SIAM J. SCI. COMPUT.
Vol. 41, No. 4, pp. A2552–A2578

ON THE NUMERICAL SOLUTION OF ELLIPTIC PARTIAL DIFFERENTIAL EQUATIONS ON POLYGONAL DOMAINS[*]

JEREMY G. HOSKINS[†], VLADIMIR ROKHLIN[†], AND KIRILL SERKH[†]

Abstract. In the present paper we describe a class of algorithms for the solution of Laplace's equation on polygonal domains with Dirichlet and Neumann boundary conditions. It is well known that in such cases the solutions may 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 behavior near corners accurately. Our approach is based on the observation that if the boundary data are 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.

Key words. boundary value problems, potential theory, corners, Laplace equation, quadrature

AMS subject classifications. 31A05, 31A25, 65R20, 65N35

DOI. 10.1137/18M1199034

1. Introduction.

In classical potential theory, elliptic partial differential equations 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).

The behavior of solutions to Laplace's equation in the vicinity of corners has been the subject of extensive analysis (see [9], [17], [20], and [28] for representative examples). In particular, it is well known that solutions to the Dirichlet and Neumann boundary value problems are unique and exist in the L^2-sense (see [6], [26]) both for the differential and integral equations. However, near corners, the solutions to both the differential and integral equations may contain singularities; these singularities pose significant challenges to many existing methods. A natural approach is to refine the computational mesh near the corners. Near the corner, new panels are introduced with sizes which depend on the distance of the panel from the corner. Typically these additional panels are Gaussian panels which are frequently employed in the Nyström discretization of smooth boundaries. The refinement continues until the contribution from the panels nearest the corner is below a specified tolerance. The panels nearest

[*]Submitted to the journal's Methods and Algorithms for Scientific Computing section July 9, 2018; accepted for publication (in revised form) March 26, 2019; published electronically August 13, 2019.

https://doi.org/10.1137/18M1199034

Funding: The work of the second author was partially supported by AFOSR FA9550-16-1-0175 and by the Office of Naval Research (award N00014-14-1-0797/16-1-2123). The work of the third author was partially supported by the NSF Mathematical Sciences Postdoctoral Research Fellowship (awards 1606262) and AFOSR.

[†]Department of Mathematics, Yale University, New Haven, CT 06511 (jeremy.hoskins@yale.edu, rokhlin@cs.yale.edu, kirill.serkh@yale.edu).

the corner are then removed from the discretization. By proceeding in this way the discretization is guaranteed to approximate the true solution to within a user-specified tolerance.

Unfortunately the resulting linear systems can become prohibitively large for complex domains. A key observation used in ([12], [15],[14], [21], [11], and [13]) is that the interactions between the portion of the curve immediately adjacent to the corner and the rest of the domain are of low rank and hence highly compressible. Using *recursive compressed preconditioning* the extra degrees of freedom introduced by the refinement can be eliminated from the linear system. In fact the refinement and compression are typically done in tandem and multiple corners can be compressed in parallel. Moreover, this compression can be executed in times scaling linearly with the number of degrees of freedom introduced. The final system has approximately the same number of degrees of freedom as it would if the corners were not present at all. This technique has been applied to Laplace's equation (as well as the Helmholtz equation, the biharmonic equation, and Maxwell's equation) with a wide variety of boundary conditions. See [7], [5], [2], and [3] for other compression-based methods for solving Laplace's equation.

In this paper we describe an alternate approach based on recent analysis of the behavior of solutions to Laplace's equation in the vicinity of corners for Dirichlet and Neumann data. In [24] it was shown that if the boundary data on either side of the corners are smooth then, near the corners, the solution to the boundary integral equation has an expansion in terms of certain noninteger powers and noninteger 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 [24]. 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. In contrast with the existing algorithms our approach requires no refinement near corners. Instead the basis functions and discretization nodes are designed specifically to capture the behavior of solutions near corners. After discretization the resulting linear systems can be solved directly using standard techniques or fast solvers.

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.

In the formulation used in this paper, the operator associated with the boundary integral equation for the exterior Neumann problem is the adjoint of the operator for the interior Dirichlet problem. We use this fact to obtain an accurate discretization for the boundary integral equation for the Neumann problem by taking the adjoint of the discretized operator for the Dirichlet problem. This method for solving Neumann boundary value problems is demonstrated in section 5.

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.

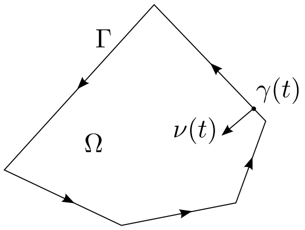


Fig. 1. *A polygonal curve in* $\mathbb{R}^2$.

Interior Dirichlet problem:

$$\begin{aligned} \nabla^2\phi(x) &= 0, \qquad x \in \Omega, \\ \lim_{\substack{x\to\gamma(t)\\ x\in\Omega}} \phi(x) &= g(t), \quad t \in [0,L]. \end{aligned} \tag{2.1}$$

Exterior Dirichlet problem:

$$\begin{aligned} \nabla^2\phi(x) &= 0, \qquad x \in \mathbb{R}^2 \setminus \overline{\Omega}, \\ \lim_{\substack{x\to\gamma(t)\\ x\in\mathbb{R}^2\setminus\overline{\Omega}}} \phi(x) &= g(t), \quad t \in [0,L]. \end{aligned} \tag{2.2}$$

Interior Neumann problem:

$$\begin{aligned} \nabla^2\phi(x) &= 0, \qquad x \in \Omega, \\ \lim_{\substack{x\to\gamma(t)\\ x\in\Omega}} \frac{\partial\phi}{\partial\nu(t)}(x) &= g(t), \quad t \in [0,L]. \end{aligned} \tag{2.3}$$

Exterior Neumann problem:

$$\begin{aligned} \nabla^2\phi(x) &= 0, \qquad x \in \mathbb{R}^2 \setminus \overline{\Omega}, \\ \lim_{\substack{x\to\gamma(t)\\ x\in\mathbb{R}^2\setminus\overline{\Omega}}} \frac{\partial\phi}{\partial\nu(t)}(x) &= g(t), \quad t \in [0,L]. \end{aligned} \tag{2.4}$$

The interior Dirichlet problem has a unique solution for all $g \in L^2([0,L])$ (see [26]). 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 $g \in L^2([0,L])$. Similarly, the exterior and interior Neumann problems have a unique solution (up to an additive constant) provided that

$$\int_0^L g(t)\,\mathrm{d}t = 0. \tag{2.5}$$

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:

$$\psi^0_{x_0}(x) = \log(\|x - x_0\|), \tag{2.6}$$

$$\psi^1_{x_0,h}(x) = \frac{\langle h, x_0 - x\rangle}{\|x_0 - x\|^2}, \tag{2.7}$$

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

$$\phi(x) = \int_0^L \psi^0_{\gamma(t)}(x)\rho(t)\,\mathrm{d}t, \tag{2.8}$$

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

$$\phi(x) = \int_0^L \psi^1_{\gamma(t),\nu(t)}(x)\rho(t)\,\mathrm{d}t, \tag{2.9}$$

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

$$g(s) = -\pi\rho(s) + \int_0^L \psi^1_{\gamma(t),\nu(t)}(\gamma(s))\,\rho(t)\,\mathrm{d}t \tag{2.10}$$

for all $s \in [0, L]$. If $g \in L^2([0, L])$ then (2.10) *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* (2.9)*.*

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

$$g(s) = \pi\rho(s) + \int_0^L \psi^1_{\gamma(t),\nu(t)}(\gamma(s))\,\rho(t)\,\mathrm{d}t \tag{2.11}$$

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

$$\int_0^L g(t)\,\mathrm{d}t = 0 \tag{2.12}$$

then (2.11) *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* (2.9)*.*

The following corollary follows immediately from Theorem 2.3.

Corollary 2.4. *Let $\rho \in L^2([0,L])$ and define $g : [0,L] \to \mathbb{R}$ by the following formula:*

$$g(s) = \pi\rho(s) + \int_0^L \left(c + \psi^1_{\gamma(t),\nu(t)}(\gamma(s))\right) \rho(t)\,\mathrm{d}t \tag{2.13}$$

for all $s \in [0,L]$, and an arbitrary constant c. If $g \in L^2([0,L])$ then (2.13) 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:

$$\phi(x) = \int_0^L \left(c + \psi^1_{\gamma(t),\nu(t)}(x)\right) \rho(t)\,\mathrm{d}t. \tag{2.14}$$

Theorem 2.5. *Let $\rho \in L^2([0,L])$ and define $g : [0,L] \to \mathbb{R}$ by the following formula:*

$$g(s) = -\pi\rho(s) + \int_0^L \psi^1_{\gamma(s),\nu(s)}(\gamma(t))\,\rho(t)\,\mathrm{d}t \tag{2.15}$$

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

$$\int_0^L g(t)\,\mathrm{d}t = 0 \tag{2.16}$$

then (2.15) has a unique solution for $\rho \in L^2([0,L])$. Furthermore, a solution to the exterior Neumann problem with boundary data g is given by (2.8).

Theorem 2.6. *Let $\rho \in L^2([0,L])$ and define $g : [0,L] \to \mathbb{R}$ by the following formula:*

$$g(s) = \pi\rho(s) + \int_0^L \psi^1_{\gamma(s),\nu(s)}(\gamma(t))\,\rho(t)\,\mathrm{d}t \tag{2.17}$$

for all $s \in [0,L]$. If $g \in L^2([0,L])$ then (2.17) has a solution for $\rho \in L^2([0,L])$. Furthermore, a solution to the interior Neumann problem with boundary data g is given by (2.8).

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 2.7. *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*

$$\nu(t) = (-\gamma_2'(t), \gamma_1'(t)). \tag{2.18}$$

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

$$\psi^1_{\gamma(s),\nu(s)}(\gamma(t)), \tag{2.19}$$

$$\psi^1_{\gamma(t),\nu(t)}(\gamma(s)), \tag{2.20}$$

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

$$\lim_{t\to s} \psi^1_{\gamma(s),\nu(s)}(\gamma(t)) = \lim_{t\to s} \psi^1_{\gamma(s),\nu(s)}(\gamma(t)) = -\frac{1}{2}\kappa(s), \tag{2.21}$$

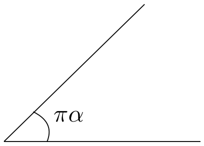


FIG. 2. *A wedge in* $\mathbb{R}^2$.

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* (2.19) *and* (2.20) *are analytic functions of* t *in a neighborhood of* s.

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

LEMMA 2.8. *Under the same assumptions as the previous lemma, if* γ *has a corner with interior angle* $\pi\alpha$ *at* t_0 *then*

$$\lim_{\substack{s\to t_0^+\\ t\to t_0^-}} \psi^1_{\gamma(s),\nu(s)}(\gamma(t)), \tag{2.22}$$

$$\lim_{\substack{s\to t_0^+\\ t\to t_0^-}} \psi^1_{\gamma(t),\nu(t)}(\gamma(s)), \tag{2.23}$$

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

$$\psi^1_{\gamma(t_0+h),\nu(t_0+h)}(\gamma(t_0 - h)) = \left(\frac{\cos \pi\alpha}{2 \sin \pi\alpha}\right) \frac{1}{h} + O(1) \tag{2.24}$$

if γ *is smooth in a neighborhood of the corner.*

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

DEFINITION 2.9. *For a given boundary* Γ, *we define the kernel,* $K : \Gamma \times \Gamma \to \mathbb{R}$ *by the formula*

$$K(x, y) = \psi^1_{y,\nu(y)}(x). \tag{2.25}$$

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*

$$\tilde{K}(x, y) = c + \psi^1_{y,\nu(y)}(x), \tag{2.26}$$

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 arc length parametrization of Γ, denoted by γ, is defined for $t \in [-1, 1]$ rather than $[0, 2]$.

LEMMA 2.10. *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

$$\psi_{\gamma(t),\nu(t)}(\gamma(s)) = \frac{s\sin\pi\alpha}{t^2+s^2+2st\cos\pi\alpha} \tag{2.27}$$

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 2.11 (see [24]). *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:*

$$\overline{L} = \left\lceil \frac{\alpha N}{2} \right\rceil, \tag{2.28}$$

$$\underline{L} = \left\lfloor \frac{\alpha N}{2} \right\rfloor, \tag{2.29}$$

$$\overline{M} = \left\lceil \frac{(2-\alpha) N}{2} \right\rceil, \tag{2.30}$$

$$\underline{M} = \left\lfloor \frac{(2-\alpha) N}{2} \right\rfloor. \tag{2.31}$$

Suppose further that ρ is defined via the formula

$$\begin{aligned}\rho(t) = b_0 &+ \sum_{i=1}^{\overline{L}} b_i |t|^{\frac{2i-1}{\alpha}} + \sum_{i=1}^{\underline{M}} b_{\overline{L}+i}|t|^{\frac{2i}{2-\alpha}}\left(\log|t|\right)^{\sigma_{N,\alpha}(i)}\\ &+ \sum_{i=1}^{\overline{M}} c_i \operatorname{sgn}(t)|t|^{\frac{2i-1}{2-\alpha}} + \sum_{i=1}^{\underline{L}} c_{\overline{M}+i}\operatorname{sgn}(t)|t|^{\frac{2i}{\alpha}}\left(\log|t|\right)^{\nu_{N,\alpha}(i)},\end{aligned} \tag{2.32}$$

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

$$\sigma_{N,\alpha}(i) = \begin{cases} 1 & \text{if } \frac{2i}{2-\alpha} = \frac{2j-1}{\alpha} \text{ for some } j\in\mathbb{Z},\ 1\le j\le \left\lceil\frac{\alpha N}{2}\right\rceil,\\ 0 & \text{otherwise,}\end{cases} \tag{2.33}$$

$$\nu_{N,\alpha}(i) = \begin{cases} 1 & \text{if } \frac{2i}{\alpha} = \frac{2j-1}{2-\alpha} \text{ for some } j\in\mathbb{Z},\ 1\le j\le \left\lceil\frac{(2-\alpha) N}{2}\right\rceil,\\ 0 & \text{otherwise.}\end{cases} \tag{2.34}$$

If g is defined by

$$g(t) = \pi\rho(s) + \int_{-1}^{1} \psi^1_{\gamma(t),\nu(t)}(\gamma(s))\rho(t)\,\mathrm{d}t \tag{2.35}$$

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

$$g(t) = \sum_{n=0}^{\infty} \beta_n |t|^n + \sum_{n=0}^{\infty} \gamma_n \operatorname{sgn}(t)|t|^n \tag{2.36}$$

for all $-1\le t\le 1$. Conversely, suppose that g has the form (2.36). Suppose further that N is an arbitrary positive integer. Then, for all angles $\pi\alpha$ there exist unique real numbers $b_0, b_1, \ldots, b_N$ and $c_0, c_1, \ldots, c_N$ such that ρ, defined by (2.32), solves (2.35) to within an error $\mathrm{O}(t^{N+1})$.

Remark 2.12. A similar result holds for the case where the factor of π in (2.35) is replaced by $-\pi$; the change in sign corresponds to replacing the boundary integral equation for the exterior Dirichlet problem (2.11) with the boundary integral equation corresponding to interior Dirichlet problem (2.10). Similar expansions also hold for both the exterior and interior Neumann problems, in which case the singular powers are obtained by subtracting one from the singular powers arising in the Dirichlet problem.

The following corollary characterizes the behavior of the solutions to (2.10) and (2.13) in the vicinity of a corner.

COROLLARY 2.13. *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 arc length parametrization of Γ in the vicinity of the corner, with $\psi(0)$ coinciding with the corner. If the boundary data, g, are analytic on either side of the corner then there exist unique real numbers $b_0, b_1, \ldots, b_N$ and $c_0, c_1, \ldots, c_N$ such that the density, ρ, defined by* (2.32) *satisfies* (2.10) *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 2.14. *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 $a \le x_1 < \cdots < x_n \le b$, 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*

$$\int_a^b f_i(x)\,\mathrm{d}x = \sum_{j=1}^{n} f_i(x_j) w_j, \qquad i = 1, \ldots, m, \tag{2.37}$$

and with precision $\epsilon > 0$ if

$$\left| \int_a^b f_i(x)\,\mathrm{d}x - \sum_{j=1}^{n} f_i(x_j) w_j \right| < \epsilon, \quad i = 1, \ldots, m. \tag{2.38}$$

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

DEFINITION 2.15. *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 2.16. *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 2.17. *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 2.18. The classical n-point Gauss–Legendre quadrature is both a Gaussian quadrature for the Legendre polynomials $P_0(x), \dots, P_{2n-1}(x)$ and an inner product quadrature for $P_0(x), \dots, P_{n-1}(x)$.

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

$$(Tf)(x_j) = f(x_j) \tag{2.39}$$

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

The following definition describes the stability of an interpolation scheme.

Definition 2.20. *Let $\alpha_1, \dots, \alpha_k$ and $x_1, \dots, x_k$ be an interpolation scheme with interpolation operator T. Let U be the $k \times k$ matrix with entries*

$$U_{ij} = \alpha_i(x_j), \quad i, j = 1, \dots, k. \tag{2.40}$$

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

$$\kappa(U) \le 10, \tag{2.41}$$

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

Remark 2.21. In [19] it shown that given k bounded linearly independent functions $\alpha_1, \dots, \alpha_k : [a, b] \to \mathbb{R}$, there exist interpolation nodes $x_1, \dots, x_k \in [a, b]$ such that the resulting interpolation scheme is stable in the sense of Definition 2.20.

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

Definition 2.22. *We say that an interpolation scheme on the interval $[a, b]$ interpolates a collection of functions $f_1, \dots, f_k : [a, b] \to \mathbb{R}$ with precision ϵ if*

$$|Tf_j(x) - f_j(x)| < \epsilon \tag{2.42}$$

for $j = 1, \dots, k$ 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 2.23 (see [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*

$$A = U \Sigma V^*. \tag{2.43}$$

Here V^* denotes the transpose, the entries of Σ, denoted by $\sigma_1 \ge \sigma_2 \ge \cdots \ge \sigma_{\min(m,n)} \ge 0$, 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.

Definition 2.24. *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, i.e., when the columns of A are replaced by square-integrable functions, and is given by the following theorem (see [4]).

Theorem 2.25. *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 nonnegative diagonal entries $\sigma_1, \ldots, \sigma_m$ such that*

$$f_j(x) = \sum_{i=1}^{k} u_i(x) \sigma_i v_{ij} \tag{2.44}$$

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 2.26. Using Theorem 2.25, the definition of ϵ-rank extends to finite collections of functions.

2.9. Adjoint method.

The following lemma relates the discretization of the inverse of an operator to the adjoint of the discretization of its inverse.

Lemma 2.27. *Suppose $A : L^2([0,L]) \to L^2([0,L])$ is a bounded invertible operator and that A_ϵ is an operator such that*

$$\left| \langle f, A^{-1} g \rangle - \langle f, A_\epsilon^{-1} g \rangle \right| \leq \epsilon \|f\| \|g\| \tag{2.45}$$

for all f and g which are piecewise analytic. Here $\langle \cdot, \cdot \rangle$ denotes the inner product on $L^2([0,L])$ and $\| \cdot \|$ denotes the norm for $L^2([0,L])$. Then, for all piecewise analytic functions f and g,

$$\left| \langle f, (A^{-1})^* g \rangle - \langle f, \left(A_\epsilon^{-1}\right)^* g \rangle \right| \leq \epsilon \|f\| \|g\|, \tag{2.46}$$

where * denotes the adjoint.

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 bisected 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 prespecified 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 points are used on each subinterval. It outputs a collection of discretization nodes and weights defining a nested Gauss–Legendre discretization.

Algorithm 3.1

```
Initial discretization:
Step 1. Construct the $2K$ Gauss–Legendre nodes $x_1, \dots, x_{2K}$ on the interval $(a, b)$.
Step 2. Let $P_k^{[a,b]}$ denote the $k$th order Legendre polynomial on the interval $[a, b]$.
        Determine the coefficients, $\alpha_1, \dots, \alpha_{2K}$, in the expansion of $f_j$ in terms of
        the Legendre polynomials $P_k^{[a,b]}$, $k = 0, \dots, 2K - 1$, by solving the following
        linear system,

        (3.1)    $f_j(x_i) = \sum_{l=1}^{2K} \alpha_l P_l^{[a,b]}(x), \quad i = 1, \dots, 2K.$

Step 3. If $\sum_{i=K+1}^{2K} |\alpha_i|^2 < \epsilon$ then the $K$th 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
        $\epsilon$ 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 $f_j$, $j = 1, \dots, n$. 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 $x_1, \dots, x_N \in [a, b]$ and weights $w_1, \dots, w_N$ of the collection of functions $f_1, \dots, f_k : [a, b] \to \mathbb{R}$, produced by Algorithm 3.1 provides the following:

(a) an ϵ-accurate quadrature rule for $f_1, \dots, f_k$ (see Definition 2.14);

(b) an ϵ-accurate inner-product quadrature for $f_1, \dots, f_k$ (see Definition 2.17);

(c) a stable, ϵ-accurate interpolation scheme for $f_1, \dots, f_k$, where the interpolating functions are Legendre polynomials on a binary partition of $[a, b]$ (see Definitions 2.20 and 2.22).

3.2. SVD of a collection of functions. In this paper we use the following algorithm, adapted from [27], for computing the SVD of a collection of functions. The input of the algorithm is a set of functions $f_1, \dots, f_m : [a, b] \to \mathbb{R}$ together with a user-specified precision ϵ. The outputs of the algorithm are the left singular functions $u_1, \dots, u_k : [a, b] \to \mathbb{R}$ and the corresponding singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_k > 0$; see Theorem 2.25. In fact, the algorithm produces the vectors $(u_i(x_j))$, $j = 1, \dots, N$, $i = 1, \dots, k$, where $x_1, \dots, x_N$ are the nodes of the nested Gauss–Legendre discretization of the functions $f_1, \dots, f_k$ on $[a, b]$ (see Remark 3.1).

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 $u_1, \dots, u_n$, 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.

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 (3.5) is approximately one (see [3]).

Remark 3.3. Algorithm 3.3 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

Algorithm 3.2

Step 1. Construct a nested Gauss–Legendre discretization $x_1, \dots, x_N$ and $w_1, \dots, w_N$ of functions $f_1, \dots, f_k : [a, b] \to \mathbb{R}$ via Algorithm 3.1.

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 SVD of the matrix A to obtain the factorization

$$A = U \Sigma V^*, \tag{3.2}$$

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 ith entry $(u_j)_i$ given by

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

Algorithm 3.3

Step 1. Using Algorithm 3.1 construct a nested Gauss–Legendre discretization of $u_1, \dots, u_n$ to tolerance ϵ. Let $x_1, \dots, x_N$ denote the discretization nodes and $w_1, \dots, w_N$ denote the corresponding weights.

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

$$X_{ij} = u_i(x_j)\sqrt{w_j}, \tag{3.3}$$

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

$$r_i = \sum_{j=1}^{N} u_i(x_j) w_j \approx \int_a^b u_i(x)\, \mathrm{d}x, \tag{3.4}$$

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 $i = 1, 2, \dots, n$ 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 $\widetilde{x}_i = x_{\sigma_i}$, $i = 1, \dots, n$ and form the $n \times n$ matrix U with entries

$$U_{ij} = u_i(x_{\sigma_j})\sqrt{w_{\sigma_j}}. \tag{3.5}$$

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

Algorithm 3.4

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 $x_1^L, \dots, x_N^L$ and $w_1^L, \dots, w_N^L$ denote the corresponding nodes and weights.

Step 2. Use Algorithm 3.2 to construct an SVD of the functions $f_1, \dots, f_{2n}$ with precision ϵ_0. Let $u_1, \dots, u_k$ denote the resulting singular functions, where σ_k is the smallest singular value greater than ϵ.

Step 3. Use Algortihm 3.3 to construct a k-point generalized Chebyshev quadrature for $u_1, \dots, u_k$ with precision ϵ.

Reduction to generalized Gaussian quadrature:

Step 4. Given an m-point quadrature scheme $x_1^m, \dots, x_m^m$, $w_1^m, \dots, w_m^m$ integrating $u_1, \dots, u_k$ 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 $u_1, \dots, u_k$ 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.

$\widetilde{x}_1, \dots, \widetilde{x}_n$ are known. Let $\widetilde{U}$ be the $n \times n$ matrix with entries given by

$$\widetilde{U}_{ij} = u_i(\widetilde{x}_j)\sqrt{\widetilde{w}_j}. \tag{3.6}$$

Then the operator $\widetilde{U}^{-1}$ maps the vector $(f(\widetilde{x}_1)\sqrt{\widetilde{w}_1}, \dots, f(\widetilde{x}_n)\sqrt{\widetilde{w}_n})$ to the vector of interpolation coefficients $c_1, \dots, c_n$ (see Definition 2.19). Furthermore, this operator is well-conditioned [3].

Remark 3.4. We note that the weights $\widetilde{w}_1, \dots, \widetilde{w}_n$ produced by Algorithm 3.3 can be negative. However, the algorithm still produces a stable interpolation scheme in the sense of Remark 3.3, the only difference being that the square roots of the weights can be complex.

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 $f_1, \dots, f_{2n} : [a, b] \to \mathbb{R}$ which are square integrable with respect to a weight function ω, it returns a set of n nodes $x_1, \dots, x_n$ and weights $w_1, \dots, w_n$ which integrate the functions $f_1, \dots, f_{2n}$ with precision ϵ.

It proceeds by using Algorithm 3.3 to construct a generalized Chebyshev quadrature for $f_1, \dots, f_{2n}$ and then eliminating n nodes one at a time. Each time it removes a node the algorithm performs a nonlinear optimization on the locations of the nodes to ensure that the functions $f_1, \dots, f_{2n}$ are still integrated with precision ϵ. The resulting quadrature scheme is guaranteed to integrate the collection of functions with precision ϵ.

Observation 1. 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 2.11, as well as a quadrature scheme which integrates the kernels K and $\tilde{K}$ given by Definition 2.9 times the densities near the corner.

The second part is the discretization of the integral equations (2.10) and (2.11) via a Nyström scheme using the quadratures and interpolation scheme constructed in the previous part.

4.1. Corner discretization.

In this section we construct a discretization of the basis functions in (2.32). Our algorithm uses a technical observation involving the roots of certain singular functions.

4.1.1. Principal observations.

The following observation establishes the rank of a certain collection of functions appearing in Theorem 2.11, and is used in the discretization of the functions representing the solution to (2.10) and (2.11) in the vicinity of corners.

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

$$\mathcal{F} = \{t^{\mu} \mid \mu = 0 \cup [1/2, 40]\,\}. \tag{4.1}$$

For $\epsilon = 10^{-16}$ the ϵ-rank of this collection of functions is 33 (see Definition 2.24). 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 3.2. All singular values after the 33rd are less than 10^{-16}.

Observation 3. We choose the upper limit of our range of μ in (4.1) 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 [16]).

Observation 4. In practice, if Algorithm 3.2 is used to compress finite collections of powers, t^{μ_i}, where $\mu_i \in [0,50]$, $i = 1, \dots, N$, $t \in [0,1/2]$, then the $(k+1)$st singular

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

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

function has exactly k roots. The resulting roots, $r_1, \dots, r_k$, together with the first k singular functions, $u_1, \dots, u_k$, 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 $(k+1)$st singular value.

4.1.2. Algorithm. 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 Observation 2. 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 $s_1, \dots, s_{N_D}$, a set of discretization weights $w_1, \dots, w_{N_D}$, and an orthonormal set of interpolation functions $u_1, \dots, u_{N_D} : [a, b] \to \mathbb{R}$, which interpolate the monomials t^μ, $\mu = 0 \cup [\mu_0, \mu_1]$, with precision ϵ_{quad} on the interval $[a, b]$.

Remark 4.1. 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 $s_i \in (10^{-16}, \frac{1}{2}]$ $i = 1, \dots, 33$, a corresponding set of (positive) weights w_i, $i = 1, \dots, 33$, and a set of orthonormal interpolating functions $u_i : (10^{-16}, \frac{1}{2}] \to \mathbb{R}$, $i = 1, \dots, 33$. 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.

4.2. Quadrature in the vicinity of corners. In light of Lemma 2.10, in order to integrate the kernels K and $\tilde{K}$ times density, it will also be necessary to approximate

Algorithm 4.1

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 $\mu_0 \le \nu_j \le \mu_1$ for $j = 2, \dots, N$.

Step 2. Use Algorithm 3.2 to compute the SVD of the set of functions $f_j(t) = t^{\nu_j}$, $j = 1, \dots, N$, $a \le t \le b$, with precision ϵ_{svd}, and obtain a set of singular values $\sigma_1 \ge \sigma_2 \ge \cdots \ge \sigma_M$ and corresponding singular functions $u_1, \dots, u_M$.

Step 3. Let k denote the number of singular values greater than ϵ_{quad}. Compute the roots of the $(k+1)$st singular value and denote them by $r_1, \dots, r_s$ (see Observation 4). If $s \neq k$ then Algorithm 3.3 should be used instead to determine suitable points $r_1, \dots, r_k$.

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

$$U_{ij} = u_i(r_j), \tag{4.2}$$

and the vector Y of length k with entries

$$Y_i = \int_a^b u_i(x)\,\mathrm{d}x. \tag{4.3}$$

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

integrals of the form

$$\int_0^{\frac{1}{2}} \frac{s_i \sin(\pi\alpha)}{s_i^2 + t^2 - 2 s_i t \cos(\pi\alpha)} t^{\mu} \mathrm{d}t, \tag{4.4}$$

where s_i, $i = 1, \dots, N_D$ are the discretization nodes constructed by Algorithm 4.1, and $\mu = i/\alpha, i/(2-\alpha)$, $i = 0, 1, \dots, M$. The integer M is a cutoff 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 (4.4) 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 $t_1^{(i)}, \dots, t_{N_i}^{(i)}$ and weights $w_1^{(i)}, \dots, w_{N_i}^{(i)}$ which integrate (4.4) for all $\alpha \in [\alpha_0, \alpha_1]$.

For each angle $\pi\alpha$, by combining the results of Algorithms 4.1 and 4.2, we construct an $N_D \times N_D$ matrix K^{α} which maps the density near a corner (evaluated at the discretization nodes $s_1, \dots, s_{N_D}$) 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 4.2. *Let $u_1, \dots, u_{N_D}$ be the functions produced by Algorithm* 4.1 *and $s_1, \dots, s_{N_D}$ and $w_1, \dots, w_{N_D}$ be the corresponding nodes and weights, respectively.*

Algorithm 4.2

Step 1. Sample the range $[\alpha_0, \alpha_1]$ at K points $a_1, \dots, a_K \in [\alpha_0, \alpha_1]$. 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 3.4 to obtain a Gaussian or near-Gaussian quadrature for the functions

$$\frac{s_i \sin(\pi a_k)\sqrt{w_i}}{s_i^2 + t^2 - 2 s_i t \cos(\pi a_k)} t^{j/a_k}, \tag{4.5}$$

$$\frac{s_i \sin(\pi a_k)\sqrt{w_i}}{s_i^2 + t^2 - 2 s_i t \cos(\pi a_k)} t^{j/(2-a_k)}, \tag{4.6}$$

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

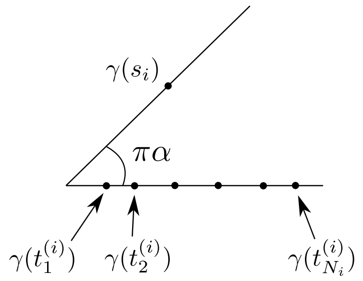


FIG. 3. *Quadrature and discretization nodes near a corner.*

Let U_{ij} denote the $N_D \times N_D$ matrix with entries

$$U_{ij} = u_i(s_j)\sqrt{w_j}. \tag{4.7}$$

Define the $N_D \times N_D$ matrix K_{ij}^{α} by

$$K_{ij}^{\alpha} = \sqrt{w_i} \sum_{k=1}^{N_i} K\big(\gamma(s_i), \gamma(t_k^{(i)})\big) w_k^{(i)} \left(\sum_{\ell=1}^{N_D} u_\ell(t_k^{(i)}) \, (U^{-1})_{\ell j} \right), \tag{4.8}$$

where $t_i^{(k)}, w_i^{(k)}$, $i = 1, \dots, N_D$, $k = 1, \dots, N_i$, are the nodes and weights, respectively, generated by Algorithm 4.2 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.

Remark 4.3. We note that for ρ in the span of $u_1, \dots, u_{N_D}$ the sum

$$\sum_{j=1}^{N_D} K_{ij}^{\alpha} \rho(s_j) \sqrt{w_j} \tag{4.9}$$

approximates the integral

$$\sqrt{w_i} \int_0^{\frac{1}{2}} K(\gamma(s_i), \gamma(t)) \, \rho(t) \, \mathrm{d}t, \tag{4.10}$$

where

$$\rho(t) = \sum_{\ell=1}^{N_D} u_\ell(t) \sum_{j=1}^{N_D} (U^{-1})_{\ell j}\, \rho(s_j) \sqrt{w_j}. \tag{4.11}$$

Remark 4.4. 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 4.5. The interpolation scheme constructed via Algorithm 4.1 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 4.2 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 $(\frac{\pi}{32}, (2 - \frac{1}{32})\pi)$.

4.3. Nyström discretization. In this section we discretize the boundary integral equations (2.10) and (2.11) using a Nyström method. In the following we restrict our attention to the interior Dirichlet problem (2.10); 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 (2.10) with precision ϵ. We proceed by enforcing equality at the discretization nodes, which yields the system of equations

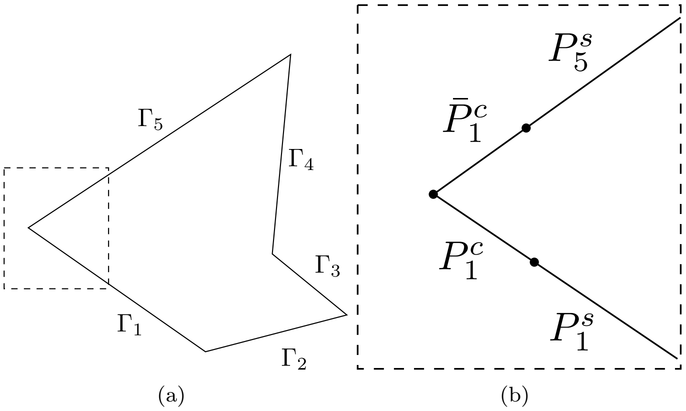
$$g(s_i)\sqrt{w_i} = -\pi \rho(s_i)\sqrt{w_i} + \sqrt{w_i} \int_0^L K(\gamma(s_i), \gamma(t))\, \rho(t)\, \mathrm{d}t, \quad i = 1, \ldots, N. \tag{4.12}$$

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 4(b), $\bar{P}_1^c = P_5^c$.

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 noncorner 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 4.1.



(a)

(b)

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

Next, to discretize (4.12) we require a quadrature scheme which approximates integrals of the form

$$
\begin{aligned}
\int_0^L K(\gamma(s_i),\gamma(t))\,\rho(t)\,\mathrm{d}t &= \sum_{j=1}^{N_c}\int_{P_j^c} K(\gamma(s_i),\gamma(t))\,\rho(t)\,\mathrm{d}t\\
&\quad+\sum_{j=1}^{N_s}\int_{P_j^s} K(\gamma(s_i),\gamma(t))\,\rho(t)\,\mathrm{d}t, \quad i=1,\dots,N. \qquad (4.13)
\end{aligned}
$$

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 4.2. 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 4.1 and the ranges of powers given in Remark 4.5, the result is a 33×33 matrix whose entries are given by Lemma 4.2.

The discretization procedure is summarized in the following algorithm. Here we assume our region Γ is divided into K line segments, $\Gamma_1,\dots,\Gamma_K$ with lengths $\ell_1,\dots,\ell_K$, respectively. We denote the unit tangent to Γ_j by (h_{xj},h_{yj}), $j=1,\dots,K$. Additionally, we denote the corners of Γ by $C_1,\dots,C_K$, 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}).

Remark 4.6. For ease of exposition the algorithm we describe uses a single Gauss–Legendre panel per edge. In practice it is often beneficial to divide P_i^s into a collection of Gauss–Legendre panels, where the sizes of the subpanels are repeatedly halved approaching the corner panels. Typically no more than two or three such bisections are required.

4.4. Evaluation of the solution to the differential equations. Given the solution ρ to the boundary integral equation (2.10) or (2.13), the solution to the

Algorithm 4.3

Discretization:

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

$$r_i = \sigma \min\{L_{i-1}, L_i\}, \tag{4.14}$$

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 $(s_1, \dots, s_{N_D}, w_1, \dots, w_{N_D})$ produced by Algorithm 4.1. Specifically, P_j^c has N_D nodes located at $(x_i^{(j)}, y_i^{(j)})$ with weight $w_i^{(j)}$ defined by

$$\left(x_i^{(j)}, y_i^{(j)}\right) = (c_{xj}, c_{yj}) + r_j s_i \left(h_{xj}, h_{yj}\right), \tag{4.15}$$

$$w_i^{(j)} = w_i r_j \tag{4.16}$$

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

$$\left(\bar{x}_i^{(j)}, \bar{y}_i^{(j)}\right) = \left(c_{x,j}, c_{y,j}\right) - r_j s_i \left(h_{x,(j-1)}, h_{y,(j-1)}\right), \tag{4.17}$$

$$\bar{w}_i^{(j)} = w_i r_j \tag{4.18}$$

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

Step 3. For each edge Γ_j, discretize the remainder P_j^s (i.e., 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 $x_1, \dots, x_N$, $w_1, \dots, w_N$ 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 as follows. If $x_i \notin P_k^c$ or $x_j \notin P_k^c$ for all $k = 1, \dots, K$, then

$$X_{ij} = -\pi \delta_{ij} + K(x_i, x_j)\sqrt{w_i w_j}, \tag{4.19}$$

where δ_{ij} is the , delta, and K is the kernel function given in Definition 2.9. If x_i is the sth discretization node of P_j^c (where $s = 1$ corresponds to the closest node to the corner) and x_j is the rth discretization node of the adjacent corner panel $\bar{P}_j^c$ (where $r = 1$ corresponds to the closest node to the corner on the opposite side) then set

$$X_{ij} = (K^\alpha)_{sr}, \tag{4.20}$$

where K^α is the $N_D \times N_D$ matrix constructed in Lemma 4.2.

boundary value problem (2.1) or (2.2), respectively, is obtained by evaluating the integral

$$u(x) = \int_0^L \frac{\langle \gamma(t) - x, \nu(t) \rangle}{\|\gamma(t) - x\|^2}\, \rho(t)\, \mathrm{d}t \tag{4.21}$$

for any $x \in \mathbb{R}^2 \setminus \Gamma$. When the target point x is sufficiently far from the boundary this integral can be evaluated by the sum

$$u(x) = \sum_{j=1}^{N} \frac{\langle x_j - x, \nu(x_j)\rangle}{\|x_j - x\|^2}\, \rho(x_j)\, w_j, \tag{4.22}$$

where $x_1, \ldots, x_N$ are the discretization nodes and $w_1, \ldots, w_N$ are the corresponding weights constructed in section 4.3. This approximation is accurate since the discretization nodes and weights were constructed so that, together with the corresponding weights, they integrate the density multiplied by smooth functions.

When the target point x is evaluated close to the boundary, though the integrand in (4.21) remains smooth, the integral (4.21) must be evaluated by other means due to the near-singular nature of the integrand. For the Dirichlet problem the density ρ can be interpolated to an appropriate set of quadrature nodes and weights which accurately compute integrals of the form (4.21) for x near Γ.

Similarly, for the Neumann problem, given a solution ρ to the boundary integral equation (2.15) or (2.17), the solution to the boundary value problem (2.3) or (2.4), respectively, is obtained by evaluating the integral

$$u(x) = \int_0^L \log \|\gamma(t) - x\|\, \rho(t)\, \mathrm{d}t \tag{4.23}$$

for any $x \in \mathbb{R}^2 \setminus \Gamma$. When the target point x is sufficiently far from the boundary this integral can be evaluated by the sum

$$u(x) = \sum_{j=1}^{N} \log \|x_j - x\|\, \rho(x_j)\, w_j, \tag{4.24}$$

where $x_1, \ldots, x_N$ are the discretization nodes and $w_1, \ldots, w_N$ are the corresponding weights constructed in section 4.3. Lemma 2.27 guarantees that the discretization nodes and weights constructed for the Dirichlet problem will accurately compute (4.23) for all points x sufficiently far away from the boundary Γ. If the Neumann problem is solved using the approach in Lemma 2.27, then in the vicinity of the corner the density cannot be interpolated accurately in the same manner as for the Dirichlet problem. A more involved procedure which allows evaluation of the solution of the boundary value problem near corners will be described at a later date.

5. Numerical results. To solve the integral equations of potential theory corresponding to both the interior and exterior problems with Dirichlet and Neumann boundary conditions 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 5–11. The algorithm was implemented in Fortran 77 and the experiments were run on a 2.7 GHz Apple laptop with 8 Gb RAM.

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 unit strength dipole placed *outside* the region. We solve the linear system to obtain the potential and use (2.9) 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 (2.10) and (2.13), which we then use to construct the solution to

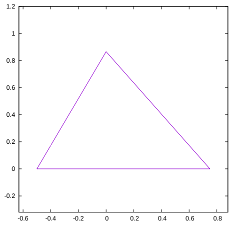


Fig. 5. Γ_1: *a triangle in* $\mathbb{R}^2$.

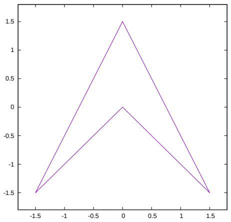


Fig. 6. Γ_2: *a chevron in* $\mathbb{R}^2$.

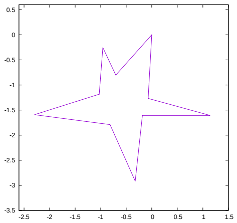


Fig. 7. Γ_3: *a star in* $\mathbb{R}^2$ *with* 10 *corners.*

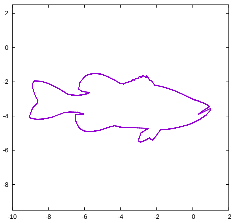


Fig. 8. Γ_4: *a fish in* $\mathbb{R}^2$ *with* 128 *corners.*

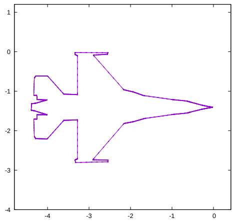


Fig. 9. Γ_5: *a plane in* $\mathbb{R}^2$ *with* 51 *corners.*

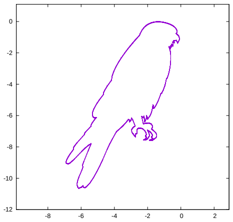


Fig. 10. Γ_6: *a falcon in* $\mathbb{R}^2$ *with* 228 *corners.*

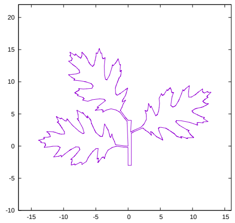


Fig. 11. Γ_7: *an oak branch in* $\mathbb{R}^2$ *with* 372 *corners.*

the boundary value problems (2.1) and (2.2) 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

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

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 3
Numerical results for the interior (I) and exterior (E) Neumann problems.

Boundary conditions	Curve	Number of nodes	Maximum absolute error	Precomputation time	Total run time	Condition number
E	Γ_1	294	$7.77 \cdot 10^{-16}$	0.2290	0.2374	16.36
I	Γ_1	390	$7.77 \cdot 10^{-16}$	0.2326	0.2517	14.52
E	Γ_2	648	$5.33 \cdot 10^{-15}$	0.2026	0.3086	101.7
I	Γ_2	648	$5.33 \cdot 10^{-15}$	0.2098	0.3337	60.89
E	Γ_3	1300	$2.55 \cdot 10^{-15}$	0.2389	1.2700	80.66
I	Γ_3	1300	$1.67 \cdot 10^{-15}$	0.2697	1.4899	129.3
E	Γ_4	20736 (2757)	$2.35 \cdot 10^{-12*}$	1.124	62.590	130.9
I	Γ_4	20736 (2755)	$1.65 \cdot 10^{-10*}$	1.187	61.945	285.0

described in [18] 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. Precomputation time refers to the time required to load the discretization nodes, quadratures, and singular functions; the time required to obtain the interpolation matrices U and U^{-1}; and the time to generate the discretization.

Similarly, to demonstrate the accuracy of our algorithm for the interior Neumann problem, we choose our boundary data g to be the result of an incident charge placed outside the region. We solve the linear system to obtain the potential and use (2.8) to construct the solution away from the boundary. Similarly, for the exterior Neumann problem, we choose our boundary data g to be the result of an incident charge placed inside the region. The linear system is solved to obtain the solution to the integral equations (2.15) and (2.17), which we then use to construct the solution to the boundary value problems (2.3) and (2.4) away from the boundary. In both cases an analytic solution exists (it is just the potential produced by the charge) 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 3. 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 [18] 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.

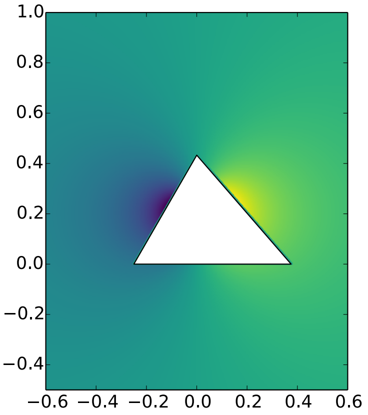


FIG. 12. *Solution to the external Dirichlet problem with data generated by a unit strength dipole at* $(0, 0.2)$ *with orientation* $(1, 0)$.

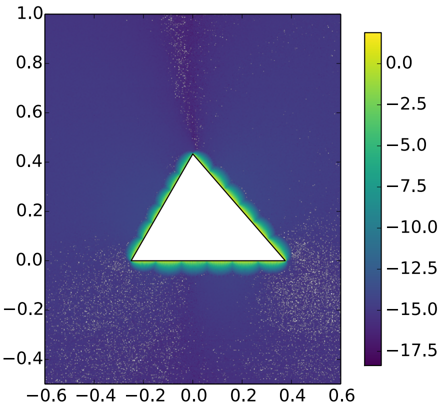


FIG. 13. *Absolute error* $(\log_{10})$ *in the solution of the external Dirichlet problem with data generated by a unit strength dipole at* $(0, 0.2)$ *with orientation* $(1, 0)$.

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.

Figures 12 and 13 show the solution and its error $(\log_{10})$ for the Dirichlet problem in the exterior of a triangle with vertices at $(-0.5, 0)$, $(0.75, 0)$, and $(0, \frac{\sqrt{3}}{2})$. The boundary data are generated by an internal unit strength dipole located at $(0, 0.2)$ oriented parallel to the x-axis. The errors near the boundary are due to the use of discretization nodes as quadrature nodes. They could be eliminated by interpolating to a finer mesh when evaluating close to a panel. Note that the corner panels behave in a way which is qualitatively similar to the Gaussian panels despite the singularities. Similarly, Figures 14 and 15 show the solution and its error $(\log_{10})$ for the Dirichlet problem in the interior of a quadrilateral with vertices at $(-1.5, -1.5)$, $(0, 0)$, $(1.5, -1.5)$, and$(0, 1.5)$. The boundary data are generated by an external unit strength dipole located at $(0.5, 0.5)$ oriented in the direction $(\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}})$.

As a further illustration of the algorithm, we apply the approach of this paper to calculate the potential 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. For the Neumann problem a point charge is used instead of a dipole. The boundary conditions are chosen so that the solution vanishes on the boundary of Ω. One level of the compression algorithm described in [18] is used to reduce the number of unknowns in the final linear system. The total field is shown in Figures 16, 17, and 18; results are given in Table 4. Due to the singular nature of the field at the source in Figure 17 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 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 4 correspond to the condition numbers of the compressed systems.

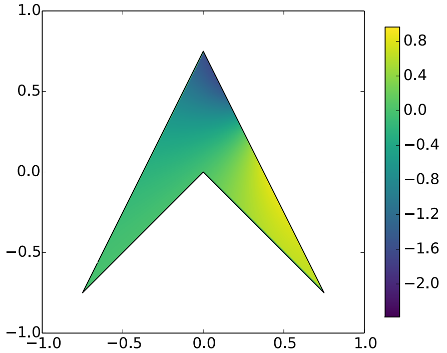


Fig. 14. *Solution to the internal Dirichlet problem with data generated by a unit strength dipole at* $(0, 0.2)$ *with orientation* $(1, 0)$.

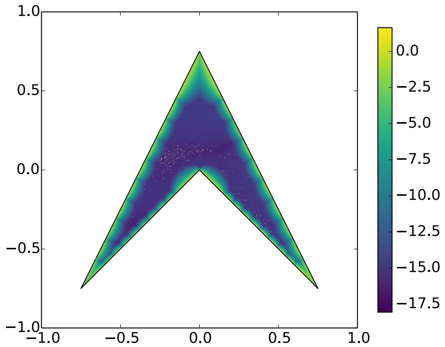


Fig. 15. *Absolute error* ($\log_{10}$) *in the solution of the internal Dirichlet problem with data generated by a unit strength dipole at* $(0, 0.2)$ *with orientation* $(1, 0)$.

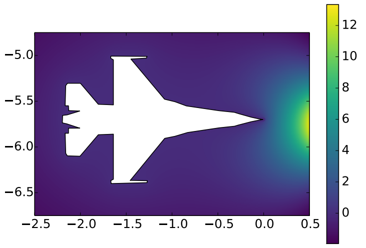


Fig. 16. *Solution for the scattering of a dipole from a plane in* $\mathbb{R}^2$.

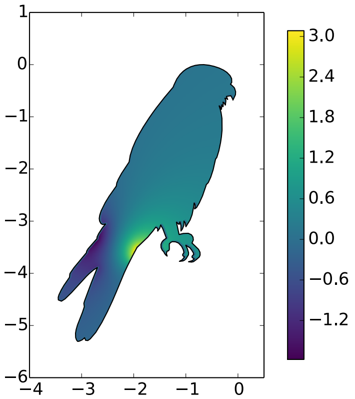


Fig. 17. *Solution for the interior scattering from a dipole for a falcon in* $\mathbb{R}^2$.

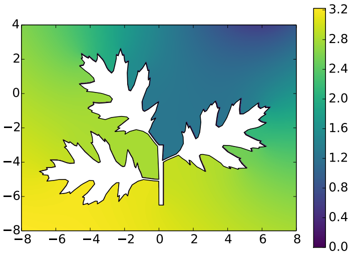


Fig. 18. *Solution for the scattering of a point from a branch in* $\mathbb{R}^2$.

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 and Neumann boundary conditions. The approach is based on the

TABLE 4
Numerical results for the interior (ID) and exterior (ED) Dirichlet problems and the exterior Neumann problem (EN).

Boundary conditions	Curve	n	N_c	N_d	T_0	T_s	Condition number
ED	Γ_5	51	725	16422	78.86	16.168	355.70
ID	Γ_6	228	2523	29640	58.16	38.438	301.07
EN	Γ_7	372	4025	60408	251.64	225.44	5458.8

observation in [24] 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. Curved boundaries with corners. While this paper only deals with the solution of Laplace's equation on polygonal domains, in [23] this analysis is extended to curved boundaries with corners. The numerical apparatus for solving Laplace's equation on regions of this type will be described in a future paper.

6.2. 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.

6.3. The Helmholtz equation and biharmonic equation and Maxwell's equations on domains with 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 [25] and the biharmonic equation [22] on polygonal domains. 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, Cambridge, 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] J. BREMER, *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. Amer. 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, Baltimore, MD, 1996.

[9] P. Grisvard, *Elliptic Problems in Nonsmooth Domains*, Pitman Advanced Publication Program, Boston, 1985.

[10] S. Hao, A. H. Barnett, P. G. Martinsson, and P. Young, *High-order accurate methods for Nyström discretization of integral equations with weakly singular kernels on smooth curves in the plane*, Adv. Comput. Math., 40 (2014), pp. 245–272.

[11] J. Helsing, *Integral equation methods for elliptic problems with boundary conditions of mixed type*, J. Comput. Phys., 228 (2009), pp. 8892–8907.

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

[13] J. Helsing and T. Johansson, *Fast reconstruction of harmonic functions from Cauchy data using integral equation techniques*, Inverse Probl. Sci. Eng., 18 (2010), pp. 381–399.

[14] J. Helsing and R. Ojala, *Corner singularities for elliptic problems: Integral equations, graded meshes, quadrature, and compressed inverse preconditioning*, J. Comput. Phys., 227 (2008), pp. 8820–8840.

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

[16] R. R. Lederman and V. Rokhlin, *On the Analytical and Numerical Properties of the Truncated Laplace Transform II*, Technical report, Yale University, New Haven, CT, 2015.

[17] R. S. Lehman, *Development of the mapping function at an analytic corner*, Pacific J. Math., 7 (1957), pp. 1437–1449.

[18] P. Martinsson and V. Rokhlin, *A fast direct solver for boundary integral equations in two dimensions*, J. Comput. Phys., 205 (2005), pp. 1–23.

[19] P.-G. Martinsson, V. Rokhlin, and M. Tygert, *On interpolation and integration in finite-dimensional spaces of bounded functions*, Commun. Appl. Math. Comput. Sci., 1 (2006), pp. 133–142.

[20] V. G. Maz'ya and S. M. Nikol'skii, *Analysis* IV: *Linear and Boundary Integral Equations*, (Encyclopedia Math. Sci.), Springer, New York, 1991.

[21] R. Ojala, *A robust and accurate solver of Laplace's equation with general boundary conditions on general domains in the plane*, J. Comput. Math., 30 (2012), pp. 433–448.

[22] M. Rachh and K. Serkh, *On the Solution of Stokes Equation on Regions with Corners*, arXiv:1711.04072, 2017.

[23] K. Serkh, *On the Solution of Elliptic Partial Differential Equations on Regions with Corners* III: *Curved Boundaries*, manuscript.

[24] K. Serkh, *On the solution of elliptic partial differential equations on regions with corners* II: *Detailed analysis*, Appl. Comput. Harmon. Anal., 46 (2019), pp. 250–287.

[25] K. Serkh and V. Rokhlin, *On the solution of the Helmholtz equation on regions with corners*, Proc. Natl. Acad. Sci. USA, 113 (2016), pp. 9171–9176.

[26] 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.

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

[28] 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*, Prikl. Mat. Mekh., 48 (1984), pp. 120–124.