In the present paper we describe a class of algorithms for the solution of Laplace's equation on polygonal domains with Dirichlet boundary conditions. It is well known that in such cases the solutions have singularities near the corners, which poses a challenge for many existing methods. We present a high-order solver for Laplace's equation on polygonal domains requiring relatively few degrees of freedom to resolve the behaviour near corners accurately. Our approach is based on the observation that if the boundary data is smooth on each edge of the polygon, then in the vicinity of each corner the solution to the corresponding boundary integral equation has an expansion in terms of certain (analytically available) singular powers. We construct a set of discretization nodes and weights which accurately integrate and interpolate these singular powers, enabling the construction of high-order Nyström schemes requiring relatively few discretization nodes. Our results are illustrated with several numerical examples.

On the Numerical Solution of Elliptic Partial Differential Equations on Polygonal Domains

Jeremy Hoskins, Vladimir Rokhlin[◇], Kirill Serkh[‡]

Technical Report YALEU/DCS/TR-1538

‡ This author was supported in part by the NSF Mathematical Sciences Postdoctoral Research Fellowship (award no. 1606262) and AFOSR FA9550-16-1-0175
◇ This author was supported in part by AFOSR FA9550-16-1-0175 and by the Office of Naval Research (award no. N00014-14-1-0797/16-1-2123)

Keywords: *Boundary Value Problems, Potential Theory, Corners, Singular Solutions, Integral Equations, Partial Differential Equations, Elliptic Equations*

1 Introduction

In classical potential theory, elliptic partial differential equations (PDEs) are reduced to second kind boundary integral equations by representing the solutions to the differential equations by single-layer or double-layer potentials on the boundaries of the regions. After discretization, the resulting linear systems are generally better-conditioned than direct discretizations of the differential equation. For regions with smooth boundaries there exists a variety of methods, both direct and iterative, for solving these linear systems quickly and with high precision (see [10], for example).

However, near corners the solutions to both the differential and integral equations have singularities; these singularities pose significant challenges to many existing approaches. Generally, the singular behaviour is resolved by adding a large number of degrees of freedom near the corners (i.e. nested discretizations) and then compressing the resulting discretized systems. In this paper we present a high-order solver for Laplace's equation on polygonal domains which completely avoids brute force discretization (see for example [2], [3], [5], [7], [11], and [12] for alternative approaches).

The behaviour of solutions to Laplace's equation in the vicinity of corners has been the subject of extensive analysis (see [9], [14], [17], and [24] for representative examples). In particular, it is well-known that solutions are unique and exist in the L^2-sense (see [6], [22]) both for the differential and integral equations. Moreover, there has been considerable work detailing both the spaces to which solutions belong, as well as the leading-order terms of the singularity near the corner. Recently, it was shown in [20] that if the boundary data on either side of the corners is smooth then, near the corners, the solution to the boundary integral equation has an expansion in terms of certain non-integer powers and non-integer powers multiplied by logarithms. Moreover, the result is constructive: explicit formulas exist for the singular powers, depending on the angle of the corner, and a formula is given for the mapping which takes the coefficients of the Taylor series of the boundary data to the coefficients of the singular terms.

Our approach in this paper is based on the analysis in [20]. Specifically, we construct a set of quadrature nodes and weights which accurately integrate and interpolate these analytically available singular powers, enabling the implementation of a high-order Nyström scheme requiring relatively few discretization nodes.

The structure of the paper is as follows. In Section 2 we describe the necessary mathematical preliminaries. In Section 3 we present the necessary numerical preliminaries. Section 4 contains the numerical apparatus. Section 5 contains numerical results illustrating the performance of the algorithm.

2 Mathematical Preliminaries

2.1 Boundary Value Problems

Let Ω be the interior of a polygonal domain in $\mathbb{R}^2$ and $\gamma : [0, L] \to \mathbb{R}^2$ a counterclockwise arc length parametrization of its boundary. Let $\nu(t)$ be the inward-pointing normal to γ at $t \in [0, L]$, and let Γ denote the boundary of Ω (Figure 1). For boundary data $g : [0, L] \to \mathbb{R}$ we consider the following problems.

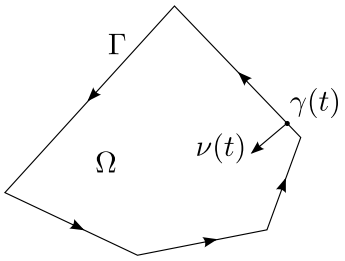

Figure 1: A polygonal curve in $\mathbb{R}^2$.

Interior Dirichlet problem:

$$\begin{aligned} &\nabla^2\phi(x) = 0 && x \in \Omega, \\ &\lim_{\substack{x\to\gamma(t)\\ x\in\Omega}} \phi(x) = g(t) && t \in [0, L]. \end{aligned} \tag{1}$$

Exterior Dirichlet problem:

$$\begin{aligned} &\nabla^2\phi(x) = 0 && x \in \mathbb{R}^2 \setminus \overline{\Omega}, \\ &\lim_{\substack{x\to\gamma(t)\\ x\in\mathbb{R}^2\setminus\overline{\Omega}}} \phi(x) = g(t) && t \in [0, L]. \end{aligned} \tag{2}$$

The interior Dirichlet problem has a unique solution for all $g \in L^2([0, L])$, (see [22]). For the exterior Dirichlet problem, we impose the additional condition that ϕ is bounded as x goes to infinity, in which case a unique solution exists for all $g \in L^2([0, L])$.

Remark 2.1. *For ease of exposition we restrict our discussion to regions with polygonal boundaries. The analysis and techniques of this paper easily extend to multiply connected domains.*

2.2 Integral equations of potential theory

In classical potential theory, boundary value problems are solved by representing the solution of the differential equation inside the region as a potential induced by charges and dipoles on the boundary. Let $\psi^0_{x_0}(x)$ denote the potential of a unit charge at $x_0 \in \mathbb{R}^2$ and let $\psi^1_{x_0,h}(x)$ denote the potential of a unit dipole at $x_0 \in \mathbb{R}^2$ oriented in the direction h. Specifically, $\psi^0_{x_0}, \psi^1_{x_0,h} : \mathbb{R}^2 \setminus x_0 \to \mathbb{R}$ are given by the following formulas

$$\psi^0_{x_0}(x) = \log(\|x - x_0\|), \tag{3}$$

$$\psi^1_{x_0,h}(x) = \frac{\langle h, x_0 - x\rangle}{\|x_0 - x\|^2}. \tag{4}$$

where $\|\cdot\|$ denotes the standard Euclidean distance and $\langle\cdot,\cdot\rangle$ denotes the inner product.

The potential due to a charge distribution ρ on the boundary Γ is normally referred to as a single-layer potential and is given by

$$\phi(x) = \int_0^L \psi^0_{\gamma(t)}(x)\rho(t)\,\mathrm{d}t, \tag{5}$$

for any $x \in \mathbb{R}^2 \backslash \Gamma$. Similarly, the potential due to a dipole distribution ρ on the boundary, is referred to as a double-layer potential and is given by

$$\phi(x) = \int_0^L \psi^1_{\gamma(t),\nu(t)}(x)\rho(t)\,\mathrm{d}t, \tag{6}$$

for any $x \in \mathbb{R}^2 \setminus \Gamma$.

2.3 Reduction of boundary value problems to boundary integral equations

The reduction of the boundary value problems in Section 2.1 to boundary integral equations is given by the following theorems.

Theorem 2.1. *Let $\rho \in L^2([0,L])$ and define $g : [0,L] \to \mathbb{R}$ by the following formula*

$$g(s) = -\pi\rho(s) + \int_0^L \psi^1_{\gamma(t),\nu(t)}(\gamma(s))\,\rho(t)\,\mathrm{d}t, \tag{7}$$

for all $s \in [0,L]$. If $g \in L^2([0,L])$ then (7) has a unique solution for $\rho \in L^2([0,L])$. Furthermore, the solution to the interior Dirichlet problem with boundary data g is given by (6).

Theorem 2.2. *Let $\rho \in L^2([0,L])$ and define $g : [0,L] \to \mathbb{R}$ by the following formula*

$$g(s) = \pi\rho(s) + \int_0^L \psi^1_{\gamma(t),\nu(t)}(\gamma(s))\,\rho(t)\,\mathrm{d}t, \tag{8}$$

for all $s \in [0,L]$. If $g \in L^2([0,L])$ such that

$$\int_0^L g(t)\,\mathrm{d}t = 0, \tag{9}$$

then (8) has a unique solution for $\rho \in L^2([0,L])$. Furthermore, the solution to the exterior Dirichlet problem with boundary data g is given by (6).

The following corollary follows immediately from Theorem 2.2.

Corollary 2.3. *Let $\rho \in L^2([0,L])$ and define $g : [0,L] \to \mathbb{R}$ by the following formula*

$$g(s) = \pi\rho(s) + \int_0^L \left(c + \psi^1_{\gamma(t),\nu(t)}(\gamma(s))\right)\,\rho(t)\,\mathrm{d}t, \tag{10}$$

for all $s \in [0,L]$, and an arbitrary constant c. If $g \in L^2([0,L])$ then (10) has a unique solution for $\rho \in L^2([0,L])$. Furthermore, the solution to the exterior Dirichlet problem with boundary data g is given by the following formula

$$\phi(x) = \int_0^L \left(c + \psi^1_{\gamma(t),\nu(t)}(x)\right)\rho(t)\,\mathrm{d}t. \tag{11}$$

2.4 Properties of the double-layer potential

The following lemma establishes the regularity of the function $\psi^1_{x_0,h}(x)$ when $x_0, x \in \Gamma$ and h is normal to Γ. It can be found in [1], for example.

Lemma 2.4. *Let $\gamma : [0, L] \to \mathbb{R}^2$ be a curve parametrized by arclength and $\nu(t)$ be the normal vector to $\gamma(t) = (\gamma_1(t), \gamma_2(t))$, $0 < t < L$, satisfying*

$$\nu(t) = (-\gamma_2'(t), \gamma_1'(t)). \tag{12}$$

Suppose that for some integer $k \geq 2$, γ is C^k in a neighbourhood of a point s, $0 < s < L$. Then

$$\psi^1_{\gamma(s),\nu(s)}(\gamma(t)), \tag{13}$$

$$\psi^1_{\gamma(t),\nu(t)}(\gamma(s)), \tag{14}$$

are C^{k-2} functions of t in a neighborhood of s and

$$\lim_{t\to s} \psi^1_{\gamma(s),\nu(s)}(\gamma(t)) = \lim_{t\to s} \psi^1_{\gamma(s),\nu(s)}(\gamma(t)) = -\frac{1}{2}\kappa(s), \tag{15}$$

where $\kappa : [0, L] \to \mathbb{R}$ is the signed curvature of γ. Similarly, if γ is analytic in a neighborhood of a point s, where $0 < s < L$, then (13) and (14) are analytic functions of t in a neighborhood of s.

The following lemma describes the behaviour of ψ^1 in the vicinity of a corner.

Lemma 2.5. *Under the same assumptions as the previous lemma, if γ has a corner with interior angle $\pi\alpha$ at t_0 then*

$$\lim_{\substack{s\to t_0^+ \\ t\to t_0^-}} \psi^1_{\gamma(s),\nu(s)}(\gamma(t)), \tag{16}$$

$$\lim_{\substack{s\to t_0^+ \\ t\to t_0^-}} \psi^1_{\gamma(t),\nu(t)}(\gamma(s)), \tag{17}$$

do not exist. In particular, along $s - t_0 = -(t - t_0) = h$,

$$\psi^1_{\gamma(t_0+h),\nu(t_0+h)}(\gamma(t_0 - h)) = \Big(\frac{\cos\pi\alpha}{2\sin\pi\alpha}\Big)\frac{1}{h} + O(1) \tag{18}$$

if γ is smooth in a neighbourhood of the corner.

We conclude this section with the following definition which will be used in the remainder of the paper.

Definition 2.1. *For a given boundary Γ, we define the kernel, $K : \Gamma \times \Gamma \to \mathbb{R}$ by the formula*

$$K(x, y) = \psi^1_{y,\nu(y)}(x). \tag{19}$$

Here, by a slight abuse of notation, we denote the normal derivative to Γ at a point $y \in \mathbb{R}^2$ by $\nu(y)$, instead of $\nu(\gamma^{-1}(y))$. For the exterior Dirichlet problem it is convenient to introduce a modified kernel, $\tilde{K} : \Gamma \times \Gamma \to \mathbb{R}$ defined by

$$\tilde{K}(x, y) = c + \psi^1_{y,\nu(y)}(x), \tag{20}$$

where c is an arbitrary constant. We note that both K and $\tilde{K}$ are smooth away from the corners of Γ.

2.5 Representation of solutions in the vicinity of corners

The following lemma gives an explicit formula for the kernel when Γ is an open wedge with interior angle $\pi\alpha$ and sides of unit length (Figure 2). In order to simplify the formulas, the arclength parametrization of Γ, denoted by γ, is defined for $t \in [-1, 1]$ rather than $[0, 2]$.

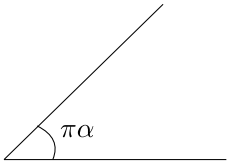

Figure 2: A wedge in $\mathbb{R}^2$.

Lemma 2.6. *Let Γ be an open wedge of side lengths one and interior angle $\pi\alpha$ with $0 < \alpha < 2$. Let $\gamma : [-1, 1] \to \Gamma$ be an arc length parametrization of Γ and $\nu : [-1, 1] \to \mathbb{R}^2$ be the inward-pointing normal to Γ. Then*

$$\psi_{\gamma(t),\nu(t)}(\gamma(s)) = \frac{s \sin \pi\alpha}{t^2 + s^2 + 2st \cos \pi\alpha}, \tag{21}$$

if $s < 0, t > 0$ or $s > 0, t < 0$. For all other values of s and t, $\psi_{\gamma(t),\nu(t)}(\gamma(s)) = 0$.

The following theorem gives an explicit representation of solutions near corners in the case where Γ is the wedge defined in the previous lemma.

Theorem 2.7 ([20])**.** *Suppose that $0 < \alpha < 2$ and that N is a positive integer. Let $\lceil \cdot \rceil$ and $\lfloor \cdot \rfloor$ denote the ceiling and floor functions, respectively, and define $\overline{L}$, $\underline{L}$, $\overline{M}$, and $\underline{M}$ by the following formulas*

$$\overline{L} = \left\lceil \frac{\alpha N}{2} \right\rceil, \tag{22}$$

$$\underline{L} = \left\lfloor \frac{\alpha N}{2} \right\rfloor, \tag{23}$$

$$\overline{M} = \left\lceil \frac{(2-\alpha) N}{2} \right\rceil, \tag{24}$$

$$\underline{M} = \left\lfloor \frac{(2-\alpha) N}{2} \right\rfloor. \tag{25}$$

Suppose further that ρ is defined via the formula

$$\begin{aligned} \rho(t) =& b_0 + \sum_{i=1}^{\overline{L}} b_i |t|^{\frac{2i-1}{\alpha}} + \sum_{i=1}^{\underline{M}} b_{\underline{L}+i} |t|^{\frac{2i}{2-\alpha}} (\log |t|)^{\sigma_{N,\alpha}(i)} \\ &+ \sum_{i=1}^{\overline{M}} c_i \operatorname{sgn}(t) |t|^{\frac{2i-1}{2-\alpha}} + \sum_{i=1}^{\underline{L}} c_{\overline{M}+i} \operatorname{sgn}(t) |t|^{\frac{2i}{\alpha}} (\log |t|)^{\sigma_{N,(2-\alpha)}(i)} \end{aligned} \tag{26}$$

where $b_0, b_1, \ldots, b_N$ and $c_1, c_2, \ldots, c_N$ are arbitrary real numbers and the function $\sigma_{\alpha,N}(i)$ is defined as follows

$$\sigma_{N,\alpha}(i) = \begin{cases} 1 & \text{if } \frac{2i}{2-\alpha} = \frac{2j-1}{\alpha} \text{ for some } j \in \mathbb{Z},\ 1 \leq j \leq \left\lceil \frac{\alpha N}{2} \right\rceil \\ 0 & \text{otherwise.} \end{cases} \tag{27}$$

If g is defined by

$$g(t) = \pi\rho(s) + \int_{-1}^{1} \psi^1_{\gamma(t),\nu(t)}(\gamma(s))\rho(t)\,\mathrm{d}t. \tag{28}$$

then there exist sequences of real numbers β_0, β_1, ... and γ_0, γ_1, ... such that

$$g(t) = \sum_{n=0}^{\infty} \beta_n |t|^n + \sum_{n=0}^{\infty} \gamma_n \operatorname{sgn}(t)|t|^n, \tag{29}$$

for all $-1 \leq t \leq 1$. Conversely, suppose that g has the form (29). Suppose further that N is an arbitrary positive integer. Then, for all angles $\pi\alpha$ there exist unique real numbers b_0, b_1, ..., b_N and c_0, c_1, ..., c_N such that ρ, defined by (26), solves equation (28) to within an error $\mathrm{O}(t^{N+1})$.

Remark 2.2. *A similar result holds for the case where the factor of π in (28) is replaced by $-\pi$; the change in sign corresponds to replacing the boundary integral equation for the exterior Dirichlet problem (8) with the boundary integral equation corresponding to interior Dirichlet problem (7).*

The following corollary characterizes of the behaviour of the solutions to (7) and (10) in the vicinity of a corner.

Corollary 2.8. *Let Γ be the boundary of a polygonal region and suppose one of its corners has interior angle $\pi\alpha$ where $\alpha \in (0, 2)$. Let $\psi : (-\delta, \delta) \to \mathbb{R}^2$ be an arclength parametrization of Γ in the vicinity of the corner, with $\psi(0)$ coinciding with the corner. If the boundary data, g, is analytic on either side of the corner on an interval of length $R > \delta > 0$, then there exist unique real numbers b_0, b_1, ..., b_N and c_0, c_1, ..., c_N such that the density, ρ, defined by (26) satisfies equation (7) to within an error $\mathrm{O}(t^{N+1})$ for t within δ of the corner.*

2.6 Quadratures

In this section we introduce terminology and definitions related to quadratures to be used in this paper.

Definition 2.2. *Suppose that $f_1, \ldots, f_m$ is a sequence of square-integrable functions defined on the interval $[a, b]$. An n-point quadrature rule is a sequence of n distinct points $a \leq x_1 < \cdots < x_n \leq b$, called nodes, and a sequence of n real numbers $w_1, \ldots, w_n$, called weights. We say that the quadrature rule $x_1, \ldots, x_n, w_1, \ldots, w_n$ integrates the collection of functions exactly if*

$$\int_a^b f_i(x)\,\mathrm{d}x = \sum_{j=1}^{n} f_i(x_j) w_j, \qquad i = 1, \ldots, m, \tag{30}$$

and with precision $\epsilon > 0$ *if*

$$\left| \int_a^b f_i(x)\,\mathrm{d}x - \sum_{j=1}^{n} f_i(x_j) w_j \right| < \epsilon, \quad i = 1, \ldots, m. \tag{31}$$

Three classes of quadrature formulas, called generalized Chebyshev quadratures, generalized Gaussian quadratures, and inner product quadratures, are defined as follows.

Definition 2.3. *A quadrature formula will be referred to as a* generalized Chebyshev quadrature *with respect to a set of* $2n$ *functions* $f_1, \ldots, f_{2n} : [a, b] \to \mathbb{R}$ *and a weight function* $\omega : [a, b] \to \mathbb{R}^+$ *if it consists of* $2n$ *weights and nodes and integrate the functions* f_i *exactly with the weight function* ω *for all* $i = 1, \ldots, 2n$.

Definition 2.4. *A quadrature formula will be referred to as a* generalized Gaussian quadrature *with respect to a set of* $2n$ *functions* $f_1, \ldots, f_{2n} : [a, b] \to \mathbb{R}$ *and a weight function* $\omega : [a, b] \to \mathbb{R}^+$ *if it consists of* n *nodes and* n *positive weights, and integrates the functions* f_i *with the weight function* ω *exactly for all* $i = 1, \ldots, 2n$.

Definition 2.5. *A quadrature formula will be referred to as an* inner-product quadrature *with respect to a set of* k *functions* $f_1, \ldots, f_k : [a, b] \to \mathbb{R}$ *and a weight function* $\omega : [a, b] \to \mathbb{R}^+$ *if it consists of* n *nodes and* n *weights, and integrates the* $k(k+1)/2$ *functions* $f_i f_j$, $i, j = 1, \ldots, k$, *exactly with respect to the weight function* ω.

Remark 2.3. *The classical* n*-point Gauss-Legendre quadrature is both a Gaussian quadrature for the Legendre polynomials* $P_0(x), \ldots, P_{2n-1}(x)$ *and an inner product quadrature for* $P_0(x), \ldots, P_{n-1}(x)$.

2.7 Interpolation

In this section we introduce terminology and definitions related to interpolation schemes which will be used in the remainder of this paper.

Definition 2.6. *A* k*-point linear interpolation scheme on the interval* $[a, b]$ *is a collection of linearly independent functions* $\alpha_1, \ldots, \alpha_k : [a, b] \to \mathbb{R}$, *a set of* k *nodes* $a \le x_1 < x_2 < \cdots < x_k \le b$, *and a linear map* $T : L^2([a, b]) \to \operatorname{span}\{\alpha_1, \ldots, \alpha_k\}$ *such that*

$$(Tf)(x_j) = f(x_j) \tag{32}$$

for all $j = 1, \ldots, k$. *The functions* $\alpha_1, \ldots, \alpha_k$ *are called the interpolation functions, the points* $x_1, \ldots, x_k$ *are called the interpolation nodes, and the mapping* T *is called the interpolation mapping. The coefficients* $c_1, \ldots, c_k$ *of* $Tf(x)$ *with respect to the basis* $\{\alpha_1, \ldots, \alpha_k\}$ *are called the interpolation coefficients for the function* f.

The following definition describes the stability of an interpolation scheme.

Definition 2.7. *Let* $\alpha_1, \ldots, \alpha_k$ *and* $x_1, \ldots, x_k$ *be an interpolation scheme with interpolation operator* T. *Let* U *be the* $k \times k$ *matrix with entries*

$$U_{ij} = \alpha_i(x_j), \quad i, j = 1, \ldots, k. \tag{33}$$

We say that the interpolation scheme is numerically stable *if*

$$\kappa(U) \le 10, \tag{34}$$

where $\kappa(U)$ *is the condition number of the matrix* U.

Remark 2.4. *In [16] it shown that given k bounded linearly-independent functions $\alpha_1, \ldots, \alpha_k : [a,b] \to \mathbb{R}$, there exist interpolation nodes $x_1, \ldots, x_k \in [a,b]$ such that the resulting interpolation scheme is stable in the sense of Definition 2.7.*

The following definition describes the relationship between an interpolation scheme and a collection of functions.

Definition 2.8. *We say that an interpolation scheme on the interval $[a,b]$ interpolates a collection of functions $f_1, \ldots, f_k : [a,b] \to \mathbb{R}$ with precision ϵ if*

$$|Tf_j(x) - f_j(x)| < \epsilon, \tag{35}$$

for $j = 1, \ldots, k$ and all $x \in (a,b)$.

2.8 Singular value decompositions

In this section we describe the singular value decomposition (SVD) of real matrices and finite collections of square-integrable functions.

Lemma 2.9 ([8])**.** *Suppose A is a real $n \times m$ matrix. Then there exists an $n \times n$ orthonormal matrix U, an $m \times m$ orthonormal matrix V, and an $n \times m$ diagonal matrix Σ with positive real entries such that*

$$A = U\Sigma V^*. \tag{36}$$

Here V^* denotes the transpose, the entries of Σ, denoted by $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{\min(m,n)} \geq 0$, are called the singular values of A, and the columns of U and V are the left and right singular vectors, respectively.

The following definition connects the singular values of a matrix A to its rank, see [] for example.

Definition 2.9. *Let A be an $n \times m$ real matrix, $n \geq m$, with singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_m \geq 0$. For $\epsilon > 0$ we define the ϵ-rank of A to be the smallest integer k such that $\sigma_{k+1} < \epsilon$.*

An analogous decomposition can also be defined for collections of functions, ie. when the columns of A are replaced by square-integrable functions, and is given by the following Theorem (see [4]).

Theorem 2.10. *Suppose $f_1, f_2, \ldots, f_m$ are real-valued functions in $L^2([a,b])$. Then there exist orthonormal functions $u_1, \ldots, u_m : [a,b] \to \mathbb{R}$, an $m \times m$ orthonormal matrix $V = [v_{ij}]$, and an $m \times m$ real diagonal matrix Σ with non-negative diagonal entries $\sigma_1, \ldots, \sigma_m$ such that*

$$f_j(x) = \sum_{i=1}^{k} u_i(x)\sigma_i v_{ij}, \tag{37}$$

for all $a \leq x \leq b$ and $1 \leq j \leq m$. By convention we order the functions $u_1, \ldots, u_k$ and choose the matrix V so that $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_k \geq 0$.

Remark 2.5. *Using Theorem 2.10, the definition of ϵ-rank extends to finite collections of functions.*

3 Numerical preliminaries

In the following four sections we describe several numerical tools to be used in the remainder of this paper.

3.1 Nested Gauss-Legendre discretizations

A nested Gauss-Legendre discretization is a collection of classical Gauss-Legendre nodes and weights on a binary partition of $[a, b]$. In this section we describe an algorithm which constructs a nested Gauss-Legendre discretization which adaptively discretizes a collection of functions to within a given tolerance ϵ. For each function the interval is first bissected and a Legendre expansion of the function is computed up to some fixed order $2K$ on each of the subintervals. For each subinterval, if any of the last K coefficients is greater than a pre-specified tolerance ϵ, the subinterval is subdivided once again. The process continues recursively until no subinterval has more than K coefficients of size greater than ϵ in its Legendre expansion.

We summarize this procedure in the following algorithm (see [4] for a more detailed discussion). As input it takes a collection of functions $f_1, \ldots, f_n : (a, b) \to \mathbb{R}$, a precision ϵ, and an integer K which controls how many of points are used on each subinterval. It outputs a collection of discretization nodes and weights defining a nested Gauss-Legendre discretization.

Algorithm 1.
Initial discretization:

Step 1. Construct the $2K$ Gauss-Legendre nodes $x_1, \ldots, x_{2K}$ on the interval (a, b).

Step 2. Let P_k denote the kth order Legendre polynomial. Determine the coefficients, $\alpha_1, \ldots, \alpha_{2K}$, in the expansion of f_j in terms of the Legendre polynomials P_k, $k = 0, \ldots, 2K - 1$, by solving the following linear system,

$$f_j(x_i) = \sum_{l=1}^{2K} \alpha_l \sqrt{\frac{2}{b-a}}\, P_l\left(2\frac{x_i - a}{b-a} + a\right), \quad i = 1, \ldots, 2K. \tag{38}$$

Step 3. If $\sum_{i=K+1}^{2K} |\alpha_i|^2 < \epsilon$, then the Kth order Legendre expansion for f_j on $[a, b]$ is sufficient. If the sum of the squares of the last K coefficients is greater than ϵ, then subdivide (a, b) into two intervals $[a, a/2 + b/2]$, and $[a/2 + b/2, b]$ and repeat the procedure recursively on each subinterval.

Step 4. Repeat steps 1-3 for each function f_j, $j = 1, \ldots, n$. Merge the resulting partitions of $[a, b]$ and form a discretization with K Gauss-Legendre nodes on each subinterval.

Remark 3.1. *The nested Gauss-Legendre discretization with nodes $x_1, \ldots, x_N \in [a, b]$ and weights $w_1, \ldots, w_N$ of the collection of functions $f_1, \ldots, f_k : [a, b] \to \mathbb{R}$, produced by Algorithm 1 provides the following:*

a) an ϵ-accurate quadrature rule for $f_1, \ldots, f_k$ (see Definition 2.2),

b) an ϵ-accurate inner-product quadrature for $f_1, \ldots, f_k$ (see Definition 2.5),

c) a stable, ϵ-accurate interpolation scheme for $f_1, \ldots, f_k$, where the interpolating functions are Legendre polynomials on a binary partition of $[a, b]$ (see Definitions 2.7 and 2.8).

3.2 SVD of a collection of functions

In this paper we use the following algorithm, adapted from [23], for computing the SVD of a collection of functions. The input of the algorithm is a set of functions $f_1, \ldots, f_m : [a, b] \to \mathbb{R}$ together with a user-specified precision ϵ. The outputs of the algorithm are the left singular functions $u_1, \ldots, u_k : [a, b] \to \mathbb{R}$ and the corresponding singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_k > 0$, see Theorem 2.10. In fact, the algorithm produces the vectors $(u_i(x_j))$, $j = 1, \ldots, N$, $i = 1, \ldots, k$, where $x_1, \ldots, x_N$ are the nodes of the nested Gauss-Legendre discretization of the functions $f_1, \ldots, f_k$ on $[a, b]$ (see Remark 3.1).

Algorithm 2.

Step 1. Construct a nested Gauss-Legendre discretization $x_1, \ldots, x_N$ and $w_1, \ldots, w_N$ of functions $f_1, \ldots, f_k : [a, b] \to \mathbb{R}$ via Algorithm 1.

Step 2. Construct the $n \times m$ matrix, A, with (i, j)th entry defined by

$$A_{ij} = \sqrt{w_i} f_j(x_i).$$

Step 3. Compute the singular value decomposition of the matrix A to obtain the factorization

$$A = U \Sigma V^*, \tag{39}$$

where U is an $n \times m$ matrix with orthogonal columns, V is an $m \times m$ matrix with orthogonal columns, and Σ is an $m \times m$ diagonal matrix whose jth entry is σ_j.

Step 4. For each j such that $\sigma_j \geq \epsilon$, form the vector $u_j \in \mathbb{R}^n$ with i^{th} entry $(u_j)_i$ given by

$$[u_j]_i = U_{ij} / \sqrt{w_i}.$$

3.3 Generalized Chebyshev quadrature and interpolation

In this section we outline an algorithm which generates a generalized Chebyshev quadrature and interpolation scheme. It takes as input a set of orthonormal functions $u_1, \ldots, u_n$, an interval $[a, b]$, and a precision $\epsilon > 0$. The algorithm outputs an n-point generalized Chebyshev quadrature and interpolation scheme which is accurate to within the specified precision.

Algorithm 3.

Step 1. Using Algorithm 1 construct a nested Gauss-Legendre discretization of $u_1, \ldots, u_n$ to tolerance ϵ. Let $x_1, \ldots, x_N$ denote the discretization nodes and $w_1, \ldots, w_N$ denote the corresponding weights.

Step 2. Form the $n \times N$ matrix X with entries defined by

$$X_{ij} = u_i(x_j)\sqrt{w_j}, \tag{40}$$

and the vector $r \in \mathbb{R}^n$ with entries

$$r_i = \sum_{j=1}^{N} u_i(x_j) w_j \approx \int_a^b u_i(x)\,\mathrm{d}x, \tag{41}$$

where the last approximation is accurate to within ϵ.

Step 3. Perform the pivoted Gram-Schmidt algorithm with reorthogonalization on the matrix X as follows:

a) Set $X_0 = X$. For $i = 1, 2, \ldots, n$ let σ_i denote the index of the column of X_i with the largest ℓ^2 norm.

b) Let X_i be the matrix obtained by interchanging the σ_ith and the ith columns of X_{i-1}. Orthogonalize the ith column of X_i to the previous $i-1$ columns. Orthogonalize the remaining $N-i$ columns of X_i to the ith column.

Step 4. Let $\widetilde{x}_i = x_{\sigma_i}$, $i = 1, \ldots, n$ and form the $n \times n$ matrix U with entries

$$U_{ij} = u_i(x_{\sigma_j})\sqrt{w_{\sigma_j}}. \tag{42}$$

Step 5. Solve the linear system $U\tilde{v} = r$ for $\tilde{v} = (\tilde{v}_1, \ldots, \tilde{v}_n) \in \mathbb{R}^n$. Return the nodes $\widetilde{x}_1, \ldots, \widetilde{x}_n$ and corresponding weights $\tilde{w}_1, \ldots, \tilde{w}_n$, where $\tilde{w}_i = \tilde{v}_i\sqrt{w_{\sigma_i}}$.

Remark 3.2. *Since the rows of the matrix X are orthonormal, it can be shown that the condition number of the matrix U given in (42) is approximately one (see [3]).*

Remark 3.3. *Algorithm 3 produces an interpolation scheme which is stable in the following sense. Suppose that $f \in L^2([a,b])$ and that the values of f at the points $\widetilde{x}_1, \ldots, \widetilde{x}_n$ are known. Let $\widetilde{U}$ be the $n \times n$ matrix with entries given by*

$$\widetilde{U}_{ij} = u_i(\widetilde{x}_j)\sqrt{\widetilde{w}_j}. \tag{43}$$

Then the operator $\widetilde{U}^{-1}$ maps the vector $(f(\widetilde{x}_1)\sqrt{\widetilde{w}_1}, \ldots, f(\widetilde{x}_n)\sqrt{\widetilde{w}_n})$ to the vector of interpolation coefficients $c_1, \ldots, c_n$ (see Definition 2.6). Furthermore, this operator is well-conditioned [3].

3.4 Generalized Gaussian quadratures

In this section we describe an algorithm, first introduced in [4], which constructs a generalized Gaussian quadrature for a collection of functions. In particular, given a sequence of functions $f_1, \ldots, f_{2n} : [a,b] \to \mathbb{R}$ which are square-integrable with respect to a weight function ω, it returns a set of n nodes $x_1, \ldots, x_n$ and weights $w_1, \ldots, w_n$ which integrate the functions $f_1, \ldots, f_{2n}$ with precision ϵ.

It proceeds by using Algorithm 3 to construct a generalized Chebyshev quadrature for $f_1, \ldots, f_{2n}$ and then eliminating n nodes one at a time. Each time it removes a node the algorithm performs a non-linear optimization on the locations of the nodes to ensure that the functions $f_1, \ldots, f_{2n}$ are still integrated with precision ϵ. The resulting quadrature scheme is guaranteed to integrate the collection of functions with precision ϵ.

Algorithm 4 ([4])**.**

Generation of initial quadrature:

Step 1. Construct a nested Gauss-Legendre discretization to tolerance $\epsilon_0 \ll \epsilon$ (typically $\epsilon_0 < \epsilon/1000$ is sufficient). Let $x_1^L, \ldots, x_N^L$ and $w_1^L, \ldots, w_N^L$ denote the corresponding nodes and weights.

Step 2. Use Algorithm 2 to construct an SVD of the functions $f_1, \ldots, f_{2n}$ with precision ϵ_0. Let $u_1, \ldots, u_k$ denote the resulting singular functions, where σ_k is the smallest singular value greater than ϵ.

Step 3. Use Algortihm 3 to construct a k-point generalized Chebyshev quadrature for $u_1, \ldots, u_k$ with precision ϵ.

Reduction to generalized Gaussian quadrature:

Step 4. Given an m-point quadrature scheme $x_1^m, \ldots, x_m^m$, $w_1^m, \ldots, w_m^m$ integrating $u_1, \ldots, u_k$ with precision ϵ, try to produce an $(m-1)$-point quadrature scheme via the following steps:

- *a. Rank the nodes in order of increasing importance via the steps described in Stage 2 of [4], page 1777.*
- *b. Remove the node flagged as least significant to obtain $m-1$ nodes and weights. Run the Gauss-Newton algorithm on the locations of the nodes and the values of the weights to obtain an $(m-1)$-point quadrature integrating $u_1, \ldots, u_k$ with precision ϵ.*
- *c. If the Gauss-Newton algorithm fails to converge, replace the node removed in Step 4b and remove the next node in the ordering produced in Step 4a.*
- *d. Continue in this manner, attempting to remove each of the m nodes in turn, until an $(m-1)$-point quadrature is obtained. If all nodes are tried and none are successfully removed then the algorithm returns an m-point quadrature. If an $(m-1)$-point quadrature is obtained with the required accuracy, accept this quadrature and repeat this procedure beginning at Step 4a.*

Observation 3.4. We note that the algorithm occasionally fails to eliminate exactly n nodes. However, in such cases it typically returns a quadrature with a number of nodes close to the optimal number. Even when the algorithm fails, the resulting quadratures are guaranteed to integrate the specified functions with precision ϵ.

4 Numerical apparatus

In this section we describe the main algorithm of this paper. It consists of the following two distinct parts.

The first part is the construction of an interpolation scheme which interpolates the densities in the vicinity of corners, described in Theorem 2.7, as well as a quadrature scheme which integrates the kernels K and $\tilde{K}$ given by Definition 2.1 times the densities near the corner.

The second part is the discretization of the integral equations (7) and (8) via a Nyström scheme using the quadratures and interpolation scheme constructed in the previous part.

4.1 Corner discretization

The following observation establishes the rank of a certain collection of functions appearing in Theorem 2.7, and is used in the discretization of the functions representing the solution to (7) and (8) in the vicinity of corners.

Observation 4.1. Consider the family of functions $\mathcal{F} \subset L^2([0, 1/2])$ defined by

$$\mathcal{F} = \{t^\mu \mid \mu = 0 \cup [1/2, 40]\}. \tag{44}$$

For $\epsilon = 10^{-16}$ the ϵ-rank of this collection of functions is 33 (see Definition 2.9). Hence there exists a 33-point interpolation scheme which interpolates all functions in this family in the L^2-sense to within 10^{-16}.

Table 1 lists the first fifty-three singular values of the family of functions, $\mathcal{F}$, calculated using Algorithm 2. All singular values after the 33^{rd} are less than 10^{-16}.

Observation 4.2. We choose the upper limit of our range of μ in (44) to be 40, noting that increasing this upper bound does not significantly increase the rank of the collection of functions.

The following observation, based on numerical experiments, is useful in the construction of stable interpolation schemes for certain familes of functions. It is closely related to the properties of the left-singular vectors of the truncated Laplace transform (see [13]).

Observation 4.3. In practice, if Algorithm 2 is used to compress finite collections of powers, t^{μ_i}, where $\mu_i \in [0, 50]$, $i = 1, \ldots, N$, $t \in [0, 1/2]$, then the $(k+1)^{\text{st}}$ singular function has exactly k roots. The resulting roots, $r_1, \ldots, r_k$, together with the first k singular functions, $u_1, \ldots, u_k$, yield a stable k-point interpolation scheme which interpolates the family of functions $\{t^{\mu_i}\}_1^N$ with precision $\sim \sigma_{k+1}$, where σ_{k+1} is the $(k+1)^{\text{st}}$ singular value.

The previous observation can be used as the basis for an algorithm to construct an interpolation scheme which interpolates the family of functions $\mathcal{F}$ given in Lemma 4.1. As input it takes an interval $[a, b]$, a range of exponents $[\mu_0, \mu_1]$, and real numbers $\epsilon_{\text{quad}} \gg \epsilon_{\text{svd}} > 0$ (typically $\epsilon_{\text{svd}} < \epsilon_{\text{quad}}/1000$ is sufficient). As output it returns a set of discretization nodes $s_1, \ldots, s_{N_D}$, a set of discretization weights $w_1, \ldots, w_{N_D}$, and an orthonormal set of interpolation functions $u_1, \ldots, u_{N_D} : [a, b] \to \mathbb{R}$, which interpolate the monomials t^μ, $\mu = 0 \cup [\mu_0, \mu_1]$ with precision ϵ_{quad} on the interval $[a, b]$.

Remark 4.4. *If* $\epsilon_{\text{quad}} = 10^{-16}$, $\epsilon_{\text{svd}} = 10^{-25}$, $a = 10^{-16}$, $b = \frac{1}{2}$, $\mu_0 = \frac{1}{2}$, *and* $\mu_1 = 40$, *then the algorithm outputs a set of 33 nodes* $s_i \in (10^{-16}, \frac{1}{2}]$ $i = 1, \ldots, 33$, *a corresponding set of (positive) weights* w_i, $i = 1, \ldots, 33$, *and a set of orthonormal interpolating functions* $u_i : (10^{-16}, \frac{1}{2}] \to \mathbb{R}$, $i = 1, \ldots, 33$. *The condition number of the interpolation operator mapping the values of* u_i *at the discretization nodes to the coefficients of the expansion in terms of the* u_i *is* 1.0031.

Algorithm 5.

k	σ_k
1	$0.4093476434043761 \cdot 10^{1}$
2	$0.6732190945529545 \cdot 10^{0}$
3	$0.1991437561410569 \cdot 10^{0}$
4	$0.4249343499170006 \cdot 10^{-1}$
5	$0.9160204770814034 \cdot 10^{-2}$
6	$0.2137216704928454 \cdot 10^{-2}$
7	$0.5318584271903745 \cdot 10^{-3}$
8	$0.1391816187224777 \cdot 10^{-3}$
9	$0.3792387773306178 \cdot 10^{-4}$
10	$0.1068481388222104 \cdot 10^{-4}$
11	$0.3096942318385288 \cdot 10^{-5}$
12	$0.9198859485549570 \cdot 10^{-6}$
13	$0.2791642683875489 \cdot 10^{-6}$
14	$0.8635018805679370 \cdot 10^{-7}$
15	$0.2717003840419811 \cdot 10^{-7}$
16	$0.8682258916589030 \cdot 10^{-8}$
17	$0.2813815939355092 \cdot 10^{-8}$
18	$0.9237895713433608 \cdot 10^{-9}$
19	$0.3069234590658991 \cdot 10^{-9}$
20	$0.1031074378345823 \cdot 10^{-9}$
21	$0.3499643548451834 \cdot 10^{-10}$
22	$0.1199340456049553 \cdot 10^{-10}$
23	$0.4147541466692028 \cdot 10^{-11}$
24	$0.1446579417390698 \cdot 10^{-11}$
25	$0.5086231278647431 \cdot 10^{-12}$
26	$0.1802078487712332 \cdot 10^{-12}$
27	$0.6431572556457143 \cdot 10^{-13}$
28	$0.2311469201117825 \cdot 10^{-13}$
29	$0.8363097810786797 \cdot 10^{-14}$
30	$0.3045467269488719 \cdot 10^{-14}$
31	$0.1116004460259128 \cdot 10^{-14}$
32	$0.4114668217771291 \cdot 10^{-15}$
33	$0.1526146413581282 \cdot 10^{-15}$
34	$0.5693364549045461 \cdot 10^{-16}$
35	$0.2135613404631228 \cdot 10^{-16}$
36	$0.8050767939545915 \cdot 10^{-17}$
37	$0.3047710412066927 \cdot 10^{-17}$
38	$0.1157380293328680 \cdot 10^{-17}$
39	$0.4403751727167243 \cdot 10^{-18}$
40	$0.1676913636789886 \cdot 10^{-18}$
41	$0.6384607793139196 \cdot 10^{-19}$
42	$0.2428980974022445 \cdot 10^{-19}$
43	$0.9230745083840269 \cdot 10^{-20}$
44	$0.3503590059042384 \cdot 10^{-20}$
45	$0.1328107460265278 \cdot 10^{-20}$
46	$0.5027813015037113 \cdot 10^{-21}$
47	$0.1900716323097294 \cdot 10^{-21}$
48	$0.7174490364875012 \cdot 10^{-22}$
49	$0.2703495093270228 \cdot 10^{-22}$
50	$0.1016764759511848 \cdot 10^{-22}$
51	$0.3815365222653915 \cdot 10^{-23}$
52	$0.1422486286856146 \cdot 10^{-23}$
53	$0.5169475795740504 \cdot 10^{-24}$

Table 1: Singular values for the collection of functions t^{μ}, $10^{-16} \leq t \leq \frac{1}{2}$, where $\mu \in \{0\} \cup \left[\frac{1}{2}, 40\right]$.

Step 1. Sample the powers in the range $[\mu_0, \mu_1]$ *using suitably rescaled* $(N-1)$ *Gauss-Legendre nodes to obtain a discrete set of* N *powers* $\nu_1 < \nu_2 < \cdots < \nu_N$ *where* $\nu_1 = 0$, *and* $\mu_0 \leq \nu_j \leq \mu_1$ *for* $j = 2, \ldots, N$.

Step 2. Use Algorithm 2 to compute the SVD of the set of functions $f_j(t) = t^{\nu_j}$, $j = 1, \ldots, N$, $a \leq t \leq b$, *with precision* ϵ_{svd}, *and obtain a set of singular values* $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_M$ *and corresponding singular functions* $u_1, \ldots, u_M$.

Step 3. Let k *denote the number of singular values greater than* ϵ_{quad}. *Compute the roots of the* $(k+1)^{\text{st}}$ *singular value and denote them by* $r_1, \ldots, r_s$ *(see Observation 4.3). If* $s \neq k$ *then Algorithm 3 should be used instead to determine suitable points* $r_1, \ldots, r_k$.

Step 4. Construct the $k \times k$ *matrix* U *with entries* U_{ij} *defined by*

$$U_{ij} = u_i(r_j), \tag{45}$$

and the vector Y *of length* k *with entries*

$$Y_i = \int_a^b u_i(x)\,\mathrm{d}x. \tag{46}$$

Step 5. Solve the linear system $UW = Y$ *for the vector* $W = (w_1, \ldots, w_k) \in \mathbb{R}^k$. *For* N *sufficiently large (typically* $N \leq 200$*) the* k*-point interpolation scheme with roots* $r_1, \ldots, r_k$, *weights* $w_1, \ldots, w_k$, *and functions* $u_1, \ldots, u_k$ *interpolates the inputed family of functions to the desired precision.*

4.2 Quadrature in the vicinity of corners

In light of Lemma 2.6, in order to integrate the kernels K and $\tilde{K}$ times density, it will also be necessary to approximate integrals of the form

$$\int_0^{\frac{1}{2}} \frac{s_i \sin(\pi\alpha)}{s_i^2 + t^2 - 2s_i t\cos(\pi\alpha)} t^{\mu}\mathrm{d}t, \tag{47}$$

where s_i, $i = 1, \ldots, N_D$ are the discretization nodes constructed by Algorithm 5, and $\mu = i/\alpha, i/(2-\alpha)$, $i = 0, 1, \ldots, M$. The integer M is a cut-off on the maximum number of terms in the expansion of the density in singular powers. In practice it is chosen sufficiently large so that the resulting quadrature scheme does not change when M is increased.

The following algorithm constructs a set of quadrature nodes for each discretization node s_i which approximates integrals of the form (47) with the specified precision ϵ. Specifically, it takes as input a discretization node s_i with corresponding weight w_i, a range of angles $[\pi\alpha_0, \pi\alpha_1]$, two integers and a precision ϵ. It returns a set of quadrature nodes $t_1^{(i)}, \ldots, t_{N_i}^{(i)}$ and weights $w_1^{(i)}, \ldots, w_{N_i}^{(i)}$ which integrate (47) for all $\alpha \in [\alpha_0, \alpha_1]$.

Algorithm 6.

Step 1. Sample the range $[\alpha_0, \alpha_1]$ at K points $a_1, \ldots, a_K \in [\alpha_0, \alpha_1]$. Typically, the points $\{a_j\}$ are obtained using suitably-scaled Gauss-Legendre nodes. The integer K is chosen sufficiently large so that adding an additional point does not increase the ϵ-rank.

Step 2. For each discretization node s_i with corresponding weight w_i, use Algorithm 4 to obtain a Gaussian or near-Gaussian quadrature for the functions

$$\frac{s_i \sin(\pi a_k)\sqrt{w_i}}{s_i^2 + t^2 - 2 s_i t \cos(\pi a_k)} t^{j/a_k},$$
$$\frac{s_i \sin(\pi a_k)\sqrt{w_i}}{s_i^2 + t^2 - 2 s_i t \cos(\pi a_k)} t^{j/(2-a_k)}, \tag{48}$$

where $k = 1, \ldots, K,$ and $j = 1, \ldots, M$, $t \in (0, \frac{1}{2})$.

For each angle $\pi\alpha$, by combining the results of Algorithms 5 and 6, we construct an $N_D \times N_D$ matrix K^α which maps the density near a corner (evaluated at the discretization nodes $s_1, \ldots, s_{N_D}$) to the integral of the kernel times the density evaluated at the discretization nodes on the opposite side of the corner. The following lemma gives an explicit formula for the entries of this matrix.

Lemma 4.1. *Let $u_1, \ldots, u_{N_D}$ be the functions produced by Algorithm 5 and $s_1, \ldots, s_{N_D}$, and $w_1, \ldots, w_{N_D}$ be the corresponding nodes and weights, respectively. Let U_{ij} denote the $N_D \times N_D$ matrix with entries*

$$U_{ij} = \sqrt{w_j}\, u_i(s_j). \tag{49}$$

Define the $N_D \times N_D$ matrix K^α_{ij} by

$$K^\alpha_{ij} = \sqrt{w_i}\sqrt{w_j} \sum_{k=1}^{N_i} K\big(\gamma(s_i), \gamma(t_k^{(i)})\big) w_k^{(i)} \left(\sum_{\ell=1}^{N_D} u_\ell(t_k^{(i)})\, (U^{-1})_{\ell j} \right), \tag{50}$$

where $t_j^{(i)}, w_j^{(i)}$, $i = 1, \ldots, N_D$, $j = 1, \ldots, N_i$ are the nodes and weights, respectively, generated by Algorithm 6 for a given range $[\alpha_0, \alpha_1]$, see Figure 3. If $\alpha \in [\alpha_0, \alpha_1]$ then K^α maps the density near a corner with angle $\pi\alpha$ to the potential generated on the opposite edge of the corner in an L^2 sense.

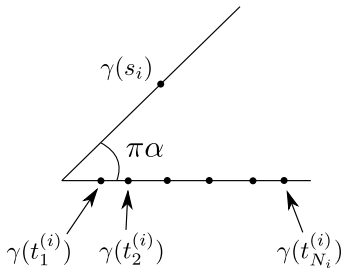

Figure 3: Quadrature and discretization nodes near a corner.

Remark 4.5. *Note that rescaling the interval from* $(0, 1/2)$ *to* $(0, c)$*, where* $c > 0$ *is some arbitrary real number, does not change the entries of the matrix* K^α*.*

Remark 4.6. *The interpolation scheme constructed via Algorithm 5 is universal in the sense that the set of discretization nodes, weights, and functions is independent of the angle of the corner. The quadrature rules obtained using Algorithm 6 are valid for a range of angles; using six such ranges of angles we obtain sets of quadratures which work for all angles in the range* $\left(\frac{\pi}{32}, \left(2 - \frac{1}{32}\right)\pi\right)$*.*

4.3 Nyström discretization

In this section we discretize the boundary integral equations (7) and (8) using a Nyström method. In the following we restrict our attention to the interior Dirichlet problem (7); the exterior Dirichlet problem is solved in a similar fashion. We first construct a discretization of the boundary with nodes $s_1, \ldots, s_N$, and weights $w_1, \ldots, w_N$, which enable interpolation of the left- and right-hand sides of (7) with precision ϵ. We proceed by enforcing equality at the discretization nodes, which yields the system of equations

$$g(s_i)\sqrt{w_i} = -\pi\rho(s_i)\sqrt{w_i} + \sqrt{w_i}\int_0^L K(\gamma(s_i), \gamma(t))\,\rho(t)\,\mathrm{d}t, \quad i = 1, \ldots, N. \tag{51}$$

Scaling by the square root of the weights is equivalent to solving the problem in the L^2 sense, and results in discretized operators with condition numbers which are close to those of the original physical systems [3].

We obtain the discretization in the following manner. Suppose the boundary of Ω is a polygon composed of K edges $\Gamma_1, \ldots, \Gamma_K$, with lengths $\ell_1, \ldots, \ell_K$, respectively. We denote the corners by C_i, $i = 1, \ldots, K$, indexed so that C_j is adjacent to Γ_{j-1} and Γ_j, where we define Γ_0 to be Γ_K. We proceed by dividing Γ into a set of panels (intervals) each of which is contained entirely within one of the edges. Let P_i^c denote the corner panels (those which terminate at a corner) and P_i^s denote the remaining panels. For a given corner panel P_i^c, let $\bar{P}_i^c$ denote the corner panel which is adjacent to the same vertex as P_i^c (here we define $\bar{\bar{P}}_i^c = P_i^c$). An illustration of this process is shown in Figure 4, noting that in Figure 4b, $\overline{P}_1^c = P_5^c$.

We choose the lengths of the corner panels so that for each corner, C_i, the two panels which meet at C_i (P_i^c and $\bar{P}_i^c$) are of equal length. Typically, their lengths are chosen to be a fixed fraction of the minimum of the lengths of the edges to which they belong. Let P_i^s denote the remaining panel on Γ_i, $N_c = 2K$ denote the number of corner panels, and $N_s = K$ denote the number of non-corner panels.

Since the panels P_i^s are separated from the corners, the density is guaranteed to be smooth on each of them, and hence Gauss-Legendre quadrature nodes (or nested Gauss-Legendre discretizations) can be used to interpolate the densities as well as to integrate the kernel times the density. On the corner panels P_i^c, the density is singular and thus a specialized discretization is required. On these panels we use rescaled sets of the discretization nodes generated by Algorithm 5.

Next, to discretize (51) we require a quadrature scheme which approximates integrals

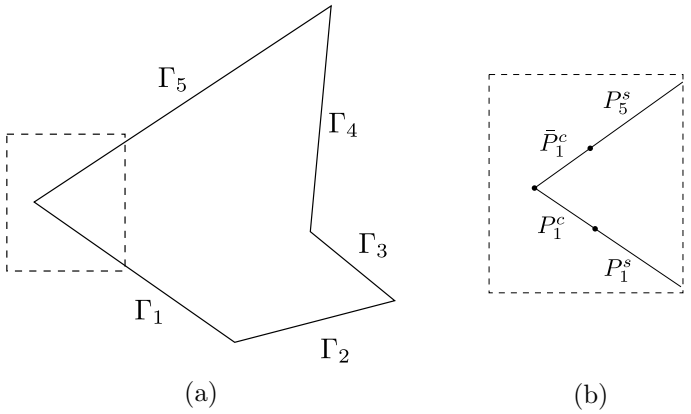

Figure 4: Discretization of a polygonal domain in $\mathbb{R}^2$.

of the form

$$\int_0^L K(\gamma(s_i),\gamma(t))\,\rho(t)\,\mathrm{d}t = \sum_{j=1}^{N_c}\int_{P_j^c} K(\gamma(s_i),\gamma(t))\,\rho(t)\,\mathrm{d}t + \sum_{j=1}^{N_s}\int_{P_j^s} K(\gamma(s_i),\gamma(t))\,\rho(t)\,\mathrm{d}t, \quad i=1,\ldots,N. \tag{52}$$

If $s_i \notin P_j^c$ for some $1 \le j \le N_c$, then the kernel is smooth and the discretization nodes can be used as quadrature nodes on all panels. Similarly, if $s_i \in P_j^c$, then the integrals over the Gaussian panels P_i^s as well as the integrals over the corner panels which are not adjacent to P_j^c can once again be computed accurately using the discretization nodes as quadrature nodes.

If $s_i \in P_j^c$, in order to compute the contribution to the integral from the panel $\bar{P}_j^c$, we use the quadrature nodes generated by Algorithm 6. To write the system in terms of the density evaluated at the discretization nodes we use our interpolation scheme to interpolate from the discretization nodes to the quadrature nodes. With the tolerances given in Remark 4.4 and the ranges of powers given in Remark 4.6, the result is a 33×33 matrix whose entries are given by Lemma 4.1.

The discretization procedure is summarized in the following algorithm. Here we assume our region Γ is divided into K line segments, $\Gamma_1,\ldots,\Gamma_K$, with lengths $\ell_1,\ldots,\ell_K$ respectively. We denote the unit tangent to Γ_j by (h_{xj}, h_{yj}), $j=1,\ldots,K$. Additionally, we denote the corners of Γ by $C_1,\ldots,C_K$, indexed so that C_j is adjacent to Γ_{j-1} and Γ_j (defining Γ_0 to be Γ_K), and the location of C_j by (c_{xj}, c_{yj}).

Algorithm 7.
Discretization:

Step 1. For each corner determine the length r_i of the corner panels. We choose

$$r_i = \sigma \min\{L_{i-1}, L_i\}, \tag{53}$$

where $0 < \sigma < 0.5$ is an arbitrary constant depending on the curve Γ (we find that $\sigma = 0.2$ is usually sufficient).

Step 2. Construct the corner panels P_j^c and $\bar{P}_j^c$ by rescaling the discretization nodes and weights $(s_1, \ldots, s_{N_D},\ w_1, \ldots, w_{N_D})$ produced by Algorithm 5. Specifically, P_j^c has N_D nodes located at $(x_i^{(j)}, y_i^{(j)})$ with weight $w_i^{(j)}$ defined by

$$\left(x_i^{(j)}, y_i^{(j)}\right) = (c_{xj}, c_{yj}) + r_j s_i \, (h_{xj}, h_{yj}) \tag{54}$$

$$w_i^{(j)} = w_i r_j, \tag{55}$$

for $i = 1, \ldots, N_D$, $j = 1, \ldots, K$. Similarly, $\bar{P}_j^c$ has D nodes located at $(\bar{x}_i^{(j)}, \bar{y}_i^{(j)})$ with weight $\bar{w}_i^{(j)}$ defined by

$$\left(\bar{x}_i^{(j)}, \bar{y}_i^{(j)}\right) = \left(c_{x,j}, c_{y,j}\right) - r_j s_i \big(h_{x,(j-1)}, h_{y,(j-1)}\big), \tag{56}$$

$$\bar{w}_i^{(j)} = w_i r_j, \tag{57}$$

for $i = 1, \ldots, N_D$, $j = 1, \ldots, K$.

Step 3. For each edge Γ_j, discretize the remainder P_j^s (ie. the portion not included in P_j^c or $\bar{P}_{j+1}^c$) using a k-point Gauss-Legendre discretization or nested Gauss-Legendre discretization.

Construction of the linear system:

Step 4. Let $x_1, \ldots, x_N$, $w_1, \ldots, w_N$ denote the discretization nodes and weights, respectively, produced by Steps 1-3 of the algorithm. Form the $N \times N$ matrix X with entries X_{ij} defined by

$$X_{ij} = -\pi\delta_{ij} + K(x_i, x_j)\sqrt{w_i w_j} \tag{58}$$

where δ_{ij} is the Kronecker delta, and K is the kernel function given in Definition 2.1.

Step 5. Correct the elements of X corresponding to interactions of adjacent corner panels. Specifically, if $x_{i_1}, \ldots, x_{i_{N_D}}$ are the discretization nodes of the corner panel P_j^c (with interior angle $\pi\alpha$) and $x_{j_1}, \ldots, x_{j_{N_D}}$ are the discretization of the adjacent corner panel $\bar{P}_j^c$, set

$$X_{i_s, j_r} = (K^\alpha)_{sr} \qquad s, r = 1, \ldots, N_D, \tag{59}$$

where K^α is the $N_D \times N_D$ matrix constructed in Lemma 4.1.

5 Numerical results

To solve the integral equations of potential theory corresponding to both the interior and exterior Dirichlet problem on polygonal domains, we use the above approach to discretize the integral equations. The resulting linear systems are solved using standard techniques. To illustrate the performance of our algorithm we solve the exterior and interior Dirichlet problem on the domains shown in Figures 5, 6, 7, and 8. The algorithm was implemented in Fortran 77 and the experiments were run on a 2.7 GHz Apple laptop with 8 Gb RAM.

Boundary conditions	Curve	Number of nodes	Maximum error	Precomputation time	Total run time	Condition number
E	Γ_1	294	$2.55 \cdot 10^{-15}$	0.2012	0.2098	14.52
I	Γ_1	294	$2.66 \cdot 10^{-15}$	0.2222	0.2308	16.36
E	Γ_2	648	$9.71 \cdot 10^{-16}$	0.2159	0.3249	101.7
I	Γ_2	648	$9.02 \cdot 10^{-16}$	0.2278	0.3543	65.63
E	Γ_3	1300	$4.44 \cdot 10^{-15}$	0.2577	1.1592	129.3
I	Γ_3	1300	$9.99 \cdot 10^{-16}$	0.2483	1.1591	80.66
E	Γ_4	20736 (1561)	$4.44 \cdot 10^{-15*}$	1.253	44.651	391.0
I	Γ_4	20736 (1564)	$1.41 \cdot 10^{-12*}$	1.174	43.100	241.3

Table 2: Numerical results for the interior (I) and exterior (E) Dirichlet problems.

To demonstrate the accuracy of our algorithm for the interior Dirichlet problem, we choose our boundary data g to be the result of an incident dipole placed *outside* the region. We solve the linear system to obtain the potential and use (6) to construct the solution away from the boundary. Similarly, for the exterior Dirichlet problem, we choose our boundary data g to be the result of an incident dipole placed *inside* the region. The linear system is solved to obtain the solution to the integral equations (7) and (10), which we then use to construct the solution to the boundary value problems (1) and (2) away from the boundary. In both cases an analytic solution exists (it is just the potential produced by the dipole) and is used to determine the accuracy of our solution. Specifically, the potential is evaluated analytically and numerically at a few arbitrary points and the maximum of the difference is calculated. The results are summarized in Table 2. All times are measured in seconds, E denotes the exterior Dirichlet problem, and I denotes the interior Dirichlet problem. For the domain shown in Figure 8 one level of the compression algorithm described in [15] was used; the number of nodes in the compressed system is given in parentheses. The errors in the resulting solution are limited by the compression rather than the quadratures, which we denote by an asterisk.

Remark 5.1. *The linear systems in rows 1-6 were solved using the QR algorithm. No attempt was made to optimize the CPU performance of our implementations; the CPU times of these experiments could be significantly improved by using an algorithm appropriate to the size of the problems being solved.*

As a further illustration of the algorithm, we apply the approach of this paper to calculate the potential and equipotential lines for the geometries shown in Figures 9-11. Here we consider the scattering problem wherein a dipole is placed inside the polygon Ω for the interior Dirichlet problem and in the exterior of Ω for the exterior Dirichlet problem. The boundary conditions are chosen so that the solution vanishes on the boundary of Ω. One level of the compression algorithm described in [15] is used to reduce the number of unknowns in the final linear system. Results are given in Table 3, the total field is shown in Figures 12, 14, and 16, and corresponding equipotential lines are shown in Figures 13, 15 and 17. Due to the singular nature of the field at the source in Figure 14 we plot $\log_{10}|u|$, where u is the total field. In all other cases the source is outside the region shown. We denote the number of nodes in the discretization by N_d, the number of nodes in the compressed system by N_c, the number of corners by n, the time required

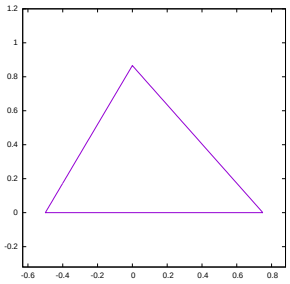

Figure 5: Γ_1 - a triangle in $\mathbb{R}^2$.

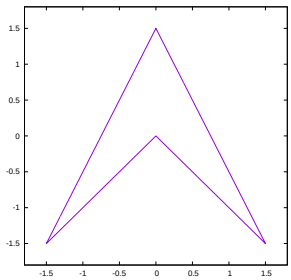

Figure 6: Γ_2 - a chevron in $\mathbb{R}^2$.

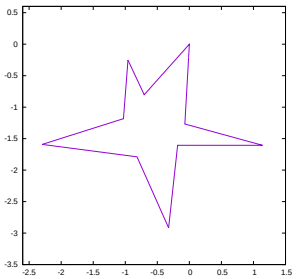

Figure 7: Γ_3 - a star in $\mathbb{R}^2$.

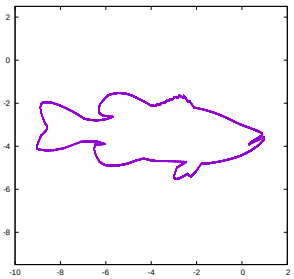

Figure 8: Γ_4 - a fish in $\mathbb{R}^2$.

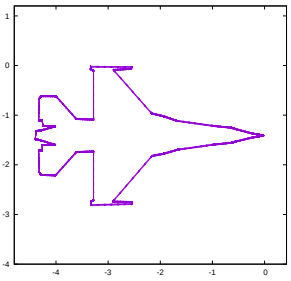

Figure 9: Γ_5 - top view of a falcon in $\mathbb{R}^2$.

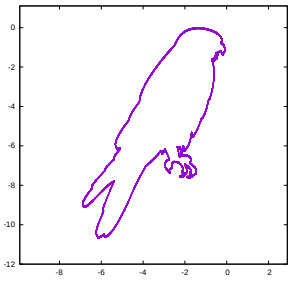

Figure 10: Γ_6 - a falcon in $\mathbb{R}^2$.

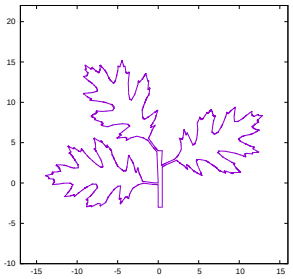

Figure 11: Γ_7 - an oak branch in $\mathbb{R}^2$.

to discretize and initialize the system by T_0 (in seconds), and the time to solve the final linear system by T_s (in seconds). The condition numbers reported in Table 3 correspond to the condition numbers of the compressed systems.

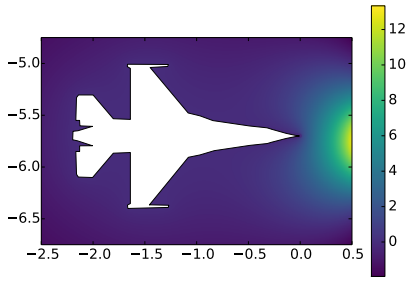

Figure 12: Scattering of a dipole from a plane in $\mathbb{R}^2$.

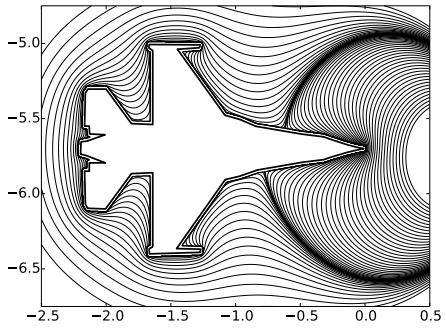

Figure 13: Equipotential lines of a plane in $\mathbb{R}^2$.

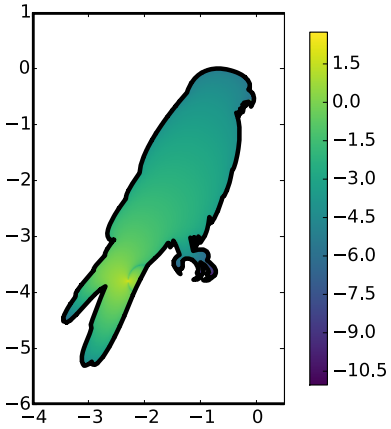

Figure 14: Log-scaled plot of interior scattering from a dipole for a falcon in $\mathbb{R}^2$.

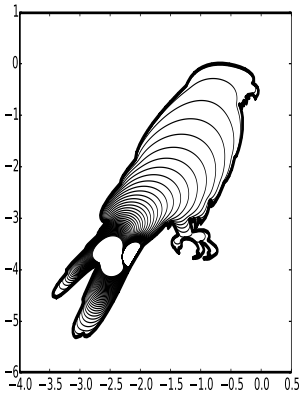

Figure 15: Equipotential lines of a falcon in $\mathbb{R}^2$.

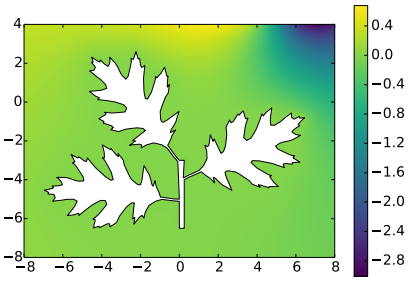

Figure 16: Scattering of a dipole from a branch in $\mathbb{R}^2$.

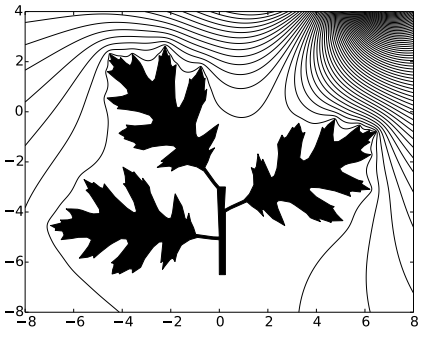

Figure 17: Equipotential lines of a branch in $\mathbb{R}^2$.

Boundary conditions	Curve	n	N_c	N_d	T_0	T_s	Condition number
E	Γ_5	51	725	16422	78.86	16.168	355.70
I	Γ_6	228	2523	29640	58.16	38.438	301.07
E	Γ_7	372	3318	60408	329.2	148.13	41581

Table 3: Numerical results for the interior (I) and exterior (E) Dirichlet problems.

6 Conclusions and extensions

In this paper we present an algorithm for solving Laplace's equation in the interior and exterior of polygonal domains subject to Dirichlet boundary conditions. The approach is based on the observation in [20] that when these boundary value problems are formulated as the boundary integral equations of classical potential theory, the solutions to the boundary integral equations in the vicinity of corners are representable by series of elementary functions. A central feature of our algorithm is the construction of specialized discretization nodes and quadrature rules to integrate these elementary functions near corners, with arbitrarily high precision while requiring relatively few nodes. We use these quadrature formulas and interpolation schemes as part of a Nyström method and demonstrate that it is highly accurate and the resulting linear systems are well-conditioned.

6.1 Neumann boundary conditions

In this paper we focus on Dirichlet boundary conditions. Similar analysis holds for Neumann boundary conditions, see [20]. In fact, with minor modifications the discretized linear system for the interior Dirichlet problem can be transposed to produce an accurate and well-conditioned discretization of the exterior Neumann problem. Likewise, the system for the exterior Dirichlet problem can be transposed to produce an accurate and well-conditioned discretization of the interior Neumann problem. An algorithm for Neumann boundary conditions will be described in a forthcoming paper.

6.2 Robin and mixed boundary conditions

Two additional classes of boundary conditions that have not yet been analyzed in detail are Robin conditions and mixed boundary conditions (Dirichlet on one side of the corner and Neumann on the other side). A detailed analysis of their properties, as well as the construction of algorithms for their solution, is currently underway and the results will be presented in forthcoming work.

6.3 Curved boundaries with corners

While this paper only deals with the solution of Laplaces equation on polygonal domains, in [19] this analysis is extended to curved boundaries with corners. Specifically, if the boundary of the domain is smooth except at a finite number of corners, then the solution to the corresponding boundary integral equations are representable by a slightly more

complicated series of elementary functions. In particular, the powers presented in Theorem 2.7 are multiplied by smooth functions and powers of logarithms. The numerical apparatus for solving Laplace's equation on regions of this type will be decribed in a future paper.

6.4 Generalization to three dimensions

The generalization of the apparatus of this paper to three dimensional polyhedra is fairly straightforward, but the detailed analysis has not been carried out. This line of research is being vigorously pursued.

6.5 The Helmholtz equation and biharmonic equation and Maxwell's equations on domains on corners

In this paper we consider the solution of boundary value problems for Laplace's equation on polygonal domains. A similar analysis holds for the Helmholtz equation [21] and the biharmonic equation [18] on polygonal domains. In particular, when the problems are formulated as the boundary integral equations of classical potential theory, the solutions are representable by series of Bessel functions in the Helmholtz case and complex powers in the biharmonic case. Papers detailing numerical algorithms that use this analysis to solve boundary value problems for the Helmholtz and biharmonic equations on polygonal domains are currently in preparation.

References

[1] K. E. Atkinson, *The Numerical Solution of Integral Equations of the Second Kind*, Cambridge University Press, 1997.

[2] J. Bremer, *A fast direct solver for the integral equations of scattering theory on planar curves with corners*, J. Comput. Phys., 231 (2012), pp. 1879–1899.

[3] ——, *On the Nyström discretization of integral equations on planar curves with corners*, Appl. Comput. Harmon. Anal., 32 (2012), pp. 45–64.

[4] J. Bremer, Z. Gimbutas, and V. Rokhlin, *A Nonlinear Optimization Procedure for Generalized Gaussian Quadratures*, SIAM J. Sci. Comput., 32 (2010), pp. 1761–1788.

[5] J. Bremer, V. Rokhlin, and I. Sammis, *Universal quadratures for boundary integral equations on two-dimensional domains with corners*, J. Comput. Phys., 229 (2010), pp. 8259–8280.

[6] R. Coifman, P. Jones, and S. Semmes, *Two elementary Proofs of the L^2 Boundedness of Cauchy Integrals on Lipschitz Curves.*, J. Am. Math. Soc., 2 (1989), pp. 553–564.

[7] A. Gillman, S. Hao, and P. G. Martinsson, *A simplified technique for the efficient and highly accurate discretization of boundary integral equations in 2d on domains with corners*, J. Comput. Phys., 256 (2014), pp. 214–219.

[8] G. Golub and C. F. Van Loan, *Matrix Computations*, Johns Hopkins University Press, 1996.

[9] P. Grisvard, *Elliptic problems in nonsmooth domains*, Boston: Pitman Advanced Pub. Program, 1985.

[10] S. Hao, A. H. Barnett, P. G. Martinsson, and P. Young, *High-order accurate Nyström discretization of integral equations with weakly singular kernels on smooth curves in the plane*, Adv. Comput. Math., 40 (2014).

[11] J. Helsing, *A fast and stable solver for singular integral equations on piecewise smooth curves*, SIAM J. Sci. Comput., 33 (2011), pp. 153–174.

[12] J. Helsing and K. Perfekt, *On the polarizability and capacitance of the cube*, Appl. Comput. Harmon. Anal., 34 (2013), pp. 445–468.

[13] R. R. Lederman and V. Rokhlin, *On the analytical and numerical properties of the truncated laplace transform II*, tech. rep., Yale University, 2015.

[14] R. S. Lehman, *Development of the Mapping Function at an Analytic Corner*, Pacific Journal of Mathematics, 7 (1957), pp. 1437–1449.

[15] P. Martinsson and V. Rokhlin, *A fast direct solver for boundary integral equations in two dimensions*, J. Comput. Phys., 1 (2005).

[16] P.-G. Martinsson, V. Rokhlin, and M. Tygert, *On interpolation and Integration in Finite-Dimensional Spaces of Bounded Functions*, Comm. App. Math. and Comp. Sci., 1 (2006), pp. 133–142.

[17] V. G. Maz'ya and S. M. Nikol'skii, *Analysis IV: Linear and Boundary Integral Equations (Encycopedia of Mathematical Sciences)*, New York: Springer-Verlag, 1991.

[18] M. Rachh and K. Serkh, *On the solution of Stokes equation on regions with corners*, arXiv:1711.04072, (2017).

[19] K. Serkh, *On the solution of elliptic partial differential equations on regions with corners III: curved boundaries*. Manuscript in preparation.

[20] ——, *On the solution of elliptic partial differential equations on regions with corners II: Detailed analysis*, Appl. Comput. Harmon. Anal., 5 (2017).

[21] K. Serkh and V. Rokhlin, *On the solution of the Helmholtz equation on regions with corners*, PNAS, 113 (2016), pp. 9171–9176.

[22] G. Verchota, *Layer potentials and regularity for the dirichlet problem for laplace's equation in lipschitz domains*, J. Funct. Anal, 59 (1984), pp. 572–611.

[23] N. Yarvin and V. Rokhlin, *Generalized gaussian quadratures and singular value decompositions of integral operators*, SIAM J. Sci. Comput., 20 (1998), pp. 699–718.

[24] V. G. Zargaryan, S. S.and Maz'ya, *The Asymptotic Form of the Solutions of the Integral Equations of Potential Theory in the Neighborhood of the Corner Points of a Contour*, PMM U.S.S.R., 48 (1984), pp. 120–124.