

Article

A Fast Method for the Off-Boundary Evaluation of Laplace Layer Potentials by Convolution Sums

Wenchao Guan ¹ , Zhicheng Wang ², Leqi Xue ¹ and Yueen Hou ^{2,*}

¹ School of Computer Science and Engineering, Guangdong Province Key Lab of Computational Science, Sun Yat-sen University, Guangzhou 510275, China; guanwch3@mail.sysu.edu.cn (W.G.); xuelq@mail2.sysu.edu.cn (L.X.)

² Computer School, Jiaying University, Meizhou 514015, China; leemoon14@outlook.com

* Correspondence: houyueen@jyu.edu.cn

Abstract: In off-boundary computations of layer potentials, the near-singularities in integrals near the boundary presents challenges for conventional quadrature methods in achieving high precision. Additionally, the significant complexity of $\mathcal{O}(n^2)$ interactions between n targets and n sources reduces the efficiency of these methods. A fast and accurate numerical algorithm is presented for computing the Laplace layer potentials on a circle with a boundary described by a polar curve. This method can maintain high precision even when evaluating targets located at a close distance from the boundary. The radial symmetry of the integral kernels simplifies their description. By exploiting the polar form of the boundary and applying a one-dimensional exponential sum approximation along the radial direction, an approximation of layer potentials by the convolution sum is obtained. The algorithm uses FFT convolution to accelerate computation and employs a local quadrature to maintain accuracy for nearly singular terms. Consequently, it achieves spectral accuracy in regions outside of a sufficiently small neighborhood of the boundary and requires $\mathcal{O}(n \log n)$ arithmetic operations. With the help of this algorithm, layer potentials can be efficiently evaluated on a computational domain.

Keywords: layer potential; nearly singular integrals; numerical integration; approximation by exponential sums



Citation: Guan, W.; Wang, Z.; Xue, L.; Hou, Y. A Fast Method for the Off-Boundary Evaluation of Laplace Layer Potentials by Convolution Sums. *Symmetry* **2024**, *16*, 764. <https://doi.org/10.3390/sym16060764>

Academic Editor: Ioan Raşa

Received: 17 May 2024

Revised: 8 June 2024

Accepted: 12 June 2024

Published: 18 June 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The boundary integral equation (BIE) method is attractive for addressing boundary value problems of partial differential equations (PDEs) like the Laplace equation [1,2], Helmholtz equation [3], biharmonic equation [4], and other classical elliptic equations in mathematical physics. It offers a number of numerically attractive properties, including reducing the dimension of the computational problem by one, enabling boundary-only discretizations for homogeneous problems, and being well-suited for problems involving infinite or semi-infinite domains. Moreover, the BIE method is widely applied in physics and engineering, including inverse acoustic and electromagnetic scattering problems [3], multimedia elasticity problems [5], wetting problems [6], Navier–Stokes equations [7,8], etc.

The BIE method seems to have less advantage in accuracy and computational complexity when it recovers the solutions of boundary value problems from the BIE solutions, requiring the evaluation of layer potentials both near and away from the source layer. This off-boundary evaluation, particularly with complex boundaries, presents two numerical challenges: 1. The accuracy of conventional quadrature methods diminishes rapidly as the target point approaches the boundary due to the near-singularity of boundary integrals. 2. There is significant computing complexity for the $\mathcal{O}(n^2)$ interactions between n targets and n sources. To address the computational challenges posed by nearly singular integrals, many innovative computational methods have been proposed. These methods include quadrature by expansion [9–12], asymptotic approximation methods [13–15],

barycentric-type formulas leveraging Cauchy's integral formula [16,17], singularity swapping methods [18,19], kernel regularization methods [20,21], a method using the Stokes theorem [22], nonlinear transformation methods [23,24], etc. However, due to the reliance of computations on target points, most of these methods are rarely developed into fast methods that are capable of facilitating the evaluation of $\mathcal{O}(n^2)$ interactions in linear or quasi-linear (linear up to a logarithmic factor) time. The scheme based on coupling quadrature by expansion and a fast multipole method [25,26] is one such fast algorithm that is capable of accurately computing layer potentials near boundaries. Nevertheless, in order to overcome extra geometry refinement, this scheme needs to be carefully designed in cases for which some parts of the source geometry come so close to other parts that they almost touch each other [25].

In this paper, we propose a fast and accurate numerical algorithm for computing the layer potentials for Laplace equations. This method requires only quasi-linear computational costs and still achieves high precision in regions that exclude a sufficiently small neighborhood of the boundary. Moreover, it permits discontinuities of layer potentials in the field across the boundary and is capable of handling cases for which some parts of the source geometry are close to other parts without too much additional work. Consider the surface integral operator $\mathcal{A} : L^2(\Gamma_1) \rightarrow L^2(\Gamma_2)$ defined by

$$\mathcal{A}f(\mathbf{x}) := \int_{\Gamma_1} K(\mathbf{x}, \mathbf{y})f(\mathbf{y})ds_{\mathbf{y}}, \quad \mathbf{x} \in \Gamma_2, \quad (1)$$

where $\Gamma_1, \Gamma_2 \subset \mathbb{R}^2$ are simple closed curves and are of class \mathcal{C}^2 , $f \in L^2(\Gamma_1)$ is a density function, and K is defined and continuous for all $\mathbf{x} \in \Gamma_2, \mathbf{y} \in \Gamma_1, \mathbf{x} \neq \mathbf{y}$. Many numerical applications of boundary integral equation methods require the computation of $\mathcal{A}f$. For instance, solutions to boundary integral equations are frequently used to evaluate single- and double-layer potentials for applications in a broad range of simulations in physics and engineering. In addition, the application of Tikhonov regularization to solve boundary value problems using potentials with densities on curves different from the actual boundary of the underlying domain also necessitates the evaluation of $\mathcal{A}f$ [1]. For cases wherein Γ_1 is a closed polar curve and Γ_2 is an origin-centered circle, two algorithms are presented in this paper. Notice that the integral kernel K for Laplace equations in (1) can be expressed as a product of a singular part and a finite sum of separable functions. In order to construct a fast method, we concentrate on approximating the singular part using convolution sums. Due to the radial symmetry of the singular part, this only requires approximation along the radial direction. By utilizing the polar form of Γ_1, Γ_2 and applying exponential sum approximations to the singular part along the radial direction [27], we derive an efficient approximation using convolution sums. An error estimate for this approximation is provided in this paper. Leveraging this efficient approximation, we propose algorithms for computing the layer potentials for Laplace equations with a computational complexity of $\mathcal{O}(n \log n)$ for the evaluation of $\mathcal{O}(n^2)$ interactions. Currently, there are few fast layer methods capable of accurately computing potentials near boundaries, among which an important one is the fast QBX method [25,26]. Our proposed method also efficiently handles such cases. Unlike the fast QBX, our method does not require handling of the boundary partition. As noted in the summary of [25], some situations, such as when some parts of the source geometry come close to other parts, often require extra geometry refinement. Our method effectively addresses such cases. Moreover, this method accommodates discontinuities in the field across the boundary, which is a feature that is not commonly addressed by existing methods.

The structure of this paper is as follows. Section 2 introduces the approximation of $\mathcal{A}f$ by convolution sums and provides an analysis of the approximation error. We then propose two algorithms based on the approximation. Subsequently, in Section 3, we present some numerical experiments to examine the performance of the algorithms. Finally, we provide conclusions about the proposed methods and discuss their drawbacks in Section 4.

2. Fast Methods for the Evaluation of Layer Potentials

The kernel K of the integral (1) can be expressed as

$$K(\mathbf{x}, \mathbf{y}) = \psi(\mathbf{x}, \mathbf{y})\phi(|\mathbf{x} - \mathbf{y}|^2). \quad (2)$$

In the context of certain layer potentials, such as those for the Laplace, biharmonic, and Stokes equations [1,4,28], the function $\psi : \Gamma_2 \times \Gamma_1 \rightarrow \mathbb{C}$ can be represented as a finite sum of separable functions. Specifically, $\psi(\mathbf{x}, \mathbf{y}) = \sum_{k=1}^{N_\psi} \psi_k(\mathbf{x})\tilde{\psi}_k(\mathbf{y})$. Additionally, the function $\phi : (0, \infty) \rightarrow \mathbb{R}$ is a logarithmic function satisfying

$$\phi(\alpha x) = \phi(x) + \phi(\alpha),$$

or ϕ is homogeneous of $\gamma \in \mathbb{R}$; meaning that for all $\alpha > 0$,

$$\phi(\alpha x) = \alpha^\gamma \phi(x),$$

or ϕ is a linear combination of a logarithmic function, a homogeneous function, and their products, as is the case for the biharmonic equation [4]. In the case of the Laplace equation, the kernel for the single-layer potential can be expressed as (2), with $\psi(\mathbf{x}, \mathbf{y}) \equiv 1$ and $\phi(x) = -\frac{1}{4\pi} \log x$. Similarly, the kernel for the double-layer potential can also be represented as (2), with $\psi(\mathbf{x}, \mathbf{y}) = (\mathbf{y} - \mathbf{x}) \cdot \mathbf{v}(\mathbf{y})$, and $\phi(x) = -\frac{1}{2\pi x}$, where $\mathbf{v}(\mathbf{y})$ is the unit outer normal vector perpendicular to the curve Γ_1 at \mathbf{y} . To handle other kernels for equations like the Helmholtz equation [3,9], the singularity subtraction method is often used. This involves splitting the kernel into two parts: K_1 , which has the same leading-order singularity as K , and a smoother function $K - K_1$, which is easier to integrate numerically. The singular kernel K_1 often has the form (2). In this paper, we focus on the evaluation of the single- and double-layer potentials for the Laplace equation.

In the subsequent sections of this paper, we suppose that Γ_1 in (1) is a boundary of a star domain and is described by the parametrization

$$\mathbf{r}(t) := (r(t) \cos(t), r(t) \sin(t)), t \in I_{2\pi} := [0, 2\pi), \quad (3)$$

where r is a twice-differentiable, 2π -periodic, positive function defined on \mathbb{R} . We suppose that Γ_2 in (1) is a circle with a radius $R > 0$ defined by

$$\mathbf{c}_R(t) := (R \cos(t), R \sin(t)), t \in I_{2\pi}. \quad (4)$$

The assumption of polar parameterization is made here to enable the squared distance function to take a form that closely resembles a convolution, as expressed in (5). This form of the squared distance function is used to construct the layer potential in the form of convolution sums, with ϕ approximated by exponential sums. With the parametrizations (3) and (4), let us define

$$\mathcal{A}f(\eta) := \mathcal{A}f(\mathbf{c}_R(\eta)) = \int_{I_{2\pi}} K(\mathbf{c}_R(\eta), \mathbf{r}(\theta)) \tilde{f}(\theta) d\theta, \quad \eta \in I_{2\pi},$$

where $\tilde{f}(\theta) := f \circ \mathbf{r}(\theta) |\mathbf{r}'(\theta)|$, $\theta \in I_{2\pi}$. To ensure clarity, we adopt this definition of $\mathcal{A} : L^2(\Gamma_1) \rightarrow L^2(I_{2\pi})$ throughout the subsequent discussions. It is important to note that when $\Gamma_1 \cap \Gamma_2 = \emptyset$, meaning that the targets are distant from the sources, the layer potential $\mathcal{A}f$ is smooth. However, due to the jump relations of the layer potential across the boundary, when $\Gamma_1 \cap \Gamma_2 \neq \emptyset$, $\mathcal{A}f$ is not smooth. Specifically, in the case of the Laplace single-layer potential, the derivative of $\mathcal{A}f$ is discontinuous at the intersection of Γ_1 and Γ_2 , and in the case of the Laplace double-layer potential, $\mathcal{A}f$ itself is discontinuous at the intersection [1].

2.1. An Approximation of $\mathcal{A}f$ by Convolution Sums

In this subsection, we propose an approximation of $\mathcal{A}f$ by convolution sums, which regularizes $\mathcal{A}f$ and is utilized in the development of a fast algorithm in the next subsection. With the parametrizations (3) and (4), we write the square of the distance function between the points at Γ_1 and Γ_2 as

$$d(\eta, \theta) := |\mathbf{c}_R(\eta) - \mathbf{r}(\theta)|^2 = (R - r(\theta))^2 + 4Rr(\theta) \sin^2\left(\frac{\eta - \theta}{2}\right). \quad (5)$$

In order to utilize the exponential sum approximation described in the following content, we rewrite $d(\eta, \theta)$ as

$$d(\eta, \theta) = d_{1,\beta}(\eta, \theta)d_{2,\beta}(\theta),$$

where

$$d_{1,\beta}(\eta, \theta) = \frac{|\mathbf{c}_R(\eta) - \mathbf{r}(\theta)|^2}{d_{2,\beta}(\theta)} = \frac{(R - r(\theta))^2}{\beta R r(\theta)} + \frac{4}{\beta} \sin^2\left(\frac{\eta - \theta}{2}\right), \text{ and } d_{2,\beta}(\theta) = \beta R r(\theta),$$

with $\beta > 0$ denoting a scale parameter.

For the case of the Laplace single-layer potential, from $\psi = 1$ and $\phi(x) = -\frac{1}{4\pi} \log x$ in (2), it follows that

$$\mathcal{A}f(\eta) = \int_{I_{2\pi}} \phi \circ d_{1,\beta}(\eta, \theta) \tilde{f}(\theta) d\theta + \int_{I_{2\pi}} \phi \circ d_{2,\beta}(\theta) \tilde{f}(\theta) d\theta. \quad (6)$$

According to Appendix A.1, the function ϕ can be approximated by M -term exponential sums on $[1, \infty)$. This approximation is based on the techniques described in [27] and is given by

$$\phi_M(x) = \sum_{k=0}^M c_k e^{-a_k x} - b_M, \quad (7)$$

where $a_k, b_k, c_k \in \mathbb{R}$, $a_0 = \log 2$, and $a_k > 0$ decreases with respect to k . Selecting an appropriate β and replacing ϕ in the first term on the right-hand side of (6) with ϕ_M yields the approximation for the Laplace single-layer potential:

$$\mathcal{S}_{M,\beta} f(\eta) := \int_{I_{2\pi}} \phi_M \circ d_{1,\beta}(\eta, \theta) \tilde{f}(\theta) d\theta + \int_{I_{2\pi}} \phi \circ d_{2,\beta}(\theta) \tilde{f}(\theta) d\theta. \quad (8)$$

The first term on the right-hand side of (8) can be represented as a sum of periodic convolutions, and this approximation is expressed as follows:

$$\mathcal{S}_{M,\beta} f = \sum_{k=0}^M c_k \mu_{k,\beta} * (\omega_{k,\beta} \tilde{f}) + \mathcal{I}_{M,\beta} \tilde{f},$$

where

$$\mu_{k,\beta}(\theta) := e^{-\frac{a_k}{\beta} 4 \sin^2\left(\frac{\theta}{2}\right)}, \omega_{k,\beta}(\theta) := e^{-\frac{a_k}{\beta} \frac{(R-r(\theta))^2}{Rr(\theta)}}, \quad (9)$$

and

$$\mathcal{I}_{M,\beta} \tilde{f} := \int_{I_{2\pi}} (\phi \circ d_{2,\beta}(\theta) - b_M) \tilde{f}(\theta) d\theta.$$

For the case of the Laplace double-layer potential, from $\psi(\mathbf{x}, \mathbf{y}) = (\mathbf{y} - \mathbf{x}) \cdot \mathbf{v}(\mathbf{y})$ and $\phi(x) = -\frac{1}{2\pi x}$, it follows that

$$\begin{aligned} \mathcal{A}f(\eta) &= \int_{I_{2\pi}} \phi \circ d_{1,\beta}(\eta, \theta) \mathbf{n}(\theta) \cdot \mathbf{r}(\theta) \tilde{f}(\theta) / d_{2,\beta}(\theta) d\theta \\ &\quad - \mathbf{c}_R(\eta) \cdot \int_{I_{2\pi}} \phi \circ d_{1,\beta}(\eta, \theta) \mathbf{n}(\theta) \tilde{f}(\theta) / d_{2,\beta}(\theta) d\theta, \end{aligned} \quad (10)$$

where $\mathbf{n}(\theta) := \mathbf{v}(\mathbf{r}(\theta))$. Here, the function ϕ can be approximated by M -term exponential sums:

$$\phi_M(x) = \sum_{k=0}^M \tilde{c}_k e^{-a_k x}, \tag{11}$$

on $[1, \infty)$, where a_k, \tilde{c}_k are defined in Appendix A.2. Similar to the evaluation method employed for the Laplace single-layer potential, replacing ϕ in (10) with ϕ_M gives

$$\begin{aligned} \mathcal{D}_{M,\beta} f(\eta) &:= \int_{I_{2\pi}} \phi_M \circ d_{1,\beta}(\eta, \theta) \mathbf{n}(\theta) \cdot \mathbf{r}(\theta) \tilde{f}(\theta) / d_{2,\beta}(\theta) d\theta \\ &+ \mathbf{c}_R(\eta) \cdot \int_{I_{2\pi}} \phi_M \circ d_{1,\beta}(\eta, \theta) \mathbf{n}(\theta) \tilde{f}(\theta) / d_{2,\beta}(\theta) d\theta. \end{aligned}$$

The function $\mathcal{D}_{M,\beta} f$ can be rewritten as

$$\mathcal{D}_{M,\beta} f = \sum_{k=0}^M \tilde{c}_k \mu_{k,\beta} * (\omega_{k,\beta} \tilde{\mathbf{f}}_1) - \mathbf{c}_R \cdot \sum_{k=0}^M \tilde{c}_k \mu_{k,\beta} * (\omega_{k,\beta} \tilde{\mathbf{f}}_2),$$

where $\tilde{\mathbf{f}}_1 := \mathbf{n} \cdot \mathbf{r} \tilde{f} / d_{2,\beta}$, $\tilde{\mathbf{f}}_2 := \mathbf{n} \tilde{f} / d_{2,\beta}$, and $\mu_{k,\beta}, \omega_{k,\beta}$ are defined in (9). Notice that in higher dimensions, the principal singularity of the Laplace layer potential remains radially symmetric. Therefore, it suffices to work with exponential approximations of ϕ to develop higher-dimensional convolution approximations.

For convenience, we denote the operator $\mathcal{A}_{M,\beta} = \mathcal{S}_{M,\beta}$ for the Laplace single-layer potential and $\mathcal{A}_{M,\beta} = \mathcal{D}_{M,\beta}$ for the Laplace double-layer potential. It is noteworthy that $\mathcal{A}_{M,\beta} f$ exhibits smoothness and approximates $\mathcal{A}f$ notably well at points distant from Γ_1 . The parameter M controls the accuracy of approximation, while β determines the distance from Γ_1 for approximating $\mathcal{A}f$. Hence, $\mathcal{A}_{M,\beta} f$ serves as regularization for $\mathcal{A}f$ and is governed by the parameters M and β . In Sections 2.3 and 2.4, we propose methods for computing $\mathcal{A}_{M,\beta} f$. Preceding that, in the upcoming section, we estimate the error between $\mathcal{A}_{M,\beta} f$ and $\mathcal{A}f$, which is referred to as the smoothing error in this paper.

2.2. The Smoothing Error

In this section, we estimate the smoothing error $\|\mathcal{A}f - \mathcal{A}_{M,\beta} f\|_{\infty,\lambda}$, where $\|g\|_{\infty,\lambda} := \sup_{\mathbf{x} \in \Gamma_2 \setminus \Gamma_{1,\lambda}} |g(\mathbf{x})|$ for a piecewise continuous function g defined on Γ_2 , $\Gamma_{1,\lambda} := \{\mathbf{x} : |\mathbf{x} - \mathbf{y}| < \lambda, \mathbf{y} \in \Gamma_1\}$, $\beta \in (0, \frac{\lambda^2}{Rr_{\max}}]$ with some $\lambda > 0$ in (8), and $r_{\max} := \max_{\theta \in I_{2\pi}} r(\theta)$. In order to present the smoothing error estimate, we define $r_{\min} := \min_{\theta \in I_{2\pi}} r(\theta)$ and introduce the notation $\|g\|_{\infty,H} := \sup_{x \in H} |g(x)|$, $H \subset [1, \infty)$, for a continuous function g defined on $[1, \infty)$. The next two theorems present the smoothing error estimate for the Laplace single- and double-layer potentials. The proofs of these theorems are in Appendix B.

Theorem 1. *In the case of the Laplace single-layer potential, there exist positive constants C_1, C_2 such that for all $\lambda, R > 0, M \in \mathbb{Z}^+$ and $f \in L^2(\Gamma_1)$, by choosing $h = \log(4\pi M)/M$ and $\beta \in (0, \frac{\lambda^2}{Rr_{\max}}]$,*

$$\begin{aligned} \|\mathcal{A}f - \mathcal{S}_{M,\beta} f\|_{\infty,\lambda} &\leq \left(C_1 \iota \exp\left(-\frac{\pi^2 M}{\log(3\iota) \log(\pi^2 M)}\right) + \right. \\ &\left. \left(2^{-s} - \left(1 + e^{\sinh(\log(4\pi M))} \right)^{-s} \right) / (4\pi s \log 2) + C_2 s 2^{-s} \log(4\pi M)^2 / M \right) \|f\|_1, \end{aligned} \tag{12}$$

where $s = \lambda^2 / (\beta R r_{\max}), \iota = (R + r_{\max})^2 / (\beta R r_{\min})$.

Theorem 2. In the case of the Laplace double-layer potential, there exist positive constants C_1, C_2 such that for all $\lambda, R > 0, M \in \mathbb{Z}^+$ and $f \in L^2(\Gamma_1)$, by choosing $h = \log(4\pi M)/M$ and $\beta \in \left(0, \frac{\lambda^2}{Rr_{\max}}\right]$,

$$\|\mathcal{A}f - \mathcal{D}_{M,\beta}f\|_{\infty,\lambda} \leq \left(C_1 \exp\left(-\frac{\pi^2 M}{\log(3\iota) \log(\pi^2 M)}\right) + \left(2^{-s} - \left(1 + e^{\sinh(\log(4\pi M))}\right)^{-s}\right) / (2\pi s) + C_2 s 2^{-s} \log(4\pi M)^2 / M \right) \tau \|f\|_1, \quad (13)$$

where $s = \lambda^2 / (\beta R r_{\max}), \iota = (R + r_{\max})^2 / (\beta R r_{\min}), \tau = \max_{x \in I_{2\pi}} |\mathbf{r}'(x)| (1/R + 1/r_{\min}) / \beta$.

Theorems 1 and 2 indicate that the smoothing error of $\mathcal{A}_{M,\beta}f$ decreases exponentially on $\Gamma_2 \setminus \Gamma_{1,\lambda}$ as the parameters M and β increase. In these theorems, the selection strategy of h is based on the Sinc quadrature error estimates from Proposition 2.1 in [27], ensuring the exponential decay of the smoothing error with M . Notice that the second term on the right-hand side of (12) and (13) depends on s , and $s = \lambda^2 / (\beta R r_{\max})$. Selecting a sufficiently small β ensures this second term becomes negligible. We choose $\beta = \lambda^2 / (50Rr_{\max})$ to ensure $2^{-s} = 2^{-50} \approx 8.9 \times 10^{-16}$, nearly reaching double precision relative to the machine epsilon. Thus, the second term on the right-hand side of (12) and (13) can be ignored relative to the first term within machine precision. The choice of β also determines $\iota = 50(R + r_{\max})^2 r_{\max} / (\lambda^2 r_{\min})$, which increases as λ decreases. Thus, in order to ensure that the first term on the right-hand side of (12) and (13) is sufficiently small, an appropriate value of M needs to be found based on ι . Theorem 1 suggests that selecting $M = C(\log \iota) + (\log \iota)^2$ with a positive constant C ensures the smoothing error in cases for which the single-layer potential remains bounded regardless of ι and λ . In addition, Theorem 2 suggests $M = C(\log \iota)$ for the double-layer potential. In summary, in this paper, we select $\beta = \lambda^2 / (50Rr_{\max}), M = -C_s(\log \beta) + (\log \beta)^2$ for the single-layer potential and $M = -C_d(\log \beta)$ for the double-layer potential, where $C_s, C_d > 0$.

2.3. A Method for Computing $\mathcal{A}_{M,\beta}f$ Using FFT Convolution

The function $\mathcal{A}_{M,\beta}f$, introduced in Section 2.1 for both the single- and double-layer potential cases, takes the form

$$\mathcal{A}_{M,\beta}f = \mathbf{C} \cdot \left(\sum_{k=0}^M w_k \mu_{k,\beta} * (\omega_{k,\beta} \mathbf{g}) \right) + B_{M,\beta}, \quad (14)$$

with $\mathbf{C} = [1], w_k = c_k, \mathbf{g} = [\tilde{f}]$, $B_{M,\beta} = \mathcal{I}_{M,\beta} \tilde{f}$ for the Laplace single potential, and $\mathbf{C} = [1, -\mathbf{c}_R], w_k = \tilde{c}_k, \mathbf{g} = [\tilde{f}_1, \tilde{f}_2], B_{M,\beta} = 0$ for the Laplace double potential. Notice that $\mathcal{A}_{M,\beta}f$ consists of a periodic convolution sum. It is well known that applying the FFT convolution method leads to a fast evaluation of $\mathcal{A}_{M,\beta}f$.

Denote $\mathbb{Z}_k := \{0, 1, \dots, k-1\}$, $k \in \mathbb{Z}^+$. Given $n \in \mathbb{Z}^+$, the values of \mathbf{r} and $|\mathbf{r}'|$ on $I_{2\pi,n} := \{2k\pi/n : k \in \mathbb{Z}_n\}$, and the values of the density function on $\mathbf{S}_n := \{\mathbf{r}(\theta_k) : \theta_k \in I_{2\pi,n}\}$. We now present a fast algorithm for computing the values of $\mathcal{A}_{M,\beta}f$ on $\mathbf{T}_{R,n} := \{\mathbf{c}_R(\theta_k) : \theta_k \in I_{2\pi,n}\}$, with a radius $R > 0$.

We next present an estimate of the computational costs of Algorithm 1. Let $\mathcal{M}_{M,n}$ denote the number of multiplications used in Algorithm 1.

Algorithm 1 An algorithm for computing $\mathcal{A}_{M,\beta}f$ using n -point FFT convolution.

Input: $M, n \in \mathbb{Z}^+, \beta, R > 0$, a function f on \mathbf{S}_n , and $\mathbf{r}, |\mathbf{r}'|$ defined on $I_{2\pi,n}$.
Output: Approximate values of $\mathcal{A}_{M,\beta}f$ on $\mathbf{T}_{R,n}$.

Step 1: Compute the periodic convolution of $\mu_{k,\beta} * (\omega_{k,\beta}\mathbf{g})(\mathbf{T}_{R,n})$ using the n -point FFT method, denoted as $\mathbf{I}_{k,\beta}(\mathbf{T}_{R,n})$ for $k = 0, \dots, M$.

Step 2: Compute the integral $B_{M,\beta}$ using the n -point periodic trapezoidal rule.

Step 3: Compute the approximate values of $\mathcal{A}_{M,\beta}f$ on $\mathbf{T}_{R,n}$ by $\mathbf{C} \cdot \sum_{k=0}^M w_k \mathbf{I}_{k,\beta}(\mathbf{T}_{R,n}) + B_{M,\beta}$.

Theorem 3. There exists $C > 0$ such that for all $M, n \in \mathbb{Z}^+$,

$$\mathcal{M}_{M,n} \leq CMn \log_2 n.$$

Proof. The number of multiplications used in Step 1 is bounded by $\mathcal{O}(Mn \log_2 n)$. The number of multiplications used in Step 2 is not greater than the amount of $\mathcal{O}(n)$. The number of multiplications used in Step 3 is bounded by $\mathcal{O}(Mn)$. Therefore, we obtain the $\mathcal{O}(Mn \log_2 n)$ upper bound on the total number of multiplications used in Algorithm 1. \square

Denote the output of Algorithm 1 as $\mathcal{A}_{M,\beta,FFT}f(\mathbf{T}_{R,n})$. The computational error $\mathcal{A}f(\mathbf{T}_{R,n}) - \mathcal{A}_{M,\beta,FFT}f(\mathbf{T}_{R,n})$ comprises both the smoothing error and the error in convolution computation. We estimated the smoothing error in $\Gamma_2 \setminus \Gamma_{1,\lambda}$ in Section 2.2. We turn to discuss the convolution error. The aliasing error of the discrete Fourier transform and the truncation error of the Fourier series contribute to the convolution error when using FFT. The decay rates of the coefficients of the Fourier series of $\mu_{k,\beta}$ and $\omega_{k,\beta}\mathbf{g}$ determine this convolution computation error. It is important to note that as β decreases, the decay rates of $\mu_{k,\beta}$ and $\omega_{k,\beta}\mathbf{g}$ also decrease, as illustrated in Figure 1, indicating the near-singularity of the layer potential. This leads to a decrease in the accuracy of computing the convolution using the n -point FFT. Indeed, when a_k/β is large, $\mu_{k,\beta}(x) \approx \sum_{j \in \mathbb{Z}} G_{a_k/\beta}(x + 2j\pi)$ for $x \in \mathbb{R}$, where $G_{a_k/\beta} := \exp(-a_k x^2/\beta)$. According to the Fourier transform of $G_{a_k/\beta}$, the j th Fourier coefficient of $\sum_{j \in \mathbb{Z}} G_{a_k/\beta}(x + 2j\pi)$ is $\sqrt{\beta/(2a_k)} \exp(-\beta j^2/(4a_k))$. Therefore, the Fourier coefficients of $\mu_{k,\beta}$ decay nearly as $\mathcal{O}(\exp(-\beta j^2/(4a_k)))$.

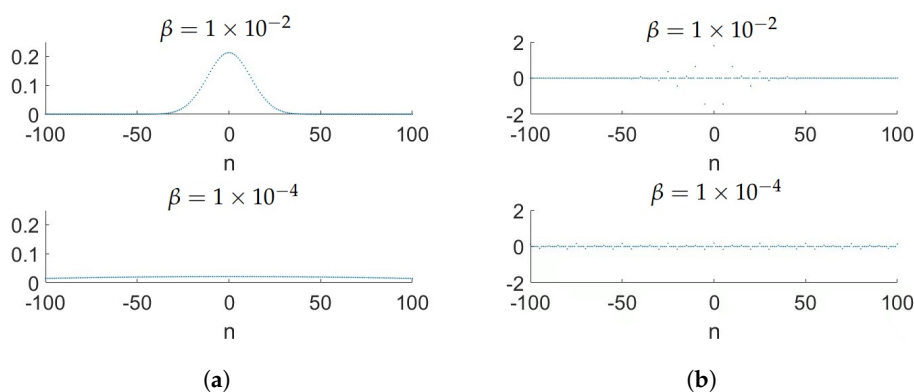


Figure 1. The Fourier coefficients of $\mu_{0,\beta}$ and $\omega_{0,\beta}$. The radius function of the boundary Γ_1 is $r(t) = 1 + 0.3 \cos(5t)$, and the target point is located at $(0.66, 0.37)$, which is a distance of 6×10^{-3} from the boundary. Top row: $\beta = 1 \times 10^{-2}$. Bottom row: $\beta = 1 \times 10^{-4}$. (a) The Fourier coefficients of $\mu_{0,\beta}$. (b) The Fourier coefficients of $\omega_{0,\beta}$.

2.4. An Improved Method for Computing $\mathcal{A}_{M,\beta}f$

In order to avoid inaccuracies in the computation of $\mu_\sigma * (\omega_\sigma\mathbf{g})$ in (14) using the FFT convolution method with very small β values, we directly evaluate it using local quadrature methods. Let $g \in L^1(I_{2\pi})$ and $\sigma \geq 0$. We define the periodic convolution as follows:

$$\mathcal{C}_\sigma g := \mu_\sigma * (\omega_\sigma g),$$

where

$$\mu_\sigma(\theta) := e^{-4\sigma \sin^2(\frac{\theta}{2})}, \quad \omega_{k,\beta}(\theta) := e^{-\sigma \frac{(R-r(\theta))^2}{Rr(\theta)}}, \quad \theta \in \mathbb{R}.$$

The form (14) can be rewritten as

$$\mathcal{A}_{M,\beta} f = \mathbf{C} \cdot \left(\sum_{k=0}^M w_k \mathcal{C}_{\frac{a_k}{\beta}} \mathbf{g} \right) + B_{M,\beta}.$$

We now address the computation of $\mathcal{C}_\sigma g$ with large values of σ . Suppose $\mathcal{C}_\sigma g$ can be accurately computed using the FFT convolution method with $\sigma \leq \sigma_0$ for a sufficiently large σ_0 . Note that the function μ_σ exhibits a sharp peak at $\theta = 0$ with a large σ . Thus, instead of the convolution method, we apply the Gauss quadrature to $\mathcal{C}_\sigma g$. For the machine epsilon $\epsilon > 0$, we define $C_\epsilon := 2\sqrt{\sigma} \arcsin \frac{\sqrt{-\log \epsilon}}{2\sqrt{\sigma}} \approx \sqrt{-\log \epsilon}$. This ensures $\mu_\sigma(\theta) < \epsilon$ for $\theta \in [-\pi, \pi] \setminus [-C_\epsilon/\sqrt{\sigma}, C_\epsilon/\sqrt{\sigma}]$. Here, we set $C_\epsilon = 6$ to make μ_σ on $[-\pi, \pi] \setminus [-C_\epsilon/\sqrt{\sigma}, C_\epsilon/\sqrt{\sigma}]$ reach double precision. Neglecting the integral part of $[\eta - \pi, \eta + \pi] \setminus [\eta - C_\epsilon/\sqrt{\sigma}, \eta + C_\epsilon/\sqrt{\sigma}]$ yields

$$\mathcal{C}_\sigma g(\eta) \approx \int_{\eta - C_\epsilon/\sqrt{\sigma}}^{\eta + C_\epsilon/\sqrt{\sigma}} \mu_\sigma(\eta - \theta) \omega_\sigma(\theta) g(\theta) d\theta.$$

Applying the m -point Gauss quadrature to the above integral, denoted by $\mathcal{C}_\sigma^m g(\eta)$, provides an evaluation of $\mathcal{C}_\sigma g(\eta)$. Here, the values of g at the quadrature points can be obtained by the Fourier interpolation from $g(\mathbf{S}_n)$. This process can be accelerated by converting the Fourier interpolation to B-spline interpolation, allowing for the computation of the m quadrature points in linear time. We also write $\mathcal{C}_\sigma^m g(\mathbf{x})$, $\mathbf{x} = \mathbf{c}_R(\eta)$ in place of $\mathcal{C}_\sigma^m g(\eta)$ for simplicity. Given a continuous function g and a point \mathbf{x} , $\mathcal{C}_\sigma g(\mathbf{x})$ decays with respect to σ , as illustrated in Figure 2a. Figure 2b–d illustrate the relative error of the Gauss–Chebyshev quadrature against σ with 25, 40, and 55 integral points, respectively. As the number of integral points increases, the relative error decreases. In addition, the relative error remains bounded within a certain range of σ , indicating that the absolute error decays with respect to σ . This demonstrates that $\mathcal{C}_\sigma g(\mathbf{x})$ with a large σ can be efficiently computed using the Gauss quadrature. Furthermore, due to the rapid decay of the integrand with a large σ , we neglect computing $\mathcal{C}_\sigma^m g$ at the points \mathbf{x} in Γ_2 such that $\text{dist}(\mathbf{x}, \Gamma_1) > C_\epsilon \sqrt{Rr_{\max}}/\sigma$ for the machine epsilon.

Let $M_x \in \mathbb{Z}$ satisfy $a_{M_x} \leq x < a_{M_x-1}$ if $x \in [a_M, a_0)$, and $M_x = 0$ if $x \geq a_0$. The Algorithm 2 derived from the above discussion is as follows:

Algorithm 2 An algorithm using the n -point FFT convolution and the m -point Gauss quadrature.

Input: $M, m, n \in \mathbb{Z}^+, \beta, \sigma_0, R > 0$, a function f on \mathbf{S}_n , and $\mathbf{r}, |\mathbf{r}'|$ defined on $I_{2\pi, n}$.

Output: Approximate values of $\mathcal{A}_{M,\beta} f$ on $\mathbf{T}_{R,n}$.

Step 1: Compute the periodic convolution of $\mathcal{C}_{\frac{a_k}{\beta}} \mathbf{g}(\mathbf{T}_{R,n})$ using the n -point FFT method, denoted as $\mathbf{I}_{k,\beta}(\mathbf{T}_{R,n})$, for $k = M_{\beta\sigma_0}, \dots, M$.

Step 2: Compute $Q_k^m \mathbf{g}(\mathbf{T}_{R,n})$ by the Gauss quadrature for $k = 0, \dots, M_{\beta\sigma_0} - 1$, with $Q_k^m \mathbf{g}(\mathbf{x}) = \mathcal{C}_{\frac{a_k}{\beta}}^m \mathbf{g}(\mathbf{x})$ for $\mathbf{x} \in \mathbf{T}_{R,n}^{k,\beta} := \{\mathbf{x} \in \mathbf{T}_{R,n} : \text{dist}(\mathbf{x}, \Gamma_1) \leq 6\sqrt{Rr_{\max}\beta/a_k}\}$ and $Q_k^m \mathbf{g}(\mathbf{x}) = 0$ for $\mathbf{x} \in \mathbf{T}_{R,n} \setminus \mathbf{T}_{R,n}^{k,\beta}$.

Step 3: Compute the integral $B_{M,\beta}$ using the n -point periodic trapezoidal rule.

Step 4: Compute the approximate values of $\mathcal{A}_{M,\beta} f$ on $\mathbf{T}_{R,n}$ by

$$\mathbf{C} \cdot \left(\sum_{k=M_{\beta\sigma_0}}^M w_k \mathbf{I}_{k,\beta}(\mathbf{T}_{R,n}) + \sum_{k=0}^{M_{\beta\sigma_0}-1} w_k Q_k^m \mathbf{g}(\mathbf{T}_{R,n}) \right) + B_{M,\beta}.$$

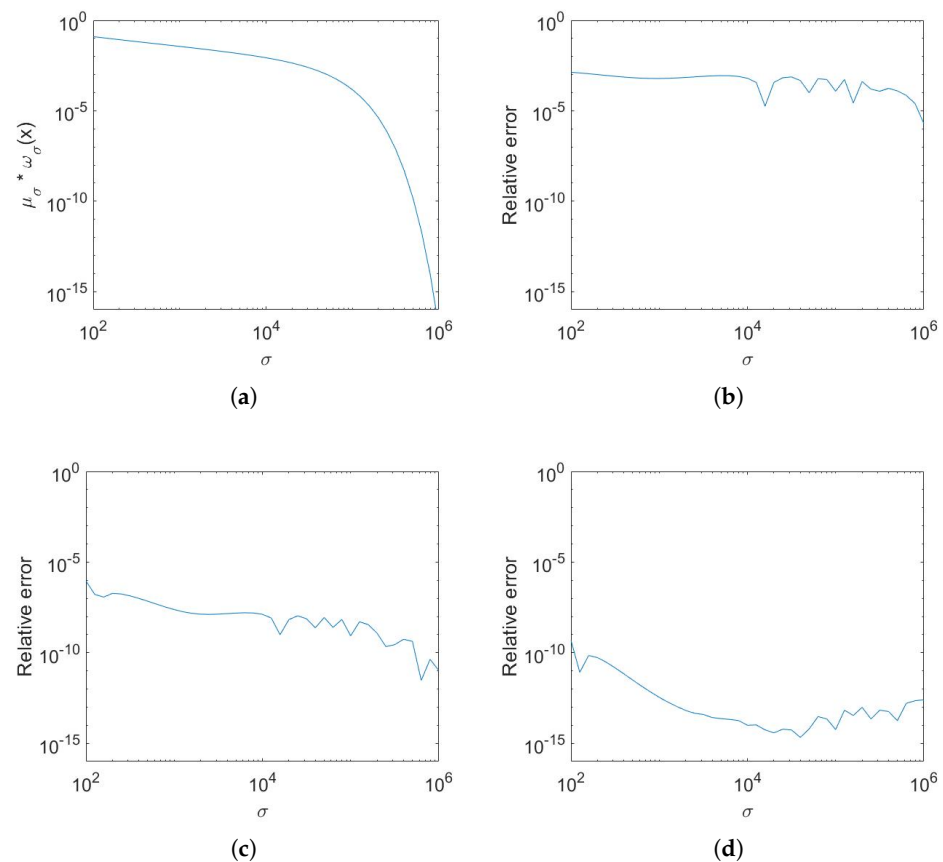


Figure 2. The values of $\mu_\sigma * \omega_\sigma(0.66, 0.37)$ and the relative error of the Gauss–Chebyshev quadrature against σ . The radius function of the boundary Γ_1 is $r(t) = 1 + 0.3 \cos(5t)$, and the target point is at a distance of 6×10^{-3} from the boundary. **(a)** The values of $\mu_\sigma * \omega_\sigma(\mathbf{x})$, $\mathbf{x} = (0.66, 0.37)$. **(b)** The relative error of the 25-point Gauss quadrature. **(c)** The relative error of the 40-point Gauss quadrature. **(d)** The relative error of the 55-point Gauss quadrature.

To close this section, we present an estimate of the computational costs of Algorithm 2. Let $\mathcal{M}_{M,m,n,\beta\sigma_0}^Q$ denote the number of multiplications used in Algorithm 2.

Theorem 4. *There exists $C > 0$ such that for all $R, \beta, \sigma_0 > 0$ and $M, m, n \in \mathbb{Z}^+$,*

$$\mathcal{M}_{M,m,n,\beta\sigma_0}^Q \leq Cn((M - M_{\beta\sigma_0}) \log_2 n + M_{\beta\sigma_0} \xi m),$$

where $\xi \in [0, 1]$ satisfies $|\mathbf{T}_{R,\ell}^{k,\beta}| \leq \xi \ell$ for all $k = 0, \dots, M_{\beta\sigma_0} - 1$, $\ell \in \mathbb{Z}^+$.

Proof. The number of multiplications used in Step 1 is bounded by $\mathcal{O}((M - M_{\beta\sigma_0} + 1)n \log_2 n)$. The number of multiplications used in Step 2 is not greater than $\mathcal{O}(M_{\beta\sigma_0} \xi mn + n \log_2 n)$. The number of multiplications used in Step 3 is bounded by $\mathcal{O}(n)$. The number of multiplications used in Step 4 is bounded by $\mathcal{O}((M - M_{\beta\sigma_0} + 1)n + M_{\beta\sigma_0} \xi n)$. Therefore, we obtain the result. \square

In the next section, the performances of the periodic trapezoidal rule and Algorithms 1 and 2 are presented. The periodic trapezoidal rule is a straightforward quadrature method to implement. This quadrature is exponentially accurate for computing real analytic functions [29]. Thus, it has high accuracy in computing layer potentials that are far from boundaries. However, when computing layer potentials near boundaries, the method converges very slowly due to the near-singularity of the integral. Additionally, its complexity

is $\mathcal{O}(n^2)$ for computing interactions between n targets and n sources. Theorem 3 indicates that Algorithm 1 has a complexity of $\mathcal{O}(n \log n)$. However, as discussed in Section 2.3, the convolution error for the nearly singular terms leads to inaccuracy near the boundary, similar to the periodic trapezoidal rule. Algorithm 2 improves the accuracy near the boundary by using the local Gauss quadrature for the nearly singular terms instead of the FFT convolution method. Moreover, Theorem 4 indicates that Algorithm 2 still maintains a complexity of $\mathcal{O}(n \log n)$.

3. Numerical Experiments

In this section, we present illustrative examples demonstrating the performance of the methods by evaluating layer potentials using the proposed methods. We begin by presenting the results for double-layer potentials when Γ_1 and Γ_2 intersect, showing errors near the intersection points with the periodic trapezoidal rule (PTR) and Algorithms 1 and 2. We then investigate the performance of these algorithms when Γ_2 is closed to Γ_1 . Finally, we employ these methods as components of an integral equation solver for Laplace Neumann boundary value problems. MATLAB (R2020a) is used for these implementations. All test computations are conducted on a PC equipped with a 3.6 GHz AMD Ryzen 5 3600 CPU and 32 GB memory.

The starfish curve of Figure 3a that we use for our numerical experiments is given by

$$\mathbf{r}(t) = (1 + 0.3 \cos(5t)) \begin{pmatrix} \cos(t) \\ \sin(t) \end{pmatrix}, \quad (15)$$

and the limaçon of Figure 3b is given by

$$\mathbf{r}(t) = (10 - (9 + 10^{-3}) \cos(t)) \begin{pmatrix} \cos(t) \\ \sin(t) \end{pmatrix}. \quad (16)$$

In each case, $t \in I_{2\pi}$.

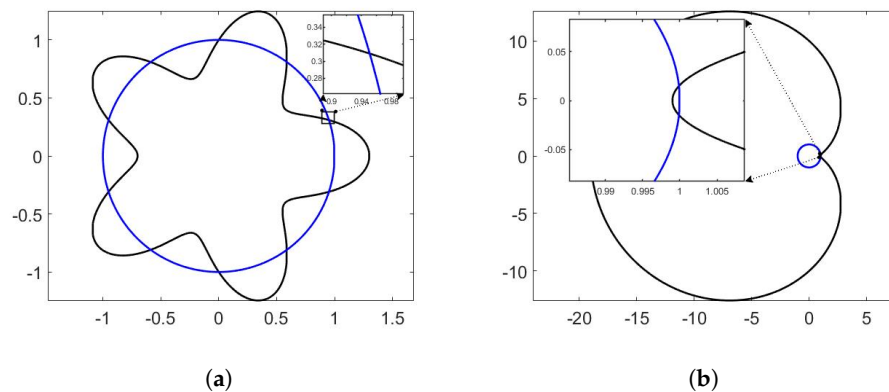


Figure 3. Test curves Γ_1 (black) and Γ_2 (blue). (a) A starfish curve (black) and a unit circle (blue). (b) A limaçon (black) and a unit circle (blue).

3.1. Double-Layer Potential Evaluation

The experiments with intersecting curves in Figure 3 and curves in close proximity in Figure 6a examine the performance of the proposed algorithm near the boundary Γ_1 . To this end, we employ these methods to compute the double-layer potential $\mathcal{D}f_d$ with a source function f_d . In order to derive an analytical expression of $\mathcal{D}f_d$ in the interior and exterior domains for result verification, we chose $f_d = 1$ on Γ_1 , and the corresponding double-layer potential satisfies

$$\mathcal{D}f_d(\mathbf{x}) = \begin{cases} -1, & \mathbf{x} \in \Omega, \\ -1/2, & \mathbf{x} \in \Gamma_1, \\ 0, & \mathbf{x} \in \mathbb{R}^2 \setminus \bar{\Omega}, \end{cases}$$

where Ω is the domain enclosed by Γ_1 .

The geometries of the first two examples with the boundaries described by (15) and (16) are shown in Figure 3. We compute 10^4 equispaced points on the unit circle by employing n -point PTR and Algorithms 1 and 2, with $n = 10^4$. In Algorithms 1 and 2, we set $\beta = \lambda^2 / (50Rr_{\max})$, $\lambda = 10^{-3}$, and $M = \lceil -40 \log_{10} \beta \rceil$ according to the analysis in Section 2.2. Here, the selection of λ ensures the accuracy of the layer potentials at distances greater than 10^{-3} from the boundary. In addition, we select $m = 55$ and $\sigma_0 = 10^5$ in Algorithm 2. The parameter m determines the number of integral points for the local Gauss quadrature used to handle the nearly singular terms. The relative errors shown in Figure 2 indicate that $m = 55$ is an appropriate choice, as the local quadrature error reaches double-precision. The σ_0 sets the range of σ values at which the fast convolution method is employed to compute $\mathcal{C}_{\sigma}g$, as discussed in Section 2.4. A larger σ_0 means more terms are evaluated using the FFT convolution method, which can result in a loss of accuracy due to the near singularity of some convolution terms.

The first example presents the error of computing $\mathcal{D}f_d$ on a unit circle Γ_2 with a starfish Γ_1 in Figure 3a. For simplicity, we write $\tilde{\mathcal{D}}f_d(\eta)$, $\eta \in \mathbb{R}$ instead of $\mathcal{D}f_d(\mathbf{c}_1(\eta))$. Figure 4a shows $\tilde{\mathcal{D}}f_d(\eta)$ for $\eta \in (0.09\pi, 0.11\pi)$. Note that the double-layer potential has a jump at $\eta = \pi/10$ since Γ_1 and Γ_2 intersect. Figure 4b–d illustrate the error of PTR and Algorithms 1 and 2, respectively, with $n = 10^4$. Algorithms 1 and 2 use the parameter setting discussed above. Compared to PTR and Algorithm 1, Algorithm 2 shows higher accuracy near the intersection point at $\eta = \pi/10$.

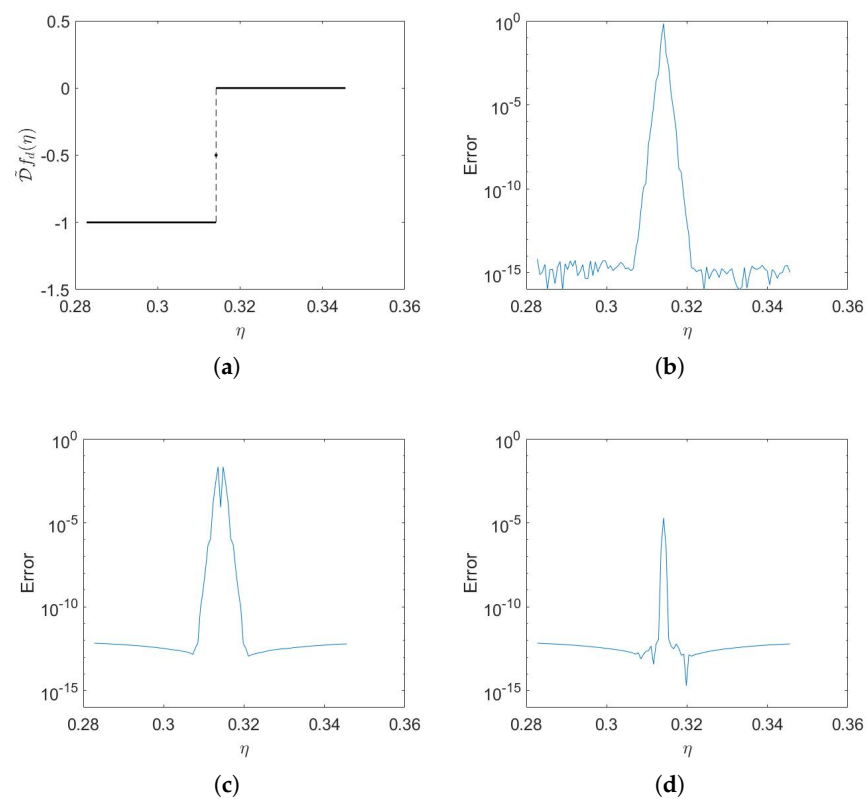


Figure 4. Double-layer potential evaluation at 10^4 equispaced points on a unit circle with the starfish boundary described by (15). (a) The graph of $\tilde{\mathcal{D}}f_d$ on $(0.09\pi, 0.11\pi)$. (b) The 10^4 -point PTR. (c) Algorithm 1 with $n = 10^4$, $\beta = 1.54 \times 10^{-8}$, $M = 312$. (d) Algorithm 2 with $m = 55$, $n = 10^4$, $\beta = 1.54 \times 10^{-8}$, $M = 312$, $\sigma_0 = 10^5$.

The second example presents the results with a limaçon Γ_1 in Figure 3b. The unit circle is inside the domain Ω except for a small segment near $(1,0)$. Note that the parts of limaçon near $(1,0)$ are very close to each other, and some of the targets are inside the “slit”. Algorithms 1 and 2 use the same parameter settings as the first example. The graph of $\tilde{\mathcal{D}}f_d$ on $(-\pi/10, \pi/10)$ is shown in Figure 5a. This function is piecewise, with only a small region centered at 0 with a width of $2 \arccos(9/(10 - 0.999)) \approx 0.03$ being 0, while the remainder is -1 almost everywhere. Figure 5b,c show that the errors of PTR and Algorithm 1 are significant at the discontinuity points and are also not negligible in the central region. In comparison, Algorithm 2 computes accurately in the central region, though the error is still significant at the discontinuity points, as shown in Figure 5d. The computational results of this method can clearly distinguish the central region, with a width of only 0.03.

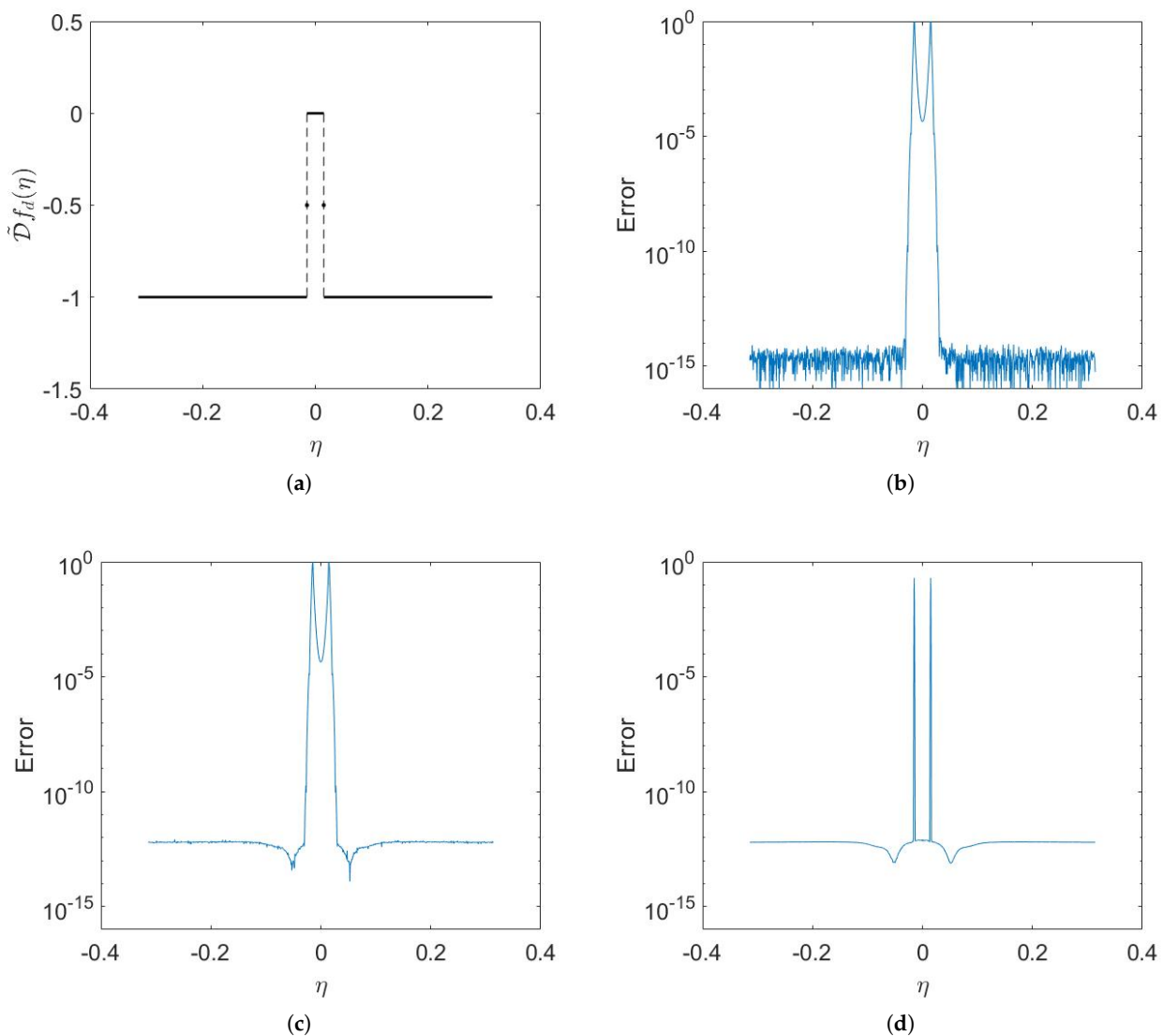


Figure 5. Double-layer potentials evaluation at 10^4 equispaced points on a unit circle with the limaçon boundary described by (16). (a) The graph of $\tilde{\mathcal{D}}f_d$ on $(-\pi/10, \pi/10)$. (b) A 10^4 -point PTR. (c) Algorithm 1 with $n = 10^4, \beta = 1.54 \times 10^{-8}, M = 312$. (d) Algorithm 2 with $m = 55, n = 10^4, \beta = 1.54 \times 10^{-8}, M = 312, \sigma_0 = 10^5$.

We next show the evaluation errors of the proposed methods on a circle of radius R close to the starfish boundary described by (15). Here, we chose $\beta = (R - r_{\max})^2 / (50Rr_{\max})$ in Algorithms 1 and 2, with the selection of other parameters following the same rules as in the first example. Figure 6a shows two curves Γ_1 and Γ_2 in this example, which are at a distance of 10^{-4} from each other. Figure 6b,c illustrate that the results of PTR and Algorithm 1 are inaccurate at points near Γ_1 , whereas Algorithm 2 achieves a precision of 13 decimal places, as shown in Figure 6d. We denote the distance between Γ_1 and Γ_2 as $\text{dist}(\Gamma_1, \Gamma_2)$. Table 1 presents the ℓ^∞ errors on $\mathbf{T}_{1.3001, n}$ and the computation time for computing $\tilde{D}f_d$ on a circle with a radius of 1.3001 with different values of n . The errors of PTR and Algorithm 1 decay slowly, while the error of Algorithm 2 maintains a precision of 13 decimal places. The computing time for Algorithms 1 and 2 grows quasi-linearly as n increases, whereas the time of PTR grows quadratically.

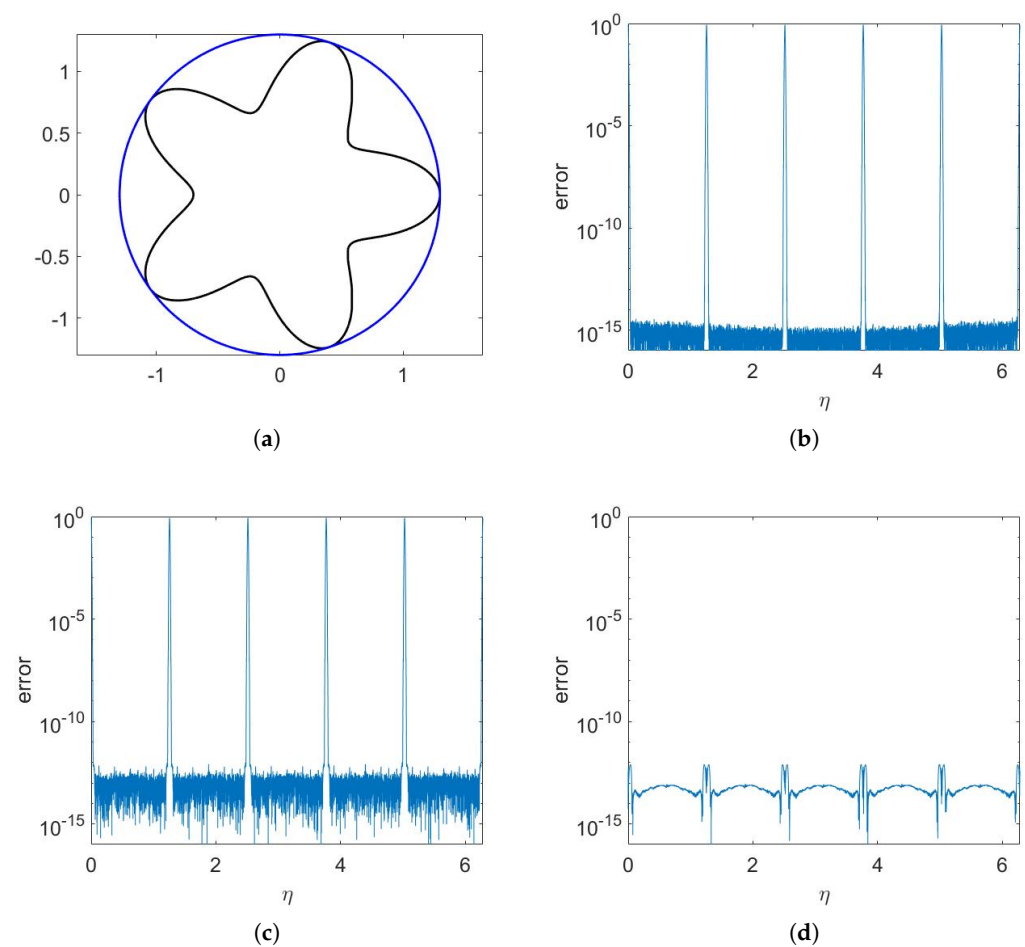


Figure 6. Double-layer potential evaluations at 10^4 equispaced points on a circle located at a distance of 10^{-4} from the starfish boundary described by (15). (a) A starfish curve (black) and a circle of radius 1.3001 (blue). (b) A 10^4 -point PTR. (c) Algorithm 1 with $n = 10^4$, $\beta = 1.18 \times 10^{-10}$, $M = 397$. (d) Algorithm 2 with $m = 55$, $n = 10^4$, $\beta = 1.18 \times 10^{-10}$, $M = 397$, $\sigma_0 = 10^5$.

Table 1. The ℓ^∞ errors of the Laplace double-layer potential evaluated on $\mathbf{T}_{1.3001,n}$ and the computation time, where n varies from 10^4 to 8×10^4 , and $\text{dist}(\Gamma_1, \Gamma_2) = 10^{-4}$.

n	ℓ^∞ Error			Time (s)		
	PTR	Algorithm 1	Algorithm 2	PTR	Algorithm 1	Algorithm 2
1×10^4	8.6×10^{-1}	8.6×10^{-1}	8.2×10^{-13}	7.5	0.6	1.5
2×10^4	2.7×10^{-1}	2.7×10^{-1}	8.4×10^{-13}	22.3	1.5	2.9
4×10^4	4.8×10^{-2}	4.8×10^{-2}	9.5×10^{-13}	75.2	2.8	5.7
8×10^4	2.1×10^{-3}	2.1×10^{-3}	9.3×10^{-13}	259.5	7.0	12.1

Table 2 shows the ℓ^∞ errors on $\mathbf{T}_{R,10^4}$ with $\text{dist}(\Gamma_1, \Gamma_2)$ varying from 10^{-1} to 10^{-4} , confirming that the proposed parameter selection for Algorithm 2 ensures the errors remain bounded regardless of $\text{dist}(\Gamma_1, \Gamma_2)$, as discussed in Section 2.2.

Table 2. The ℓ^∞ errors in computing the Laplace double-layer potentials on $\mathbf{T}_{R,10^4}$, where R varies from 1.4 to 1.3001, and $\text{dist}(\Gamma_1, \Gamma_2)$ varies from 10^{-1} to 10^{-4} .

$\text{dist}(\Gamma_1, \Gamma_2)$	ℓ^∞ Error		
	PTR	Algorithm 1	Algorithm 2
10^{-1}	3.0×10^{-15}	6.7×10^{-14}	6.7×10^{-14}
10^{-2}	6.5×10^{-15}	1.1×10^{-12}	1.1×10^{-12}
10^{-3}	4.7×10^{-4}	4.7×10^{-4}	6.6×10^{-13}
10^{-4}	8.6×10^{-1}	8.6×10^{-1}	8.2×10^{-13}

3.2. Integral Equation Solvers

In this section, we examine the proposed methods in the context of solving integral equations of the second kind. A numerical experiment of the exterior Neumann boundary value problems for the Laplace equation is conducted. Let $\Omega \subset \mathbb{R}^2$ be the bounded open domain, with boundary Γ_1 described by (15), and let $u_{ex}(\mathbf{x}) := (x_1 - 0.1)/|\mathbf{x} - (0.1, 0.4)|^2$ be the solution, where $\mathbf{x} \in \mathbb{R}^2 \setminus \Omega$. We use the single-layer potential representation $u_{ex} = \mathcal{S}\varphi$ [1]. Here, $\varphi \in C(\Gamma_1)$ satisfies

$$\left(-\frac{1}{2} + \mathcal{D}'\right)\varphi = \frac{\partial u_{ex}}{\partial \mathbf{v}},$$

where

$$(\mathcal{D}'\varphi)(\mathbf{x}) := \int_{\Gamma_1} \frac{\partial K(\mathbf{x}, \mathbf{y})}{\partial \mathbf{v}(\mathbf{x})} \varphi(\mathbf{y}) \, ds_{\mathbf{y}}, \quad \mathbf{x} \in \Gamma_1.$$

In order to obtain the potential density function f , we solve the BIEs using the Fourier Galerkin method described in [2]. The tested methods are employed to compute $\mathcal{S}\varphi$ on $\mathbb{T} \setminus \Omega$, where

$$\mathbb{T} := \{\mathbf{x} : \mathbf{x} \in \mathbf{T}_{R_k, 256}, R_k = 0.7 + 0.01k, k = 0, \dots, 90\},$$

The first column of Figure 7 shows the absolute error of the PTR with 256 points. The last two columns of Figure 7 illustrated the absolute error of Algorithms 1 and 2. Here, E_∞ in Figure 7 denotes the ℓ_∞ error of the evaluations on $\mathbb{T} \setminus \Omega$. Algorithms 1 and 2 are employed with $n = 256, \beta = \lambda^2/(50Rr_{\max}), \lambda = \max\{R - r_{\max}, 10^{-6}\}$, and $M = \lceil -28 \log_{10} \beta + \log_{10}^2 \beta \rceil$. In addition, we select $m = 55$ and $\sigma_0 = 100$ for Algorithm 2. Figure 7 illustrates that Algorithm 2 achieves a precision of nine digits in evaluating the single-layer potential, even for target points near the boundary with a distance of 1.4×10^{-5} . Conversely, both the PTR and Algorithm 1 exhibit inaccuracies close to the boundary.

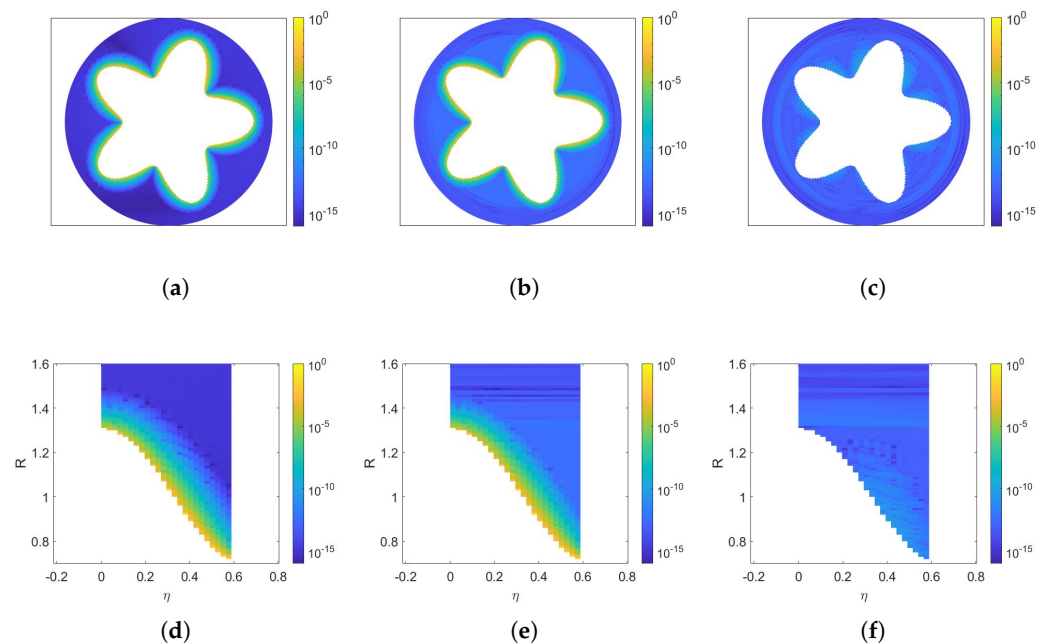


Figure 7. Absolute errors of computing single-layer potentials at targets on $\mathbb{T} \setminus \Omega$ with a minimum distance of 1.4×10^{-5} from the exterior Laplace Neumann BVPs. The first column shows the errors of the 256-point PTR. The last two columns present the errors of Algorithms 1 and 2 using the proposed parameter selection scheme with $n = 256$, $\lambda = \max\{R - r_{\max}, 10^{-6}\}$, $m = 55$, and $\sigma_0 = 100$. The bottom row shows the errors relative to the points parameterized by angle η and radius R . (a) PTR: $E_\infty = 6.4 \times 10^{-2}$. (b) Algorithm 1: $E_\infty = 6.4 \times 10^{-2}$. (c) Algorithm 2: $E_\infty = 1.4 \times 10^{-9}$. (d) PTR. (e) Algorithm 1. (f) Algorithm 2.

4. Conclusions

In this paper, we introduce a convolution sum approximation of layer potentials and provide an approximation error analysis. A parameter selection scheme for the approximation is proposed in Section 2.2 to ensure high approximating accuracy outside a sufficiently small neighborhood of the boundary. Then two fast methods using convolution sums are presented for the evaluation of layer potentials in this paper. Due to the nearly singular convolution terms, Algorithm 1 exhibits inaccuracies near the boundary similar to PTR, as shown in the experiments in Section 3. Algorithm 2, which still maintains a complexity of $\mathcal{O}(n \log n)$, improves the accuracy of the evaluation near the boundary via computing the nearly singular convolution terms by the Gauss quadrature.

The improved method offers several advantages: Firstly, this approach leverages periodic convolution sums to approximate layer potentials on a circle, enabling fast computation compared to traditional methods. This technique significantly reduces computational time. In the meantime, a local quadrature is applied to the nearly singular convolution terms in order to maintain the algorithm's accuracy near the boundary. Secondly, the algorithm allows for handling discontinuities in the field across the boundary, which is a feature that is not commonly addressed by existing methods. Lastly, our method is capable of handling scenarios where parts of the source geometry almost touch each other. This robustness is achieved without requiring much additional work, making the algorithm practical and efficient in real-world applications.

On the one hand, the algorithms proposed in this paper allow for fast off-boundary computation of layer potentials while accommodating discontinuities in the potential across the boundary. On the other hand, applications to practical problems when restricted to either the exterior or interior may result in redundant computations. Additionally, the assumption of a polar boundary limits the algorithm's applicability to practical scenarios. Nevertheless, this method still presents an opportunity to accelerate the evaluation of

integral operators with logarithmic or power-law singularities and to develop algorithms for higher dimensions.

Author Contributions: Conceptualization, W.G.; methodology, W.G.; software, W.G. and L.X.; validation, W.G. and L.X.; formal analysis, W.G.; investigation, W.G.; writing—original draft preparation, W.G.; writing—review and editing, W.G., Y.H. and L.X.; visualization, L.X. and Z.W.; project administration, W.G.; funding acquisition, W.G. and Y.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (No. 12101625), the National Key Research and Development Program of China (2023YFB3001704), the Natural Science Foundation of Guangdong Province, China (No. 2022A1515010831), the National Natural Science Foundation of China (grant No. 61976104), and in part by the 2022 Teaching Quality Project for Undergraduate Institutions of Guangdong Province.

Data Availability Statement: The datasets generated during the current study are available from the corresponding author upon reasonable request.

Acknowledgments: Discussions with Ying Jiang (Sun Yat-sen University) and the detailed comments of the anonymous referees are gratefully acknowledged.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A. Approximation by Exponential Sums

In this section, we introduce the exponential sums associated with logarithmic and inverse functions by using the technique in [27]. Following that, we provide corresponding error estimates.

Appendix A.1. Logarithmic Function

In this subsection, we consider using the exponential sum to approximate the logarithmic function $\phi(x) = -\frac{1}{4\pi} \log(x)$. This approach is based on the techniques described in [27]. The logarithmic function has the exponential integral form

$$\phi(x) = -\frac{1}{4\pi} \int_1^x \frac{1}{t} dt = -\frac{1}{4\pi} \int_1^x \int_0^\infty e^{-\rho t} d\rho dt = -\frac{1}{4\pi} \int_0^\infty \int_1^x e^{-\rho t} dt d\rho, \quad (\text{A1})$$

for $x \geq 1$. Since

$$\int_1^x e^{-\rho t} dt = \frac{1}{\rho} (e^{-\rho} - e^{-\rho x}), \quad (\text{A2})$$

substituting (A2) into the right-hand side of (A1) implies that

$$\phi(x) = \frac{1}{4\pi} \int_0^\infty \frac{1}{\rho} (e^{-\rho x} - e^{-\rho}) d\rho.$$

Applying a substitution of $\rho = \log(1 + e^\xi)$ yields

$$\phi(x) = \int_{-\infty}^{+\infty} f_1(\xi, x) d\xi, \quad x \geq 1, \quad (\text{A3})$$

where

$$f_1(\xi, x) := \frac{1}{4\pi} \frac{e^{-x \log(1+e^\xi)} - e^{-\log(1+e^\xi)}}{(1+e^{-\xi}) \log(1+e^\xi)}.$$

By substituting $\xi = \sinh(t)$ into (A3), the integrand changes into

$$f_2(t, x) := \cosh(t) f_1(\sinh(t), x).$$

We then use the truncated Sinc quadrature rules [29,30] for computing the integral as follow

$$T_{h,M}(f_2)(x) := h \sum_{\ell=-M}^M f_2(\ell h, x). \quad (\text{A4})$$

In the numerical examples of this paper, we truncate the sum (A4) and approximate the integral (A3) by

$$\begin{aligned} \tilde{T}_{h,M}(f_2)(x) &:= h \sum_{\ell=-M}^0 \tilde{f}_2(\ell h, x) - h \sum_{\ell=-M}^M \tilde{f}_2(\ell h, 1) \\ &= h \sum_{k=0}^M \tilde{f}_2(-kh, x) - b_M = \sum_{k=0}^M c_k e^{-a_k x} - b_M, \end{aligned} \quad (\text{A5})$$

where

$$\tilde{f}_2(t, x) := \frac{1}{4\pi} \frac{e^{-x \log(1+e^{\sinh(t)})} \cosh(t)}{(1+e^{-\sinh(t)}) \log(1+e^{\sinh(t)})}, \quad a_k := \log(1+e^{-\sinh(kh)}),$$

and

$$b_M := \sum_{k=-M}^M c_k e^{-a_k}, \quad c_k := \frac{h \cosh(kh)}{4\pi(1+e^{\sinh(kh)}) \log(1+e^{-\sinh(kh)})}.$$

Remark A1. There is precision loss in the evaluation of $\log(1+exp(-\sinh(kh)))$ in a_k, c_k when kh is large. In such cases, using its Taylor series for computation can provide better accuracy.

Appendix A.2. Function $1/x$

When $\phi(x) = -\frac{1}{2\pi x}$, we introduce the corresponding exponential sums in [2]. Consider the Laplace integral transform

$$\phi(x) = -\frac{1}{2\pi x} = -\frac{1}{2\pi} \int_0^{\infty} e^{-xt} dt.$$

By applying substitutions of $t = \log(1+e^\rho), \rho = \sinh(\xi)$, we have

$$\phi(x) = \int_{-\infty}^{+\infty} f_3(\xi, x) d\xi, \quad x \geq 1, \quad (\text{A6})$$

where

$$f_3(\xi, x) := \cosh(\xi) f_4(\sinh(\xi), x), \quad f_4(\rho, x) := -\frac{1}{2\pi} \frac{e^{-x \log(1+e^\rho)}}{1+e^{-\rho}}.$$

Use the truncated Sinc quadrature, we have

$$T_{h,M}(f_3)(x) := h \sum_{k=-M}^M f_3(kh, x), \quad (\text{A7})$$

and in this paper, we compute (A6) as

$$\tilde{T}_{h,M}(f_3)(x) := h \sum_{\ell=-M}^0 f_3(\ell h, x) = h \sum_{k=0}^M f_3(-kh, x) = \sum_{k=0}^M \tilde{c}_k e^{-a_k x}, \quad (\text{A8})$$

where

$$a_k = \log(1+e^{-\sinh(kh)}), \quad \tilde{c}_k = \frac{h \cosh(kh)}{1+e^{\sinh(kh)}}.$$

Remark A2. The function $\phi(x) = cx^{-\alpha}$ for $c, \alpha > 0$ can also be approximated by exponential sums (see [27]).

Appendix A.3. Error Estimates

In this subsection, we provide an error estimate for the M -term approximation of the exponential sums discussed in the preceding two subsections. To this end, we first estimate the truncated error between $T_{h,M}(f_k)$ and $\tilde{T}_{h,M}(f_k)$ for $k = 2, 3$, which represent the cases for $\phi(x) = \log(x)$ and $\phi(x) = x^{-1}$, respectively. We provide a lemma for estimating the error for the case of $\phi(x) = (-2\pi x)^{-1}$ for $x \geq s \geq 1$.

Lemma A1 (Function $(-2\pi x)^{-1}$). *There exists a positive constant C such that for all $M \in \mathbb{Z}^+$, $s \geq 1$ and $h > 0$,*

$$\left\| T_{h,M}(f_3) - \tilde{T}_{h,M}(f_3) \right\|_{\infty, [s, \infty)} \leq \left(2^{-s} - \left(1 + e^{\sinh(Mh)} \right)^{-s} \right) / (2\pi s) + Cs2^{-s}Mh^2.$$

Proof. Since

$$\left\| T_{h,M}(f_3) - \tilde{T}_{h,M}(f_3) \right\|_{\infty, [s, \infty)} = \left\| h \sum_{\ell=1}^M f_3(\ell h, \cdot) \right\|_{\infty, [s, \infty)}, \quad (\text{A9})$$

it suffices to estimate the upper bound of $\left\| h \sum_{\ell=1}^M f_3(\ell h, \cdot) \right\|_{\infty, [s, \infty)}$. According to the triangle inequality, we have

$$\left\| h \sum_{\ell=1}^M f_3(\ell h, \cdot) \right\|_{\infty, [s, \infty)} \leq \left\| h \sum_{\ell=1}^M f_3(\ell h, \cdot) - \int_0^{Mh} f_3(\xi, \cdot) d\xi \right\|_{\infty, [s, \infty)} + \left\| \int_0^{Mh} f_3(\xi, \cdot) d\xi \right\|_{\infty, [s, \infty)}. \quad (\text{A10})$$

The error estimate of the right rectangle rule implies that for all $x \in [s, \infty)$,

$$\left| h \sum_{\ell=1}^M f_3(\ell h, x) - \int_0^{Mh} f_3(\xi, x) d\xi \right| \leq \frac{Mh^2}{2} \max_{\xi \in [0, Mh]} \left| \frac{\partial f_3}{\partial \xi}(\xi, x) \right|. \quad (\text{A11})$$

Since it can be derived that there exists a positive constant C such that for all $s \geq 1$

$$\max_{(\xi, x) \in [0, Mh] \times [s, \infty)} \left| \frac{\partial f_3}{\partial \xi}(\xi, x) \right| \leq Cs2^{-(s-1)}, \quad (\text{A12})$$

it follows from (A11) and (A12) that

$$\left\| h \sum_{\ell=1}^M f_3(\ell h, \cdot) - \int_0^{Mh} f_3(\xi, \cdot) d\xi \right\|_{\infty, [s, \infty)} \leq Cs2^{-s}Mh^2. \quad (\text{A13})$$

Note that

$$\left\| \int_0^{Mh} f_3(\xi, \cdot) d\xi \right\|_{\infty, [s, \infty)} \leq \frac{1}{2\pi} \int_{\log 2}^{Mh} e^{-st} dt = \left(2^{-s} - \left(1 + e^{\sinh(Mh)} \right)^{-s} \right) / (2\pi s). \quad (\text{A14})$$

Combining (A9), (A10), (A13) and (A14) leads to the result. \square

Similarly, we can derive the error estimate for the case when $\phi(x) = \frac{1}{4\pi} \log(x)$ for $x \geq s$.

Lemma A2 (Logarithmic Function). *There exists a positive constant C such that for all $M \in \mathbb{Z}^+$, $s \geq 1$ and $h > 0$,*

$$\left\| T_{h,M}(f_2) - \tilde{T}_{h,M}(f_2) \right\|_{\infty, [s, \infty)} \leq \left(2^{-s} - \left(1 + e^{\sinh(Mh)} \right)^{-s} \right) / (4\pi s \log 2) + Cs2^{-s}Mh^2.$$

Proof. Note that

$$\|T_{h,M}(f_2) - \tilde{T}_{h,M}(f_2)\|_{\infty,[s,\infty)} = \left\| h \sum_{\ell=1}^M \tilde{f}_2(\ell h, \cdot) \right\|_{\infty,[s,\infty)}. \quad (\text{A15})$$

It suffices to bound the right-hand side of (A15). According to the triangle inequality, we have

$$\left\| h \sum_{\ell=1}^M \tilde{f}_2(\ell h, \cdot) \right\|_{\infty,[s,\infty)} \leq \left\| h \sum_{\ell=1}^M \tilde{f}_2(\ell h, \cdot) - \int_0^{Mh} \tilde{f}_2(\xi, \cdot) d\xi \right\|_{\infty,[s,\infty)} + \left\| \int_0^{Mh} \tilde{f}_2(\xi, \cdot) d\xi \right\|_{\infty,[s,\infty)}. \quad (\text{A16})$$

Similar to the proof of Lemma A1, it can be demonstrated that there exists a positive constant C such that for all $M \in \mathbb{Z}^+$, $s \geq 1$ and $h > 0$,

$$\left\| h \sum_{\ell=1}^M \tilde{f}_2(\ell h, \cdot) - \int_0^{Mh} \tilde{f}_2(\xi, \cdot) d\xi \right\|_{\infty,[s,\infty)} \leq Cs2^{-s}Mh^2. \quad (\text{A17})$$

From

$$\left\| \int_0^{Mh} \tilde{f}_2(\xi, \cdot) d\xi \right\|_{\infty,[s,\infty)} \leq \frac{1}{4\pi} \int_{\log 2}^{Mh} e^{-st} / t dt \leq \frac{1}{4\pi} \int_{\log 2}^{Mh} e^{-st} dt / \log 2$$

and (A14), it follows that

$$\left\| \int_0^{Mh} \tilde{f}_2(\xi, \cdot) d\xi \right\|_{\infty,[s,\infty)} \leq \left(2^{-s} - \left(1 + e^{\sinh(Mh)} \right)^{-s} \right) / (4\pi s \log 2). \quad (\text{A18})$$

Combining (A16)–(A18) leads to the result. \square

We restate an estimate for $\|T_{h,M}(f_3) - \phi\|_{\infty,[1,\iota]}$ with $\phi(x) = x^{-1}$ and $\iota > 1$, as presented in Lemma 4.3 in [27].

Lemma A3 (Hackbusch, Khoromskij). *Let $\phi(x) = (-2\pi x)^{-1}$. There exists a positive constant C such that for all $\iota > 1$ and $M \in \mathbb{Z}^+$, by choosing $h = \log(4\pi M) / M$,*

$$\|T_{h,M}(f_3) - \phi\|_{\infty,[1,\iota]} \leq C \exp\left(-\frac{\pi^2 M}{\log(3\iota) \log(\pi^2 M)}\right).$$

For convenience, denote

$$v(z) := \frac{e^z - 1 - z}{z},$$

on \mathbb{C} and introduce $a \lesssim b$ to mean $a \leq Cb$ for some positive constant C and $a, b \in \mathbb{R}$. We next present an estimate of $\|T_{h,M}(f_2) - \phi\|_{\infty,[1,\iota]}$ for $\phi(x) = -\frac{1}{2\pi} \log x$ and $\iota > 1$.

Lemma A4. *Let $\phi(x) = -\frac{1}{2\pi} \log x$. There exists a positive constant C such that for all $\iota > 1$ and $M \in \mathbb{Z}^+$, by choosing $h = \log(4\pi M) / M$,*

$$\|T_{h,M}(f_2) - \phi\|_{\infty,[1,\iota]} \leq C\iota \exp\left(-\frac{\pi^2 M}{\log(3\iota) \log(\pi^2 M)}\right). \quad (\text{A19})$$

Proof. Let $D_\delta := \{z \in \mathbb{C} : |\Im mz| \leq \delta\}$ with $\delta \leq 0.93 < \frac{\pi}{2}$. Since

$$f_2(z, x) = \frac{\cosh z(-x + 1 - xv(-x \log(1 + \exp(\sinh(z)))) + v(-\log(1 + \exp(\sinh(z))))}{4\pi(1 + \exp(\sinh(z)))},$$

v is analytic on \mathbb{C} , and the zeros of $1 + \exp(\sinh(z))$ are outside of D_δ , the function $f_2(\cdot, x)$ is analytic in D_δ , with $x \geq 1$.

To utilize Lemma 4.3 from [27] for the proof of this lemma, it is necessary to establish the following inequalities:

$$|f_2(z, x)| \leq \frac{1}{2\pi \log 2} \left(\left| \frac{\cosh(z)e^{-x \log(1+\exp(\sinh(z)))}}{1 + \exp(-\sinh(z))} \right| + \left| \frac{\cosh(z)e^{-\log(1+\exp(\sinh(z)))}}{1 + \exp(-\sinh(z))} \right| \right), \quad (\text{A20})$$

for all $z \in D_\delta$, $\Re z \geq 0$, $x \geq 1$, and

$$|f_2(z, x)| \leq \frac{3.33(x+1)}{4\pi} \left| \frac{\cosh(z)}{1 + \exp(-\sinh(z))} \right|, \quad (\text{A21})$$

for all $z \in D_\delta$, $\Re z < 0$, $x \geq 1$.

Firstly, we prove

$$|\log(1 + \exp(\sinh(z)))|^{-1} \leq (\log \sqrt{2})^{-1} \quad (\text{A22})$$

for all $z \in D_\delta$, $\Re z \geq 0$. In the case of $z \in D_\delta$, $\Re z > x_1(\delta) := \operatorname{arsinh}(1/\cos \delta)$, we have $\Re(\sinh(z)) = \sinh(\Re z) \cos(\Im z) > \cos(\Im z)/\cos \delta \geq 1$. It follows that

$$\begin{aligned} |\log(1 + \exp(\sinh(z)))| &\geq \Re(\log(1 + \exp(\sinh(z)))) \\ &\geq \log(1 + 2 \exp(\Re \sinh z) \cos(\Im \sinh z) + \exp(2\Re \sinh z))/2 \\ &\geq \log(\exp(\Re \sinh z) - 1) \geq \log(e - 1) \geq \log \sqrt{2}. \end{aligned}$$

Thus, the inequality (A22) holds. In the case of $z \in D_\delta$, $0 \leq \Re z \leq x_1(\delta)$, we have $\sinh(\Re z) \leq 1/\cos \delta$, which leads to

$$\Im \sinh z = \cosh(\Re z) \sin(\Im z) \leq \sqrt{1 + \cos^2 \delta} \tan \delta.$$

The restriction $0 < \delta \leq 0.93$ guarantees $|\Im \sinh z| < \pi/2$. Additionally, the inequalities $\Re z \geq 0$ and $|\Im z| < \delta < \pi/2$ imply

$$\Re(\sinh(z)) = \sinh(\Re z) \cos(\Im z) > 0.$$

Thus, it follows that

$$\begin{aligned} |\log(1 + \exp(\sinh(z)))| &\geq \log(1 + 2 \exp(\Re \sinh z) \cos(\Im \sinh z) + \exp(2\Re \sinh z))/2 \\ &> \log(1 + \exp(2\Re \sinh z))/2 > \log \sqrt{2}. \end{aligned}$$

The above inequality implies (A22). We finish the proof of (A22) for all $z \in D_\delta$, $\Re z \geq 0$. From (A22) and the triangle inequality, we obtain (A20). Applying Lemma 4.3 in [27] to (A20) leads to

$$|f_2(z, x)| \lesssim \exp\left(\Re z - \frac{\cos 0.93}{2} \exp(\Re z)\right) \quad (\text{A23})$$

for $z \in D_\delta$, $\Re z \geq 0$, and $x \in [1, \iota]$. This bound of $f_2(z, x)$ is independent of x and decays double-exponentially.

Secondly, we prove

$$\left| \frac{e^{-x \log(1+\exp(\sinh(z)))} - e^{-\log(1+\exp(\sinh(z)))}}{\log(1 + \exp(\sinh(z)))} \right| \leq 3.33(x+1) \quad (\text{A24})$$

for all $z \in D_\delta$, $\Re z < 0$, $x \geq 1$. In the case of $z \in D_\delta$, $x_0(x) < \Re z < 0$, $x \geq 1$, where x_0 is defined by (4.5) in [27], we have

$$\Re(-x \log(1 + \exp(\sinh(z)))) < 0 \quad (\text{A25})$$

according to proof (d) of Lemma 4.3 in [27]. In the case of $z \in D_\delta, \Re z \leq x_0(x), x \geq 1$, according to proof (e) of Lemma 4.3 in [27], we have

$$\Re(-x \log(1 + \exp(\sinh(z)))) \leq \frac{\log 3}{2}. \tag{A26}$$

We conclude that for all $z \in D_\delta, \Re z < 0, x \geq 1$, the inequality (A26) holds. Consider the upper bound of $|v|$ for $\Re z \leq \frac{\log 3}{2}$. We have $|v(z)| \leq \nu(R_0)$ for $|z| \leq R_0$. In the case of $|z| \geq R_0$ and $\Re z \leq \frac{\log 3}{2}$, it follows that $|v(z)| \leq (e^{\Re z} + 1)/R_0 + 1 \leq (e^{\frac{\log 3}{2}} + 1)/R_0 + 1$. We choose R_0 as the zero of $(e^{\frac{\log 3}{2}} + 1)/R_0 + 1 = \nu(R_0)$ and $|v(z)| \leq \nu(R_0) \leq 2.33$ for all $\Re(z) \leq \frac{\log 3}{2}$. Thus, the inequality (A26) implies that

$$|v(-x \log(1 + \exp(\sinh(z))))| \leq 2.33, \tag{A27}$$

for $z \in D_\delta, \Re z < 0, x \geq 1$. Combining (A27) and

$$\begin{aligned} & \left| \frac{e^{-x \log(1 + \exp(\sinh(z)))} - e^{-\log(1 + \exp(\sinh(z)))}}{\log(1 + \exp(\sinh(z)))} \right| \\ & \leq 1 + x + x|v(-x \log(1 + \exp(\sinh(z))))| + |v(-\log(1 + \exp(\sinh(z))))| \end{aligned}$$

leads to (A24) for $z \in D_\delta, \Re z < 0, x \geq 1$. Inequality (A24) implies (A21). According to (A21) and the bound of $\left| \frac{\cosh(z)}{1 + \exp(-\sinh(z))} \right|$ in proof (d)(e) of Lemma 4.3 in [27], it can be obtained that

$$|f_2(z, x)| \lesssim \iota \exp\left(|\Re z| - \frac{\cos 0.93}{2} \exp(|\Re z|)\right) \tag{A28}$$

for $z \in D_\delta, \Re z < 0$, and $x \in [1, \iota]$. By (A23) and (A28), we conclude that (A28) holds for $z \in D_\delta$ and $x \in [1, \iota]$. Therefore, according to Proposition 2.1 in [27], the choice $a = 1, b = 1/2, \delta = \delta(x)$ defined by (4.5) in [27] leads to (A19). \square

Appendix B. The Proofs of the Smoothing Error Estimate Theorems

In this appendix, we give the proofs of Theorems 1 and 2.

Proof of Theorem 1. Let $I_\lambda \subset I_{2\pi}$ satisfy $\{\mathbf{r}(\eta) : \eta \in I_\lambda\} = \Gamma_2 \setminus \Gamma_{1,\lambda}$. The definition (5) of d implies $d(\eta, \theta) \in [\lambda^2, (R + r_{\max})^2]$ for all $I_\lambda \times I_{2\pi}$. Thus, $d_{1,\beta}(\eta, \theta) = d(\eta, \theta)/d_{2,\beta}(\theta) \in [s, \iota]$. For the single-layer potential $\mathcal{A}f$, we have

$$\begin{aligned} \|\mathcal{A}f - \mathcal{S}_{M,\beta}f\|_{\infty,\lambda} &= \left\| \int_{I_{2\pi}} (\phi - \tilde{T}_{h,M}(f_2)) \circ d_{1,\beta}(\eta, \theta) \tilde{f}(\theta) d\theta \right\|_{\infty, I_\lambda} \\ &\leq \|\phi - \tilde{T}_{h,M}(f_2)\|_{\infty, [s, \iota]} \|\tilde{f}\|_1 = \|\phi - \tilde{T}_{h,M}(f_2)\|_{\infty, [s, \iota]} \|f\|_1, \end{aligned} \tag{A29}$$

where $\tilde{T}_{h,M}(f_2)$ is defined by (A5). Since

$$\|\phi - \tilde{T}_{h,M}(f_2)\|_{\infty, [s, \iota]} \leq \|\phi - T_{h,M}(f_2)\|_{\infty, [s, \iota]} + \|T_{h,M}(f_2) - \tilde{T}_{h,M}(f_2)\|_{\infty, [s, \iota]}, \tag{A30}$$

where $T_{h,M}(f_2)$ is defined by (A4), it remains to estimate $\|\phi - T_{h,M}(f_2)\|_{\infty, [s, \iota]}$ and $\|T_{h,M}(f_2) - \tilde{T}_{h,M}(f_2)\|_{\infty, [s, \iota]}$. Lemma A4 ensures that there exists a positive constant C_1 independent of M, h, s, ι such that

$$\|\phi - T_{h,M}(f_2)\|_{\infty, [s, \iota]} \leq \|\phi - T_{h,M}(f_2)\|_{\infty, [1, \iota]} \leq C_1 \iota \exp\left(-\frac{\pi^2 M}{\log(3\iota) \log(\pi^2 M)}\right) \tag{A31}$$

with $h = \log(4\pi M)/M$. Lemma A2 implies that there exists a positive constant C_2 independent of M, h, s, ι such that

$$\begin{aligned} & \left\| T_{h,M}(f_2) - \tilde{T}_{h,M}(f_2) \right\|_{\infty,[s,\iota]} \leq \left\| T_{h,M}(f_2) - \tilde{T}_{h,M}(f_2) \right\|_{\infty,[s,\infty]} \\ & \leq \left(2^{-s} - \left(1 + e^{\sinh(\log(4\pi M))} \right)^{-s} \right) / (4\pi s \log 2) + C_2 s 2^{-s} \log(4\pi M)^2 / M. \end{aligned} \quad (\text{A32})$$

Combining (A29)–(A32) leads to (12). \square

Proof of Theorem 2. Recall that $I_\lambda \subset I_{2\pi}$ satisfies $\{\mathbf{r}(\eta) : \eta \in I_\lambda\} = \Gamma_2 \setminus \Gamma_{1,\lambda}$. Then $d(\eta, \theta) \in [\lambda^2, (R + r_{\max})^2]$ for all $I_\lambda \times I_{2\pi}$, and $d_{1,\beta}(\eta, \theta) \in [s, \iota]$. For the double-layer potential $\mathcal{A}f$, employing the definition (A8) of $\tilde{T}_{h,M}(f_3)$ and the Cauchy–Schwarz inequality yields

$$\begin{aligned} \|\mathcal{A}f - \mathcal{D}_{M,\beta}f\|_{\infty,\lambda} & \leq \left\| \int_{I_{2\pi}} (\phi - \tilde{T}_{h,M}(f_3)) \circ d_{1,\beta}(\eta, \theta) \tilde{f}_1(\theta) d\theta \right\|_{\infty, I_\lambda} \\ & \quad + \left\| \int_{I_{2\pi}} (\phi - \tilde{T}_{h,M}(f_3)) \circ d_{1,\beta}(\eta, \theta) \tilde{f}_2(\eta, \theta) d\theta \right\|_{\infty, I_\lambda} \\ & \leq \left\| \phi - \tilde{T}_{h,M}(f_3) \right\|_{\infty,[s,\iota]} \frac{\max_{x \in I_{2\pi}} |\mathbf{r}'(x)|}{\beta} \left(\frac{1}{R} + \frac{1}{r_{\min}} \right) \|f\|_1, \end{aligned} \quad (\text{A33})$$

where $\tilde{f}_1 = \mathbf{r}' \cdot \mathbf{r} \tilde{f} / d_{2,\beta}$, $\tilde{f}_2(\eta, \theta) = \mathbf{r}'(\theta) \cdot \mathbf{c}_R(\eta) \tilde{f}(\theta) / d_{2,\beta}(\theta)$. Since

$$\left\| \phi - \tilde{T}_{h,M}(f_3) \right\|_{\infty,[s,\iota]} \leq \left\| \phi - T_{h,M}(f_3) \right\|_{\infty,[s,\iota]} + \left\| T_{h,M}(f_3) - \tilde{T}_{h,M}(f_3) \right\|_{\infty,[s,\iota]}, \quad (\text{A34})$$

where $T_{h,M}(f_3)$ is defined by (A7), it remains to estimate $\left\| \phi - T_{h,M}(f_3) \right\|_{\infty,[s,\iota]}$ and $\left\| T_{h,M}(f_3) - \tilde{T}_{h,M}(f_3) \right\|_{\infty,[s,\iota]}$. Lemma A3 ensures that there exists a positive constant C_1 independent of M, h, s, ι such that

$$\left\| \phi - T_{h,M}(f_3) \right\|_{\infty,[s,\iota]} \leq C_1 \exp\left(-\frac{\pi^2 M}{\log(3\iota) \log(\pi^2 M)}\right) \quad (\text{A35})$$

with $h = \log(4\pi M)/M$. Lemma A1 implies that there exists a positive constant C_2 independent of M, h, s, ι such that

$$\begin{aligned} & \left\| T_{h,M}(f_3) - \tilde{T}_{h,M}(f_3) \right\|_{\infty,[s,\iota]} \leq \left\| T_{h,M}(f_3) - \tilde{T}_{h,M}(f_3) \right\|_{\infty,[s,\infty]} \\ & \leq \left(2^{-s} - \left(1 + e^{\sinh(\log(4\pi M))} \right)^{-s} \right) / (2\pi s) + C_2 s 2^{-s} \log(4\pi M)^2 / M. \end{aligned} \quad (\text{A36})$$

Combining (A33)–(A36) leads to (13). \square

References

1. Kress, R. *Linear Integral Equations*, 3rd ed.; Springer: New York, NY, USA, 2014.
2. Atkinson, K.E. *The Numerical Solution of Integral Equations of the Second Kind*, 1st ed.; Cambridge University Press: Cambridge, UK, 1997.
3. Kress, R.; Colton, D. *Inverse Acoustic and Electromagnetic Scattering Theory*, 4th ed.; Springer: Cham, Switzerland, 2019.
4. Jiang, Y.; Wang, B.; Xu, Y. A fast Fourier Galerkin method solving a boundary integral equation for the biharmonic equation. *SIAM J. Numer. Anal.* **2014**, *52*, 2530–2554. [[CrossRef](#)]
5. Yang, K.; Feng, W.; Gao, X. A new approach for computing hyper-singular interface stresses in IIBEM for solving multi-medium elasticity problems. *Comput. Methods Appl. Mech. Eng.* **2015**, *287*, 54–68. [[CrossRef](#)]
6. Wei, X.; Jiang, S.; Klöckner, A.; Wang, X. An integral equation method for the Cahn–Hilliard equation in the wetting problem. *J. Comput. Phys.* **2020**, *419*, 109521. [[CrossRef](#)]

7. Zhang, J.; Wei, S.; Yue, P.; Kulik, A.S.; Li, G. Surface pressure calculation method of multi-field coupling mechanism under the action of flow field. *Symmetry* **2023**, *15*, 1064. [[CrossRef](#)]
8. af Klinteberg, L.; Askham, T.; Kropinski, M.C. A fast integral equation method for the two dimensional Navier-Stokes equations. *J. Comput. Phys.* **2020**, *409*, 109353. [[CrossRef](#)]
9. Kločkner, A.; Barnett, A.H.; Greengard, L.; O’Neil, M. Quadrature by expansion: A new method for the evaluation of layer potentials. *J. Comput. Phys.* **2013**, *252*, 332–349. [[CrossRef](#)]
10. Barnett, A.H. Evaluation of layer potentials close to the boundary for Laplace and Helmholtz problems on analytic planar domains. *SIAM J. Sci. Comput.* **2014**, *36*, 427–451. [[CrossRef](#)]
11. af Klinteberg, L.; Tornberg, A.K. Adaptive quadrature by expansion for layer potential evaluation in two dimensions. *SIAM J. Sci. Comput.* **2018**, *40*, A1225–A1249. [[CrossRef](#)]
12. Ding, L.; Huang, J.; Marzuola, J.L.; Tang, Z. Quadrature by two expansions: Evaluating Laplace layer potentials using complex polynomial and plane wave expansions. *J. Comput. Phys.* **2021**, *428*, 109963. [[CrossRef](#)]
13. Carvalho, C.; Khatri, S.; Kim, A.D. Asymptotic analysis for close evaluation of layer potentials. *J. Comput. Phys.* **2018**, *355*, 327–341. [[CrossRef](#)]
14. Carvalho, C.; Khatri, S.; Kim, A.D. Asymptotic approximation for the close evaluation of double-layer potentials. *SIAM J. Sci. Comput.* **2020**, *42*, A504–A533. [[CrossRef](#)]
15. Khatri, S.; Kim, A.D.; Cortez, R.; Carvalho, C. Close evaluation of layer potentials in three dimensions. *J. Comput. Phys.* **2020**, *423*, 109798. [[CrossRef](#)]
16. Helsing, J.; Ojala, R. On the evaluation of layer potentials close to their sources. *J. Comput. Phys.* **2008**, *227*, 2899–2921. [[CrossRef](#)]
17. Barnett, A.H.; Wu, B.; Veerapaneni, S. Spectrally-accurate quadratures for evaluation of layer potentials close to the boundary for the 2d Stokes and Laplace equations. *SIAM J. Sci. Comput.* **2015**, *37*, B519–B542. [[CrossRef](#)]
18. af Klinteberg, L.; Barnett, A.H. Accurate quadrature of nearly singular line integrals in two and three dimensions by singularity swapping. *BIT Numer. Math.* **2021**, *61*, 83–118. [[CrossRef](#)]
19. Bao, G.; Hua, W.; Lai, J.; Zhang, J. Singularity swapping method for nearly singular integrals based on trapezoidal rule. *SIAM J. Numer. Anal.* **2024**, *62*, 974–997. [[CrossRef](#)]
20. Beale, J.T.; Lai, M. A method for computing nearly singular integrals. *SIAM J. Numer. Anal.* **2001**, *38*, 1902–1925. [[CrossRef](#)]
21. Tlupova, S.; Beale, J.T. Regularized single and double layer integrals in 3D Stokes flow. *J. Comput. Phys.* **2019**, *386*, 568–584. [[CrossRef](#)]
22. Zhu, H.; Veerapaneni, S. High-order close evaluation of Laplace layer potentials: A differential geometric approach. *J. Comput. Phys.* **2022**, *44*, A1381–A1404. [[CrossRef](#)]
23. Hale, N.; Trefethen, L.N. New quadrature formulas from conformal maps. *SIAM J. Numer. Anal.* **2008**, *46*, 930–948. [[CrossRef](#)]
24. Hale, N.; Tee, T.W. Conformal maps to multiply slit domains and applications. *SIAM J. Sci. Comput.* **2009**, *31*, 3195–3215. [[CrossRef](#)]
25. Rachh, M.; Kločkner, A.; O’Neil, M. Fast algorithms for Quadrature by Expansion I: Globally valid expansions. *J. Comput. Phys.* **2017**, *345*, 706–731. [[CrossRef](#)]
26. Wala, M.; Kločkner, A. A fast algorithm for Quadrature by Expansion in three dimensions. *J. Comput. Phys.* **2019**, *388*, 655–689. [[CrossRef](#)]
27. Hackbusch, W.; Khoromskij, B.N. Low-rank Kronecker-product approximation to multi-dimensional nonlocal operators. Part I. Separable approximation of multi-variate functions. *Computing* **2006**, *76*, 177–202. [[CrossRef](#)]
28. Tornberg, A.K.; Greengard, L. A fast multipole method for the three-dimensional Stokes equations. *J. Comput. Phys.* **2008**, *227*, 1613–1619. [[CrossRef](#)]
29. Trefethen, L.N.; Weideman, J.A.C. The exponentially convergent trapezoidal rule. *SIAM Rev.* **2014**, *56*, 385–458. [[CrossRef](#)]
30. Stenger, F. *Numerical Methods Based on Sinc and Analytic Functions*, 1st ed.; Springer: New York, NY, USA, 1993.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.