7	127.1	137.8	146.7

Table 5: Building the $\mathcal{H}$-matrix in seconds for $\ell = 3$ (SLP)

n	$m=4$	$m=5$	$m=6$	$m=7$	$m=8$	$m=9$	$m=10$	factor
1024	8.77_{-06}	2.62_{-06}	5.98_{-07}	1.91_{-07}	–	–	–	3.62
2048	5.79_{-06}	1.93_{-06}	4.12_{-07}	1.12_{-07}	3.93_{-08}	1.02_{-08}	3.61_{-09}	3.50
4096	2.95_{-06}	1.00_{-06}	3.47_{-07}	6.88_{-08}	2.95_{-08}	6.92_{-09}	2.46_{-09}	3.53
8192	1.59_{-06}	5.06_{-07}	1.77_{-07}	3.43_{-08}	1.49_{-08}	3.50_{-09}	1.24_{-09}	3.48
16384	9.66_{-07}	3.04_{-07}	9.45_{-08}	1.64_{-08}	1.01_{-08}	3.84_{-09}	9.84_{-10}	3.29
32768	5.26_{-07}	1.35_{-07}	4.42_{-08}	8.90_{-09}	3.75_{-09}	8.86_{-10}	3.17_{-10}	3.57
65536	3.34_{-07}	6.80_{-08}	2.23_{-08}	4.39_{-09}	1.87_{-09}	4.44_{-10}	1.58_{-10}	3.72

Table 6: Matrix approximation error: $\|G - \widetilde{G}\|_2$ for $\ell = 1$ (DLP)

n	$m=4$	$m=5$	$m=6$	$m=7$	$m=8$	$m=9$	$m=10$
1024	0.1	0.1	0.1	0.1	0.1	0.1	0.1
2048	0.2	0.2	0.2	0.3	0.3	0.3	0.3
4096	0.6	0.6	0.7	0.7	0.7	0.8	0.9
8192	1.4	1.5	1.7	1.9	2.0	2.2	2.4
16384	3.2	3.6	4.1	4.6	5.1	5.5	6.0
32768	7.5	8.3	9.5	10.7	11.9	13.1	14.2
65536	17.1	19.0	21.1	24.8	27.6	30.5	33.2

Table 7: Building the $\mathcal{H}$-matrix in seconds for $\ell = 1$ (DLP)

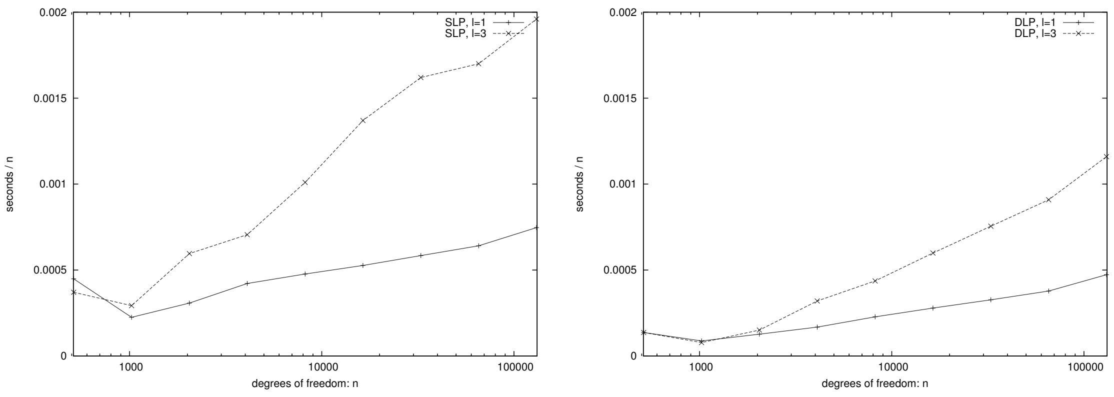


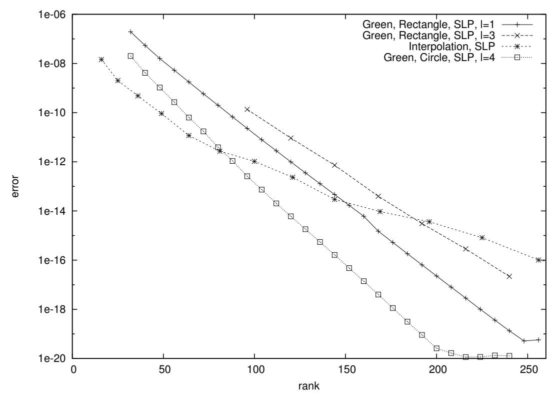
Figure 3: Building the $\mathcal{H}$-matrix in seconds/n

quadrature order $m = 7$ on a logarithmically scaled x-axis for the degrees of freedom. The relative time for SLP is slightly higher than for DLP but the characteristics of the curve are very similar, just the choice of the number of subintervals ℓ has greater influence on the relative time: the higher ℓ, the more time is needed. Altogether we observe logarithmic growth in time like $\sim \log_2 n$ as predicted by standard theory.

Finally the question is how our approach compares in numerical tests to typical techniques for the approximation of the kernel function like the interpolation method as described in [6].

Figure 4 shows the decaying error for higher local ranks for SLP with $\ell = 1$ and $\ell = 3$ and DLP with $\ell = 1$ and $\ell = 3$ compared to interpolation. For $\ell = 1$ we observe in both cases, SLP and DLP, that our approach is more accurate at higher ranks than interpolation.

So the basic algorithm is competitive as it is, but we have several options to enhance the algorithm. We expect to be able to improve our method by exploiting the relation between Dirichlet and Neumann values to reduce the rank by a factor of two. In this case our method would come close to optimal efficiency. Above that we can construct a very efficient hybrid method by using the quadrature approach with Green's formula combined with cross approximation and interpolation.



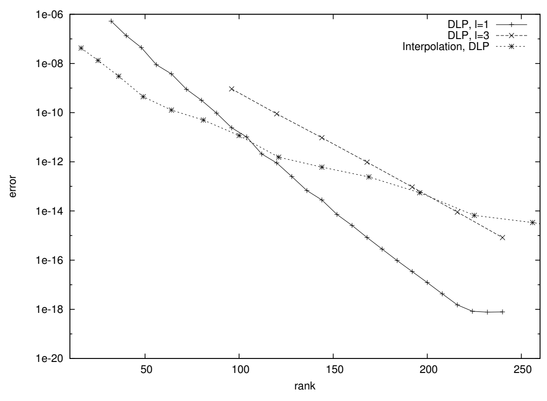


Figure 4: Error $\|G - \widetilde{G}\|_2$ for interpolation compared to our approach

5 Conclusion

Our algorithm can compete with other analytical methods like interpolation in matters of error and timing. Especially when interested in larger problems and higher accuracies our method has the advantage over interpolation.

We introduced the method only in the two-dimensional case, to keep the error analysis simple. The algorithm, as it is presented here, is functional also for the three-dimensional case. In three dimensions the cuboid parametrisation has six parts and we get a rather high local rank of $12m^2\ell^2$. Nevertheless we can make some adjustments (at the expense of the simplicity of the algorithm) and get an efficient algorithm for the three-dimensional case.

The development of an efficient hybrid method that will be competitive also in 3D and an approximation by $\mathcal{H}^2$-matrices [20, 5] is the subject of future work.

References

[1] C. R. Anderson. An implementation of the fast multipole method without multipoles. *SIAM J. Sci. Stat. Comp.*, 13:923–947, 1992.

[2] M. Bebendorf. Approximation of boundary element matrices. *Numer. Math.*, 86(4):565–589, 2000.

[3] M. Bebendorf and R. Grzhibovskis. Accelerating Galerkin BEM for linear elasticity using adaptive cross approximation. *Math. Meth. Appl. Sci.*, 29:1721–1747, 2006.

[4] M. Bebendorf and S. Rjasanow. Adaptive Low-Rank Approximation of Collocation Matrices. *Computing*, 70(1):1–24, 2003.

[5] S. Börm. *Efficient Numerical Methods for Non-local Operators: $\mathcal{H}^2$-Matrix Compression, Algorithms and Analysis*, volume 14 of *EMS Tracts in Mathematics*. EMS, 2010.

[6] S. Börm and L. Grasedyck. Low-rank approximation of integral operators by interpolation. *Computing*, 72:325–332, 2004.

[7] S. Börm, L. Grasedyck, and W. Hackbusch. Introduction to hierarchical matrices with applications. *Eng. Anal. Bound. Elem.*, 27:405–422, 2003.

[8] H. Cheng, Z. Gimbutas, P.-G. Martinsson, and V. Rokhlin. On the compression of low rank matrices. *SIAM J. Sci. Comput.*, 26(4):1389–1404, 2005.

[9] W. Dahmen and R. Schneider. Wavelets on manifolds I: Construction and domain decomposition. *SIAM J. Math. Anal.*, 31:184–230, 1999.

[10] S. A. Goreinov, E. E. Tyrtyshnikov, and N. L. Zamarashkin. A theory of pseudoskeleton approximations. *Lin. Alg. Appl.*, 261:1–22, 1997.

[11] S. A. Goreinov, N. L. Zamarashkin, and E. E. Tyrtyshnikov. Pseudo-skeleton approximations by matrices of maximal volume. *Mathmatical Notes*, 62:515–519, 1997.

[12] G. Green. An essay on the application of mathematical analysis to the theories of electricity and magnetism. Nottingham, 1828.

[13] L. Greengard, D. Gueyffier, P.-G. Martinsson, and V. Rokhlin. Fast direct solvers for integral equations in complex three-dimensional domains. *Acta Numerica*, 18:243–275, 2009.

[14] L. Greengard and V. Rokhlin. A fast algorithm for particle simulations. *J. Comp. Phys.*, 73:325–348, 1987.

[15] L. Greengard and V. Rokhlin. A new version of the fast multipole method for the Laplace in three dimensions. In *Acta Numerica 1997*, pages 229–269. Cambridge University Press, 1997.

[16] W. Hackbusch. *Elliptic Differential Equations. Theory and Numerical Treatment.* Springer-Verlag Berlin, 1992.

[17] W. Hackbusch. A sparse matrix arithmetic based on $\mathcal{H}$-matrices. Part I: Introduction to $\mathcal{H}$-matrices. *Computing*, 62:89–108, 1999.

[18] W. Hackbusch. *Hierarchische Matrizen — Algorithmen und Analysis.* Springer, 2009.

[19] W. Hackbusch and B. N. Khoromskij. A sparse matrix arithmetic based on $\mathcal{H}$-matrices. Part II: Application to multi-dimensional problems. *Computing*, 64:21–47, 2000.

[20] W. Hackbusch, B. N. Khoromskij, and S. A. Sauter. On $\mathcal{H}^2$-matrices. In H. Bungartz, R. Hoppe, and C. Zenger, editors, *Lectures on Applied Mathematics*, pages 9–29. Springer-Verlag, Berlin, 2000.

[21] W. Hackbusch and Z. P. Nowak. On the fast matrix multiplication in the boundary element method by panel clustering. *Numer. Math.*, 54:463–491, 1989.

[22] J. Makino. Yet another fast multipole method without multipoles — pseudoparticle multipole method. *J. Comp. Phys.*, 151:910–920, 1999.

[23] P.-G. Martinsson and V. Rokhlin. A fast direct solver for boundary integral equations in two dimensions. *J. Comp. Phys.*, 205:1–23, 2005.

[24] E. E. Tyrtyshnikov. Incomplete cross approximation in the mosaic-skeleton method. *Computing*, 64:367–380, 2000.

[25] L. Ying, G. Biros, and D. Zorin. A kernel-independent adaptive fast multipole algorithm in two and three dimensions. *J. Comp. Phys.*, 196(2):591–626, 2004.