# EFFICIENT DISCRETIZATION OF LAPLACE BOUNDARY INTEGRAL EQUATIONS ON POLYGONAL DOMAINS

JAMES BREMER AND VLADIMIR ROKHLIN

Abstract. We describe a numerical procedure for the construction of quadrature formulae suitable for the efficient discretization of boundary integral equations over very general curve segments. While the procedure has applications to the solution of boundary value problems on a wide class of complicated domains, we concentrate in this paper on a particularly simple case: the rapid solution of boundary value problems for Laplace's equation on two-dimensional polygonal domains. We view this work as the first step toward the efficient solution of boundary value problems on very general singular domains in both two and three dimensions. The performance of the method is illustrated with several numerical examples.

## 1. Introduction

One of the standard approaches to the numerical solution of boundary value problems for elliptic partial differential equations calls for converting them into integral equations, discretizing the integral equations via the Nyström method, and inverting the resulting discrete systems using either a fast direct solver or with the combination of an iterative method and the appropriate fast multipole method. In the case of a planar domain $\Omega$ with boundary $\partial\Omega$, the Nyström discretization of the integral equations, which take the form

$$\lambda\sigma(x) + \int_{\partial\Omega} K(x,y)\sigma(y)dy = u(x), \tag{1.1}$$

is typically effected by representing the unknown functions as piecewise polynomials. That is, it is assumed that the desired solution $\sigma$ can be represented locally by polynomials and the integral

$$\int_{\partial\Omega} K(x,y)\sigma(y)dy$$

is approximated using quadratures for functions of the form $K(x,y)P(y)$, with $P$ a polynomial of a given order. The efficiency of the resulting discretization hinges on both the suitability of the representation of solutions by piecewise polynomials and the number of nodes in the quadrature formulae.

On smooth domains, the representation of solutions $\sigma$ by piecewise polynomials is generally adequate. However, for nonsmooth domains $\Omega$, solutions of the integral equations can exhibit singular behaviors, making their representation via polynomials extremely inefficient. An obvious remedy is to devise improved representations for functions satisfying an integral equation of the form (1.1) near a singular point on the boundary curve $\partial\Omega$ and to construct the necessary quadratures for Nyström discretization. Analytical estimates of the singularities of solutions of certain boundary integral equations near a corner point are available in some cases (see, for instance, [10, 11]) and could be used to develop efficient local representations. The disadvantages of such an approach, however, are clear: the need for complicated analytical estimates of the singularities of solutions, the fact that individual cases must be treated separately, and the impossibility of treating cases in which estimates are lacking.

*Date*: December 11, 2009.

In this paper, we describe a numerical procedure for the construction of an orthonormal basis of functions spanning the space of restrictions of functions $\sigma$ satisfying a boundary integral equation

$$\lambda\sigma(x) + \int_\Gamma K(x,y)\sigma(y)dy = u(x) \tag{1.2}$$

on a contour $\Gamma$ to a small curve segment $\Gamma_0 \in \Gamma$. The resulting basis can be used to form quadrature rules for the Nyström discretization of the boundary integral equation (1.2) over $\Gamma_0$ with the number of quadratures nodes depending only on the rank of the basis. In other words, given a particular curve segment $\Gamma_0$, it is possible to construct numerically an efficient "purpose-made" quadrature for the discretization of a boundary integral equation over $\Gamma_0$. The principal step of the procedure consists of computing solutions of the restriction of the integral equation (1.2) to the curve segment $\Gamma_0$ for a small collection of right-hand-sides. In effect, the problem of efficiently discretizing an integral equation over a complex curve segment is reduced to the problem of solving the integral equation locally on that curve segment.

This procedure allows for a divide-and-conquer approach to the solution of boundary integral equations on complicated domains. For instance, given a domain $\Omega$ with boundary $\partial\Omega$ containing corner points $x_1, \ldots, x_n$, the procedure of this paper can be used to construct a collection of $n$ efficient quadrature formulae, one for the discretization of $\partial\Omega$ near each corner point $x_n$. The resulting quadrature formulae can then be used to produce what amounts to a compressed representation of the integral equation over the entire contour $\partial\Omega$. This has obvious applications to parallelization and in environments where the cost of inverting an integral equation that has been discretized as an $n \times n$ linear system is asymptotically greater than $O(n)$. It is also a viable approach to the solution of a problem too large to fit in available memory. This last application is expected to be important in the case of boundary value problems on complicated surfaces in three dimensions.

A particularly effective application of this procedure, and the focus of this paper, is the computation of collections of specialized quadrature rules for the efficient discretization of certain classes of pathological domains. In this paper, we describe the construction of quadrature formulae for efficiently discretizing Laplace boundary integral equations over two-dimensional polygonal domains. Once such quadrature rules have been constructed they can be used repeatedly to efficiently discretize Laplace boundary integral equations on such domains *without additional computations.* We refer to such a collection of quadrature formulae as a set of "universal quadratures" for polygonal domains. This example have been chosen by the authors as the most obvious and straightforward application; it is expected that the construction of universal quadratures will be possible in the case of much more general nonsmooth domains.

The approach of this paper is in marked contrast to most previously published algorithms, which involve the use of dense meshes of discretization nodes near corner points (see [16, 5, 2] for representative examples). Not only does this lead to large discrete systems of equations on domains with many corners, but the presence of densely sampled regions of a boundary curve interferes with the efficient operation of fast solvers. The recent paper [13] of Helsing and Ojala is notable in that it overcomes many of the drawbacks associated with using dense meshes of discretization nodes near corner points. In particular, it introduces a technique dubbed "recursive compressed inverse preconditioning" whereby a boundary integral equation is multiplied on the right by a preconditioner which smoothes singular solutions near corner points, thus rendering them more amenable to representation via polynomials. The hierarchical structure of the preconditioner is exploited in order to apply its inverse rapidly. Both the approach of Helsing-Ojala and the algorithm of this paper involve compressing subblocks of a discretized integral operator, but the algorithm of this paper differs in several key ways. Our approach involves a much simpler formalism which, in contrast to the recursive compressed inverse preconditioning scheme of [13], extends readily to the case of surfaces in three dimensions. Indeed, since we reduce the problem of constructing efficient quadrature rules for the discretization of a boundary integral equation to the problem of locally solving that integral equation, in most cases no additional machinery is required in order to apply our algorithm — whatever fast solver is already being used to invert the boundary integral equation can be used in the construction of quadrature rules. Finally, our approach has the advantage that for classes of domains for which universal quadratures can be constructed, essentially all complexity arising from the pathological behavior of the boundary is eliminated in the precomputation stage. That is, compression in that case is done *a priori* at the time of the quadrature precomputation instead of on-the-fly for a particular problem as in [13].

This paper is organized as follows. In Section 2, we discuss the preliminary mathematical and numerical methods which form the backbone of our approach. In Section 3, we review boundary integral methods for the solution of Laplace's equation on Lipschitz domains. In Section 4, we describe the discretization of those integral equations. Section 5 introduces the primary analytical tool of this paper, a procedure for the construction of bases spanning the set of restrictions of solutions of an integral equation to a small curve segment. In Section 6, a numerical algorithm for the construction of quadratures for the discretization of boundary integral equations on polygonal domains is described. In Section 7 the implementation of the algorithm is discussed and numerical examples are given. Finally, in Section 8, we discuss possible extensions and generalizations of the present work.

## 2. Preliminaries

### 2.1. **Interpolation on spaces of bounded functions.**

The following result, which ensures that a numerically stable interpolation scheme exists for any collection of bounded functions, appears as Theorem 2 in [20].

THEOREM 2.1. *Suppose that $S$ is an arbitrary set, $n$ is a positive integer, $f_1, \dots, f_n$ are bounded complex-valued functions on $S$, and $\epsilon$ is a positive real number such that*

$$\epsilon \leq 1.$$

*Then, there exist $n$ points $x_1, \dots, x_n$ in $S$ and $n$ functions $g_1, \dots, g_n$ on $S$ such that*

$$|g_k(x)| \leq 1 + \epsilon \tag{2.1}$$

*for all $x$ in $S$ and $k = 1, 2, \dots, n$, and*

$$f(x) = \sum_{k=1}^{n} f(x_k) g_k(x)$$

*for all $x$ in $S$ and any function $f$ defined on $S$ via the formula*

$$f(x) = \sum_{k=1}^{n} c_k f_k(x).$$

*Moreover, if the set $S$ is finite, then $g_1, \dots, g_n$ can be chosen so that (2.1) holds with $\epsilon = 0$.*

Remark 2.1. *The proof of Theorem 2.1 in* [20] *is constructive, but the procedure is computationally infeasible. Typically, however, the pivoted Gram-Schmidt procedure with reorthogonalization can be used in practice to identify a set of stable interpolation nodes for a finite collection of bounded functions on a finite set $S$. See* [6] *for a detailed discussion of interpolation and quadrature for very general classes of functions.*

### 2.2. **Generalized Chebyshev quadratures.**

Although Chebyshev quadratures are classical Gaussian quadratures on the interval $[-1, 1]$ with respect to the weight function $\omega(x) = (1 - x^2)^{-1/2}$, in practice, Chebyshev nodes and weights are often used to integrate functions on $[-1, 1]$ with respect to the weight function $\omega(x) = 1$. This practice leads to a $2n$-point quadrature which integrates exactly polynomials of order $2n - 1$, and motivates the following definition:

DEFINITION 2.1. *A quadrature formula will be referred to as a Chebyshev quadrature for a set of $2n$ linearly independent functions $\phi_1, \dots, \phi_{2n} : [a, b] \to \mathbb{R}$ if it consists of $2n$ nodes and $2n$ weights and integrates exactly the functions $\phi_i$, for all $i = 1, \dots, 2n$. The weights and nodes of a Chebyshev quadrature will be referred to as Chebyshev weights and nodes, respectively.*

The following lemma, which asserts that if a numerically stable solution for a system of linear equations exists then there also exists a numerically stable *basic* solution for that system of equations, is an immediate consequence of Theorem 2.1.

LEMMA 2.1. *If*

$$Ax = b, \tag{2.2}$$

*where $A$ is an $m \times n$ matrix of rank $m$, then there exists a vector $\tilde{x} \in \mathbb{R}^n$ with no more then $m$ nonzero entries such that*

$$A\tilde{x} = b,$$

*and*

$$\|\tilde{x}\|_1 \leq m\|x\|_1.$$

It follows from Lemma 2.1 that a numerically stable Chebyshev quadrature for a finite sequence of linearly independent functions $u_1, \ldots, u_k$ defined on an interval $[a, b]$ exists provided there exists a stable quadrature formula $x_1, x_2, \ldots, x_n, w_1, w_1, \ldots, w_n$ integrating the functions. The condition that the pre-existing quadrature rule integrates the functions $u_1, \ldots, u_k$ can be expressed by the matrix equation

$$\begin{pmatrix} u_1(x_1) & u_1(x_2) & \cdots & u_1(x_n) \\ u_2(x_1) & u_2(x_2) & \cdots & u_2(x_n) \\ \vdots & & \cdots & \vdots \\ u_k(x_1) & u_k(x_2) & \cdots & u_k(x_n) \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 \\ \vdots \\ r_k \end{pmatrix}, \tag{2.3}$$

where $r_i$, $i = 1, \ldots, k$, is defined by

$$r_i = \int_a^b u_i(x)\, dx.$$

Then, by Lemma 2.1, there exist $i_1, \ldots, i_n$ and $\tilde{w_1}, \ldots \tilde{w_k}$ such that

$$\begin{pmatrix} u_1(x_{i_1}) & u_1(x_{i_2}) & \cdots & u_1(x_{i_n}) \\ u_2(x_{i_1}) & u_2(x_{i_2}) & \cdots & u_2(x_{i_n}) \\ \vdots & & \cdots & \vdots \\ u_k(x_{i_1}) & u_k(x_{i_2}) & \cdots & u_k(x_{i_n}) \end{pmatrix} \begin{pmatrix} \tilde{w_1} \\ \tilde{w_2} \\ \vdots \\ \tilde{w_k} \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 \\ \vdots \\ r_k \end{pmatrix} \tag{2.4}$$

and

$$\sum_{j=1}^{k} |\tilde{w_j}| \leq k \sum_{j=1}^{n} |w_j|. \tag{2.5}$$

The points $x_{i_1}, \ldots, x_{i_k}$ are, of course, the nodes of a generalized Chebyshev quadrature for the functions $u_1, \ldots, u_k$ and the $\tilde{w_1}, \ldots, \tilde{w_k}$ are the corresponding weights.

The numerical computation of such a solution to the matrix equation (2.3) can be problematic; for instance, when $u_j(x) = x^{j-1}$, $j = 1, \ldots, k$, the matrix appearing on the left hand side of (2.3) is a Vandermonde matrix. However, there is a natural right preconditioner which usually makes the problem tractable. When the $u_j$ are orthonormal and the original quadrature rule has been chosen so as to integrate products of the $u_j$ (conditions which can usually be satisfied in practice), the rows of the matrix

$$\tilde{U} = \begin{pmatrix} u_1(x_1)\sqrt{w_1} & u_1(x_2)\sqrt{w_2} & \cdots & u_1(x_n)\sqrt{w_n} \\ u_2(x_1)\sqrt{w_1} & u_2(x_2)\sqrt{w_2} & \cdots & u_2(x_n)\sqrt{w_n} \\ \vdots & & \cdots & \vdots \\ u_k(x_1)\sqrt{w_1} & u_k(x_2)\sqrt{w_2} & \cdots & u_k(x_n)\sqrt{w_n} \end{pmatrix} \tag{2.6}$$

are orthonormal and the modified matrix equation

$$\tilde{U}x = b \tag{2.7}$$

has greatly enhanced numerical stability.

Remark 2.2. *A numerically stable basic solution to equation (2.7) can be obtained in practice by forming a rank-revealing QR decomposition of the matrix $\tilde{U}$ via the pivoted Gram-Schmidt algorithm with reorthogonalization. In the rare cases where that algorithm is unstable (and the authors have never encountered such a situation in practice), more recent algorithms for the construction of rank-revealing QR decompositions which are guaranteed to be stable could be substituted (see, for instance,* [12]*). See the monograph* [3] *for a detailed discussion of the numerical solution of underdetermined systems of linear equations, including the computation of basic solutions via the pivoted Gram-Schmidt algorithm.*

### 2.3. **Generalized Gaussian quadratures.**

In [18], the notion of a Gaussian quadrature was generalized as follows:

DEFINITION 2.2. *A quadrature formula will be referred to as Gaussian with respect to a set of $2n$ linearly independent functions $\phi_1, \ldots, \phi_{2n} : [a,b] \to \mathbb{R}$ if it consists of $n$ nodes and $n$ weights and integrates exactly the functions $\phi_i$, for all $i = 1, \ldots, 2n$. The weights and nodes of a Gaussian quadrature will be referred to as Gaussian weights and nodes, respectively.*

Several algorithms for the construction of generalized Gaussian quadratures have been proposed (see [18, 6, 23, 4]), all of which are based on the observation that the nodes $x_1, \ldots, x_m$ and weights $w_1, \ldots, w_m$ of a quadrature rule exact for the functions $f_1, \ldots, f_n$ satisfy the (generally underdetermined) system of nonlinear equations

$$\begin{aligned} G_1(x_1, \ldots, x_m, w_1, \ldots, w_m) &= b_1 \\ G_2(x_1, \ldots, x_m, w_1, \ldots, w_m) &= b_2 \\ &\vdots \\ G_n(x_1, \ldots, x_m, w_1, \ldots, w_m) &= b_n, \end{aligned}$$

where

$$G_i(x_1, \ldots, x_m, w_1, \ldots, w_m) = \sum_{j=1}^{m} f_i(x_j) w_j$$

and

$$b_i = \int_a^b f_i(x)\, dx.$$

The various methods of [18, 6, 23, 4] then amount to different numerical procedures for the determination of a sparse, stable solution to this underdetermined nonlinear system of equations. The most recent of these methods, [4], operates by starting from a generalized Chebyshev rule and reducing it point-by-point; that is, at each iteration, one point is deleted from an $(n+1)$-point quadrature formula exact for the collection of functions under consideration and the resulting $n$-point formula is refined using Newton iterations until it is sufficiently accurate. This procedure has been used to construct stable quadratures for very general classes of functions.

### 2.4. **Compression of sequences of functions.**

In this subsection, we discuss an analog of the singular value decomposition for sequences of functions. The following result, which can be found in [6], generalizes the SVD to this setting.

THEOREM 2.2. *Suppose that the functions $\phi_1, \ldots, \phi_m : [a,b] \to \mathbb{R}$ are square integrable. Then there exist an integer $k$, a finite orthonormal sequence of functions $u_1, \ldots, u_k : [a,b] \to \mathbb{R}$, an $m \times k$ matrix $V = (v_{ij})$ with orthonormal columns, and a sequence $s_1 \geq s_2 \geq \cdots \geq s_k > 0 \in \mathbb{R}$ such that*

$$\phi_j(x) = \sum_{i=1}^{k} u_i(x) s_i v_{ji} \tag{2.8}$$

*for all $x \in [a,b]$ and all $j = 1, \ldots, m$. The sequence $s_1, \ldots, s_k$ is uniquely determined by $k$.*

By analogy with the finite-dimensional case, we will refer to this decomposition as the SVD of a finite sequence of functions. We call the functions $u_i$ the singular functions, the columns of $V$ the singular vectors, and the values $s_i$ the singular values. The SVD is clearly a useful tool for the compression of the sequence $\phi_1, \ldots, \phi_m$: if we let $\tilde{\phi_j}(x)$ denote the $p$-term truncation

$$\tilde{\phi_j}(x) = \sum_{i=1}^{p} u_i(x) s_i v_{ji} \tag{2.9}$$

of the sum (2.8), then

$$\|\tilde{\phi_j}(x) - \phi_j(x)\|_2 \leq s_{p+1} \tag{2.10}$$

for $j = 1, \ldots, m$.

In order to form the singular values and vectors of a sequence $\phi_1, \ldots, \phi_m$ of functions numerically, a quadrature formula $x_1, \ldots, x_n, w_1, \ldots, w_n$ integrating products of the $\phi_i$ is required. In particular, given such a quadrature it is clear the singular values of the matrix $n \times m$ matrix $A$ with entries

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

are the singular values of the functions $\phi_1, \phi_2, \ldots, \phi_m$. Moreover, the $j$th singular vector of $A$ consists of the values of the $j$th singular function at the $n$ quadrature nodes $x_1, \ldots, x_n$ scaled by the square roots $\sqrt{w_1}, \ldots, \sqrt{w_n}$ of the quadrature weights. See [23] for a more detailed discussion of the numerical computation of the SVD of a collection of functions.

## 3. Boundary integral formulations

In this section, we briefly outline the solution of certain boundary value problems for Laplace's equation via integral equation methods. Thorough treatments of the classical theory can be found in [17, 21, 9, 14]. Extension of the classical theory to the case of Lipschitz domains is discussed in [15, 22, 7].

### 3.1. Boundary integral equations on smooth domains.

In this section, $\Omega$ will denote a bounded, simply-connected domain in the plane whose boundary $\partial\Omega$ is twice continuously differentiable, $\Omega^c$ will be the open region in the plane exterior to $\Omega$, and $dS$ will denote integration with respect to the arclength measure on $\partial\Omega$.

The *interior Dirichlet problem* calls for the determination of a function harmonic in $\Omega$ with prescribed values on the boundary curve $\partial\Omega$. That is, given a continuous function $f : \partial\Omega \to \mathbb{R}$, we seek $u : \Omega \to \mathbb{R}$ satisfying

$$\begin{aligned} \Delta u(x) &= 0 \text{ for } x \in \Omega \\ \lim_{\substack{x \to p \\ x \in \Omega}} u(x) &= f(p) \text{ for } p \in \partial\Omega. \end{aligned} \tag{3.1}$$

As is well-known, the unique solution can represented in the form of a potential arising from a dipole distribution $\sigma$ on $\partial\Omega$:

$$u(x) = \frac{1}{2\pi} \int_{\partial\Omega} \sigma(y) \frac{\partial}{\partial\nu_y} \log|x-y| \, dS(y), \tag{3.2}$$

where $\frac{\partial}{\partial\nu_y}$ denotes the outward normal derivative taken in the variable $y$. In particular, the function $u(x)$ defined by (3.2) is harmonic in $\Omega$ and the limit of $u(x)$ as $x$ approaches the point $p \in \partial\Omega$ from the interior of $\Omega$ is given by the jump relation

$$\lim_{\substack{x \to p \\ x \in \Omega}} u(x) = \frac{1}{2}\sigma(p) + \frac{1}{2\pi} \int_{\partial\Omega} \sigma(y) \frac{\partial}{\partial\nu_y} \log|p-y| \, dS(y). \tag{3.3}$$

It follows that if $\sigma(y)$ satisfies the integral equation

$$\frac{1}{2}\sigma(p) + \frac{1}{2\pi} \int_{\partial\Omega} \sigma(y) \frac{\partial}{\partial\nu_y} \log|p-y| \, dS(y) = f(p) \tag{3.4}$$

for all $p \in \partial\Omega$, then the function $u(x)$ given by (3.2) is a solution to problem (3.1).

Other boundary value problems for Laplace's equation can be solved in a similar fashion. In this paper, we will be concerned with the interior Dirichlet, exterior Dirichlet, exterior Neumann, and interior Neumann problems. The integral equation

$$-\frac{1}{2}\sigma(p) + \frac{1}{2\pi}\int_{\partial\Omega} \sigma(y)\frac{\partial}{\partial\nu_y}\log|p-y|\,dS(y) = f(p) \tag{3.5}$$

arises from the exterior Dirichlet problem (see [17]), the equation

$$-\frac{1}{2}\sigma(p) + \frac{1}{2\pi}\int_{\partial\Omega} \sigma(y)\left(\frac{\partial}{\partial\nu_y}\log|x-y| + 1\right)dS(y) = f(p). \tag{3.6}$$

arises from the exterior Neumann problem, and the integral equation

$$-\frac{1}{2}\sigma(p) + \frac{1}{2\pi}\int_{\partial\Omega} \sigma(y)\frac{\partial}{\partial\nu_p}\log|p-y|\,dS(y) - \sigma(p^*) = f(p) \tag{3.7}$$

is a typical mechanism for the solution of the interior Neumann problem (see [1]). Note that $p^*$ in equation (3.7) refers to an arbitrarily chosen point on the boundary $\partial\Omega$.

3.2. **Boundary integral equations on Lipschitz domains.** It is a well-known classical result that when the boundary curve $\partial\Omega$ is twice continuously differentiable, the integral operator

$$T\sigma(x) = \int K(x,y)\sigma(y)\,dS(y), \tag{3.8}$$

where $K$ is one of the potential theoretic kernels appearing in the preceding section, is compact as an operator $L^2(\partial\Omega) \to L^2(\partial\Omega)$. More recently, it was established in [8] that this is the case so long as the boundary curve is continuously differentiable. The invertibility of the various operators of the form $\pm 1/2I + T$ appearing in the preceding section then follows from the Fredholm theory; in particular, the operators arising from Dirichlet boundary conditions are invertible as operators $L^2(\partial\Omega) \to L^2(\partial\Omega)$ while the operators corresponding to Neumann conditions are isomorphisms $L^2_0(\partial\Omega) \to L^2_0(\partial\Omega)$, where $L^2_0(\partial\Omega)$ is the space of square integrable functions of zero mean on $\partial\Omega$. It is a relatively recent and deep result in analysis that the boundary operators $\pm 1/2I + T$ are still invertible in the case of domains $\Omega$ whose boundaries $\partial\Omega$ are merely Lipschitz. In that case, the integral operator (3.8) is no longer compact and the proof of the invertibility of $\pm 1/2I + T$ was first established in [22].

We assume now that our boundary curve $\partial\Omega$ is Lipschitz. Then the integral equations of the preceding section no longer hold everywhere, but rather only at points $\gamma \in \partial\Omega$ at which the boundary curve is differentiable. Of course, as is well-known, if $\partial\Omega$ is Lipschitz then it is differentiable almost everywhere (for instance, a Lipschitz function is absolutely continuous), and so in this case the integral equations of the preceding section hold almost everywhere.

Corrected formulas which hold everywhere can be derived in particularly simply situations, for instance on polygonal domains (see, for example, [17]), but *such estimates are unnecessary.* We are never interested in the pointwise behavior of a solution $\sigma$ of one of our boundary integral equations, but rather in the *distributional* behavior of a solution; i.e., the only objects of interest to us are layer potentials of the form

$$\int_{\partial\Omega} K(x,y)\sigma(y)\,dS(y).$$

That we are unable to resolve a charge *distribution* $\sigma$ pointwise is of no concern so long as the layer potential is unaffected — which is, of course, the case assuming $\sigma$ satisfies the correct boundary integral equation almost everywhere.

## 4. Discretization of integral equations

4.1. **The Nyström method.** The Nyström method is a well-known technique for the discretization of integral equations. It operates by replacing integrals with appropriately chosen quadrature formulae; i.e., via approximations of the form

$$\int K(x,y)\sigma(y)\,dS(y) \approx \sum_{l=1}^{M} K(x,y_l)\sigma(y_l)w_l. \tag{4.1}$$

Here we describe a very general Nyström framework for the discretization of the integral equations of Section 3. Recall that they are of the form

$$\lambda\sigma(x) + \int_\Gamma K(x,y)\sigma(y)\,dS(y) = u(x), \tag{4.2}$$

where $\Gamma$ is a closed curve in the plane, $\lambda$ is a real constant, and the kernel $K(x,y)$ is continuous except at points $y$ where the boundary curve fails to be differentiable.

We begin by assuming that $\Gamma$ is divided into $M$ curve segments, $\Gamma_1,\ldots,\Gamma_M$, not necessarily of equal length. For each curve segment $\Gamma_j$ we will require the following:

(1) An orthonormal collection of $k$ basis functions $\{\phi_i\}$ in $L^2(\Gamma_j)$,

(2) A linear interpolation scheme for the basis functions $\phi_1,\ldots,\phi_k$ with nodes $\lambda_1,\ldots,\lambda_k$,

(3) A "far" quadrature formula of the form

$$\int_{\Gamma_j} K(x,y)\sigma(y)\,dS(y) \approx \sum_{l=1}^{M} K(x,y_l)\sigma(y_l)w_l,$$

exact whenever $\sigma$ is in the span of the basis functions $\{\phi_i\}$ and $x$ is outside of $\Gamma_j$,

(4) A set of $k$ "diagonal" quadrature formulas of the form

$$\int_{\Gamma_j} K(x,y)\sigma(y)\,dS(y) \approx \sum_{l=1}^{M} K(x,y_l)\sigma(y_l)w_l$$

the jth of which is exact for $\sigma$ is in the span of the $\{\phi_i\}$ and $x=\lambda_j$.

The method proceeds under the assumption that the restriction of the unknown solution $\sigma$ in equation (4.2) to the curve segment $\Gamma_j$ can be represented as linear combinations of the basis functions $\{\phi_i\}$ for $\Gamma_j$. Let $\Gamma_j$ and $\Gamma_i$ be two curve segments, not necessarily distinct. We will denote by $\phi_1,\ldots,\phi_n$ the basis functions on the curve segment $\Gamma_j$, by $s_1,\ldots,s_n$ the interpolation nodes on the curve segment $\Gamma_j$, and by $t_1,\ldots,t_m$ the interpolation nodes on the curve segment $\Gamma_i$. The integral equation

$$T_{ij}\sigma(x) = u(x), \tag{4.3}$$

where $T_{ij}$ is the integral operator mapping functions on $\Gamma_j$ to functions on $\Gamma_i$ defined by

$$T_{ij}\sigma(x) = \int_{\Gamma_j} K(x,y)\sigma(y)\,dy \tag{4.4}$$

is then discretized by repeating the following sequence of steps for each interpolation node $t$ on $\Gamma_i$:

(1) The appropriate quadrature formula $x_1,\ldots,x_l,w_1,\ldots,w_l$ for functions of the form

$$K(t,s)\sigma_u(s), \quad u=1,\ldots,n,$$

is determined. That is, depending on the location of $t$ relative to the curve segment $\Gamma_j$, either the "far" quadrature rule or one of the diagonal quadrature rules is selected.

(2) The kernel $K(x,y)$ is evaluated at the points $(t,x_r)$ for $r=1,\ldots,l$ and the $1\times l$ vector $v$ with entries

$$v_r = K(t,x_r)w_r$$

is formed.

(3) The $1 \times l$ vector $v$ is multiplied on the right by the $l \times n$ matrix interpolating the basis functions $\phi_1, \dots, \phi_n$ from the interpolation nodes $s_1, \dots, s_n$ on $\Gamma_j$ to the quadrature nodes $x_1, \dots, x_l$.

(4) The entries $\{\alpha_s\}$ of the resulting $1 \times n$ vector give a single linear equation

$$\alpha_1 \sigma(s_1) + \alpha_2 \sigma(s_2) + \dots \alpha_n \sigma(s_n) = u(t)$$

constraining the values of the solution $\sigma$ at the nodes $s_1, \dots, s_n$.

The result of repeating this procedure for each of the $m$ interpolation nodes is an $m \times n$ linear system of the form

$$\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix} \begin{pmatrix} \sigma(s_1) \\ \sigma(s_2) \\ \vdots \\ \sigma(s_n) \end{pmatrix} = \begin{pmatrix} u(t_1) \\ u(t_2) \\ \vdots \\ u(t_m) \end{pmatrix}$$

discretizing the integral equation

$$T_{ij} \sigma = u.$$

Repeating the above procedure for each pair of curve segments $\Gamma_j$ and $\Gamma_i$, and accounting for the constant term in equation (4.2) results in a discrete system of $N$ equations in $N$ unknowns of the form

$$\lambda x + Ax = y,$$

where $N$ is equal to the sum of the number of interpolation nodes $n_j$ on each curve segment $\Gamma_j$ and $A$ is a matrix formed by concatenating the discrete matrices representing the $T_{ij}$.

Solving the amalgamated system yields the values of the unknown function $\sigma$ in (4.2) at the interpolation nodes of each of the curve segments $\Gamma_j$. The value of $\sigma$ at any point $x$ on $\Gamma$ can then be computed in $O(1)$ operations using the appropriate interpolation formula. Moreover, the value of a layer potential

$$u(x) = \int_\Gamma D(x, y) \sigma(y) \, dy$$

can be computed for any $x$ sufficiently far enough away from the curve $\Gamma$ in $O(N)$ operation using the far quadrature formulas for the curve segments $\Gamma_j$, $j = 1, \dots, M$. For points close to the curve, an adaptive Gaussian quadrature scheme which relies on the ability to evaluate the charge distribution at any point via interpolation can be used to compute the value of the layer potential.

### 4.2. **Quadratures for smooth curve segments.**

For a smooth curve segment $\Gamma_0$, we can construct the appropriate quadrature and interpolation formulae in the following manner. We start with a parameterization $r : [-1, 1] \to \Gamma_0$ of the curve segment and take as our basis the image of the Legendre polynomials of degree $(k-1)$ on $[-1, 1]$ under the mapping $r$. The well-known $k$-point Legendre interpolation scheme with nodes $t_1, \dots, t_k$ maps onto $\Gamma_0$ as well; i.e., we take the jth interpolation node on $\Gamma_0$ to be $x_j = r(t_j)$. Finally, we use, for each of the $(k+1)$ quadrature formulas, the image of the Legendre quadrature rule, which integrates exactly polynomials of degree $2k - 1$, under the mapping $r$.

For a proof of the convergence of the Nyström method for boundary integral equations with continuous or weakly singular kernels, see [17]. Precise error bounds are difficult to derive, but when both the unknown charge distribution $\sigma$ and the kernel $K(x, y)$ are $C^\infty$ — as is the case when the boundary curve is $C^\infty$ — this procedure achieves very rapid convergence. Indeed, the accuracy arising from this technique is typically limited by the approximation of the unknown function $\sigma$ by piecewise polynomials and in that case the order of convergence is generally $O(1/n^k)$ in the total number of discretization points $n$.

### 4.3. **Primitive discretization of corner regions.**

As noted in the last subsection, the discretization of the boundary integral equation

$$\lambda\sigma(y) + \int_\Gamma K(x,y)\sigma(y)\,dy = u(x) \tag{4.5}$$

via piecewise Gaussian quadrature formulae achieves very high rates of convergence so long as the kernel $K(x,y)$, the solution $\sigma(y)$, and the right hand side u(x) are all smooth. However, on domains with corner points — like that shown in Figure 1 — the integral kernels $K(x,y)$ from Section 3 are singular, the right-hand sides are not smooth at the corner point, and the solutions can exhibit any one of a number of different singular or discontinuous behaviors near the corner.

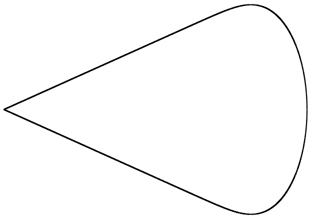


Figure 1: A domain with a single corner point.

In the precomputation stage of the algorithm of this paper, we shall have to solve boundary integral equations on domains with corners to very high precision. Standard approaches to discretizing an equation of the form (4.5) near a corner point $\gamma_0$ call for a dense mesh of points near $\gamma_0$ (see [16, 5, 2] for representative examples). We adopt the terminology of [13] and call a subdivision of the interval $[-1,1]$ into subintervals with endpoints

$$\frac{1}{2^j} \quad \text{and} \quad -\frac{1}{2^j} \quad \text{for } j = 0,1,2,\ldots,s$$

a *simply graded mesh.* The integral equation (4.5) can be discretized over a small contour $\Gamma$ containing the corner point $\gamma_0$ by first mapping $[-1,1]$ onto an interval around the corner $\gamma_0$, with 0 mapping to the corner $\gamma_0$. Gaussian quadrature formulas are then used to discretize the relevant integral equation over the image of each of the subintervals comprising the simply graded mesh on $[-1,1]$. Note that the resulting discretization omits a small region around the corner point and we will generally classify simply-graded meshes by this cutoff value.

Simply-graded meshes are a primitive tool and in some cases, they are not sufficient to resolve the solution of an integral equation to double precision accuracy (at least without performing computations in extended precision arithmetic). However, a simple and effective remedy is available. If $x_1,\ldots,x_n,w_1,\ldots,w_n$ denotes the quadrature obtained from a simply-graded mesh on the interval $[-1,1]$, then we let $\{y_j,v_j\}$ denote the quadrature rule obtained via a substitution of the form $y = x^{(2k+1)}$, where $k$ is a positive integer; that is,

$$y_j = x_j^{(2k+1)} \quad \text{and} \quad v_j = (2k+1)w_j^{(2k+1)}x_j^{2k}.$$

The image of the quadrature rule $\{y_j,v_j\}$ under a mapping onto the corner region can then be used in the discretization of the integral equation (4.5). In the authors' experience this simple modification allows for the solution of a boundary integral equation of the form (4.5) on a domain with a corner point to full double precision using double precision arithmetic. It also has the advantage of generally decreasing the number of discretization nodes required to obtain a desired accuracy.

## 5. Bases for restricted charge distributions

We now discuss the principal tool of this paper, a procedure for construction of an orthonormal basis of functions spanning the set of restrictions of charge distributions $\sigma$ satisfying a Laplace boundary integral equation

$$\lambda\sigma(x) + \int_\Gamma K(x,y)\sigma(y)dS(y) = u(x) \tag{5.1}$$

on a contour $\Gamma$ to a small neighborhood about a point $\gamma_0 \in \Gamma$. For the sake of brevity, we will restrict our attention to the solution of the interior Dirichlet problem via a double layer representation (as discussed in subsection 3.1). The other integral equations appearing in Section 3 are treated similarly.

5.1. **Bases for general curve segments.** In what follows, we shall fix a simply-connected domain $\Omega$ in the plane whose boundary $\Gamma$ is a compact, connected Lipschitz curve. Our aim is to produce a basis of functions spanning the set of restrictions of solutions $\sigma$ of the integral equation

$$\frac{1}{2}\sigma(x) + \int_\Gamma K(x,y)\sigma(y)\,dS(y) = u(x), \tag{5.2}$$

where $K(x,y)$ is the kernel

$$K(x,y) = \frac{\partial}{\partial \nu_y} \log|x-y|\,,$$

to a neighborhood of a point $\gamma_0 \in \Gamma$. We shall let $B_1$ be the disc of radius $r > 0$ centered at the point $\gamma_0$ and we shall let $B_2$ be the disc of radius $2r$ also centered at the point $\gamma_0$. Moreover, we will denote by $\Gamma_1$ the set formed from the intersection of $B_1$ and the contour $\Gamma$, by $\Gamma_2$ the intersection of $\Gamma$ and the annulus $B_2 \setminus B_1$, and by $\Gamma_3$ the portion of $\Gamma$ in the complement of $B_2$. Finally, we let $(r,\theta)$ be the usual polar coordinate system centered at the point $\gamma_0$ and for integers $j \geq 0$ we let $M_j$ and $N_j$ denote the functions $B_1 \subset \mathbb{R}^2 \to \mathbb{R}$ given by

$$M_j(r,\theta) = r^j\cos(j\theta) \quad \text{and} \quad N_j(r,\theta) = r^j\sin(j\theta);$$

that is, $M_j$ and $N_j$ are the two-dimensional multipoles on the disc $B_1$. The situation is depicted in Figure 2.

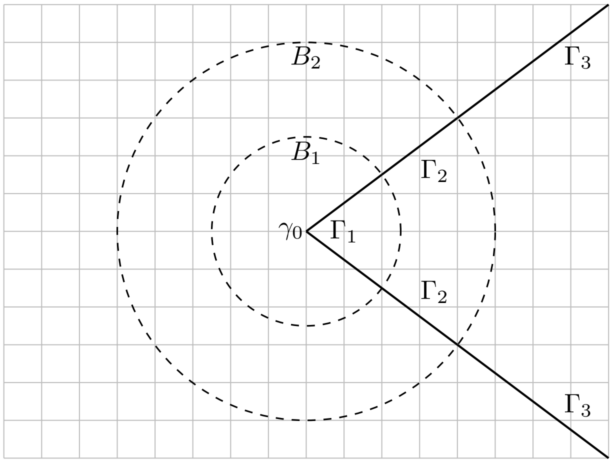


Figure 2

The boundary integral equation (5.2) can be rearranged as

$$\begin{aligned} \frac{1}{2}\sigma(x) + \int_{\Gamma_1} K(x,y)\sigma(y)dS(y) &= u(x) - \\ \int_{\Gamma_2} K(x,y)\sigma(y)dS(y) &- \int_{\Gamma_3} K(x,y)\sigma(y)dS(y). \end{aligned} \tag{5.3}$$

By virtue of the separation of the contours $\Gamma_1$ and $\Gamma_3$, the third term on the right hand side of equation (5.2) can be represented as a multipole expansion whenever $x \in \Gamma_1$. So under the assumption that the right hand side $u(x)$ satisfies the Laplace equation in $B_2$, we can introduce the approximation

$$u(x) \; - \int_{\Gamma_3} K(x,y)\sigma(y)dS(y) \approx \sum_{j=0}^{N} \alpha_j N_j(r,\theta) + \beta_j M_j(r,\theta), \tag{5.4}$$

which holds for $x \in \Gamma_1$.

Moreover, we make the assumption that for $x \in \Gamma_1$ the third term on the right hand side of equation (5.3) can be approximated as

$$\int_{\Gamma_2} K(x,y)f(y)dS(y) \approx \sum_{j=1}^{M} K(x,y_j)f(y_j)w_j,$$

where the $y_j$ lie in $\Gamma_2$; i.e., there exists an $M$-point quadrature rule discretizing the integral operator integral operator $T : L^2(\Gamma_2) \to L^2(\Gamma_1)$ defined by

$$Tf(x) = \int_{\Gamma_2} K(x,y)f(y)dS(y).$$

It follows that the restriction of $\sigma$ to the curve segment $\Gamma_1$ satisfies the integral equation

$$-\frac{1}{2}\sigma(x) + \int_{\Gamma_1} K(x,y)\sigma(y)dS(y) = \sum_{j=0}^{N} \left(\alpha_j N_j(r,\theta) + \beta_j M_j(r,\theta)\right) + \sum_{j=1}^{M} K(x,y_j)f(y_j)w_j. \tag{5.5}$$

It is now clear how to form a charge basis for $\Gamma_1$. We observe that the restriction of (5.2) to $\Gamma_1$ is invertible as an operator $L^2(\Gamma_1) \to L^2(\Gamma_1)$ and form the collection of functions obtained by solving the restricted integral equation for each of the functions of the form

$$M_j(r,\theta), \quad N_j(r,\theta), \quad \text{and} \quad K(x,y_j)$$

appearing in (5.5). The resulting functions are orthonormalized in order to form a charge basis.

Remark 5.1. *The two assumptions made above — or obvious modifications thereof — hold for virtually all problems of practical interest; indeed, some set of similar assumptions must hold for the integral equation (5.2) to be numerically tractable.*

Remark 5.2. *The procedure presented here can be easily adapted to other boundary integral equations. For instance, in the case of the exterior Neumann problem, the kernel $K(x,y)$ is now*

$$K(x,y) = \frac{\partial}{\partial \nu_x} \log|x-y|$$

*and the proper assumption on the right hand side $u(x)$ is that it can be represented as a finite sum of the normal derivatives of the multipoles $M_j$ and $N_j$ on the curve segment $\Gamma_1$.*

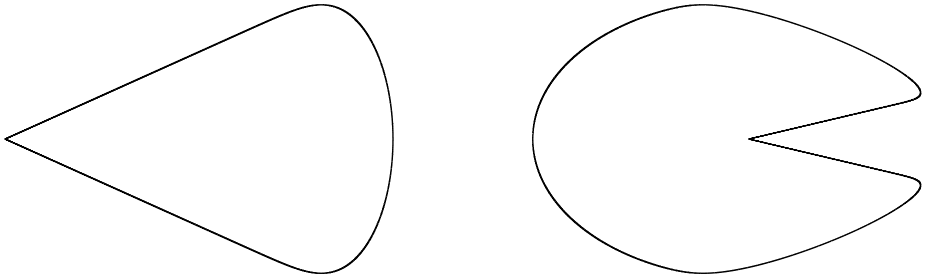
### 5.2. Universal bases for polygonal domains.

For $0 < \theta < 2\pi$ we shall call any oriented curve which is the image under a similarity transform with positive determinant (i.e., the image under rotation, translation, or scaling) of the curve in the plane parameterized over $[-1,1]$ by

$$\begin{aligned} x(t) &= |t|\cos(\theta) \\ y(t) &= -t\sin(\theta) \end{aligned} \tag{5.6}$$

a wedge of angle $\theta$. Figure 3 shows two closed curves in the plane containing wedges. Assuming counter-clockwise orientation, the contour in Figure 3(a) contains a wedge of angle $\theta < \pi$ and that in Figure 3(b) contains a wedge of angle $\theta > \pi$.

Because it is possible to classify all wedges by their angles, it is possible to build a set of "universal" bases for them. That is, by applying the procedure of the preceding section to wedges of various angles, we can construct a small collection of bases $\mathcal{B}_j$ with the following property:



(a) A “snowcone” domain containing a wedge of angle $\theta < \pi$.

(b) A “Pac-Man” domain with a wedge of angle $\theta > \pi$.

Figure 3: Two domains whose boundaries contain wedges.

> Whenever $\psi$ is the restriction to a wedge $\Gamma_0 \subset \Gamma$ of a solution $\sigma$ of the integral equation
>
> $$\lambda\sigma(x) + \int_{\Gamma} K(x,y)\sigma(y)dy = u(x), \tag{5.7}$$
>
> where $\Gamma$ is a compact closed Lipschitz curve in the plane and both the curve $\Gamma$ and the right hand side $u(x)$ satisfy the (mild) assumptions made in subsection 5.1, then $\psi$ is in the span of one of the bases $\mathcal{B}_j$.

To see that above statement is correct, two observations are necessary. First, we note that the procedure for constructing a basis for charge distributions can be applied to a range of angles rather than a single angle; that is, rather than inverting the integral equation (5.7) once for a wedge of a single angle in order to form a basis for restricted charge distributions, we sample a collection $\theta_1, \ldots, \theta_n$ of angles in a particular interval $[a,b] \subset (0, 2\pi)$ and solve the integral equation (5.7) on a wedge of each angle $\theta_j$ for the appropriate right hand sides. The resulting collection of solutions is then used to form an orthonormal basis which will — assuming a sufficient number of angles are sampled — approximately span the space of restrictions of charge distributions satisfying the integral equation (5.7) to a wedge of any angle in the range $[a,b]$. Second, we observe that owing to the invariance of Laplace's equation under scaling, rotation and translation, the obtained bases will likewise be invariant under scaling, rotation, and translation. Thus they will span the restrictions of solutions of the appropriate integral equation to a wedge regardless of its position or scale.

## 6. Numerical algorithm

We now give a detailed account of an algorithm for the construction of a set of interpolatory quadratures for the efficient Nyström discretization of a boundary integral equation of the form

$$\lambda\sigma(x) + \int_{\Gamma} K(x,y)\sigma(y)\, dS(y) = u(x), \tag{6.1}$$

where $K(x,y)$ is one of the potential theoretic kernels appearing in Section 3, over arclength parameterized wedges with angles $\theta$ in a subinterval $[a,b]$ of $(0, 2\pi)$.

The algorithm takes as input a set of angles $\theta_1, \ldots, \theta_r$ sampled from the interval $[a,b]$, an even integer $n$ specifying the number of multipoles to use as right hand sides, and an integer $l$ specifying the order of Legendre polynomials to use in construction of the “far” quadrature formula. The output of the algorithm is a collection of interpolation and quadrature schemes suitable for the Nyström discretization (as described in subsection 4.1) of (6.1) over wedges of the specified range of angles.

**Stage one: forming the spanning set.**

For each of the sampled angles $\theta$ the following sequence of steps are performed:

1. Discretize the boundary integral operator on the left side of equation (6.1) over the curve parameterized by
$$\begin{aligned} x(t) &= |t|\cos(\theta) \\ y(t) &= -t\sin(\theta) \end{aligned} \qquad |t| \leq 2$$
using a quadrature obtained by taking the image of a simply-graded mesh under a substitution of the form $u = x^{2k+1}$. Denote the quadrature nodes by $x_1, \ldots, x_q$ and the quadrature weights by $w_1, \ldots, w_q$.
2. Solve the resulting $q \times q$ linear system of equations for each of the multipoles
$$M_j(r,\theta) = r^j \cos(j\theta) \quad \text{and} \quad N_j(r,\theta) = r^j \sin(j\theta),\ j = 0, \ldots, n/2-1,$$
where $(r,\theta)$ is the usual polar coordinate system centered at the origin.
3. Solve the discretized system for the functions
$$f_j(x) = K(x, y_j)$$
where the $\{y_j\}$ are a large collection of points obtained by sampling the parameterization
$$\begin{aligned} x(t) &= |t|\cos(\theta) \\ y(t) &= -t\sin(\theta) \end{aligned}$$
at a large number of values of $|t| > 1$.
4. Restrict the set solutions so obtained to the wedge
$$\begin{aligned} x(t) &= |t|\cos(\theta) \\ (t) &= -t\sin(\theta) \end{aligned} \qquad |t| \leq 1$$
of angle $\theta$.

Denote by $\phi_1, \ldots, \phi_N$ the set of restrictions of all solutions obtained by repeating this procedure for each sampled angle $\theta$.

**Stage two: construction of an orthonormal basis.**

In this stage, the procedure described in subsection 2.4 is utilized to form an SVD of the solutions obtained in stage one of the algorithm. To wit, the following sequence of steps is performed:

1. Form the $q \times N$ matrix $A$ with entries
$$A_{ij} = \phi_j(q_i)\sqrt{w_i}.$$
2. Compute the SVD of the matrix $A$. Denote the positive singular values of $A$ by $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_k > 0$ and the corresponding singular vectors by $v_1, \ldots, v_k$.
3. Form an orthonormal basis $\sigma_1, \ldots, \sigma_k$ for the span of the $\phi_1, \ldots, \phi_N$ by letting the value of $\sigma_j$ at the quadrature node $x_i$ be given by the $i$th component of the vector $v_j$ scaled by $1/\sqrt{w_i}$. Note that given the values of a function $f$ at the quadrature nodes $x_1, \ldots, x_q$ the value of $f(x)$ at an arbitrary point $x$ can be computed via interpolation, so these values in fact determine the function $\phi_j$.

**Stage three: construction of the interpolation scheme.**

By the discussion in subsection 2.1, there exists a set of $k$ points $t_1, \ldots, t_k$ which can serve as nodes for a stable interpolation scheme for the basis function $\sigma_1, \ldots, \sigma_k$. The following sequence of steps constitute a numerical algorithm for the computation of a set of interpolation nodes.

1. Form the matrix $k \times q$ matrix $B$ with entries
$$B_{ij} = \sigma_i(x_j)\sqrt{w_j}.$$

2. Use the pivoted Gram-Schmidt algorithm with reorthogonalization to choose a set of spanning columns $j_1, \dots, j_k$ of the matrix B. Note that by construction, the rank of the matrix $B$ is $k$.
3. We shall denote by $t_1, \dots, t_k$ the $k$ quadrature nodes $x_{j_1}, \dots, x_{j_k}$ corresponding to the $k$ spanning columns of $B$.

Once the nodes $t_1, \dots, t_k$ have been computed in this fashion, we can form a matrix $C$ which interpolates the $\phi_1, \dots, \phi_k$ to an arbitrary set of points $y_1, \dots, y_m$ by solving the equation

$$C\Phi_1 = \Phi_2,$$

where $\Phi_1$ is the matrix whose columns consist of the values of the basis functions $\phi_1, \dots, \phi_k$ evaluated at the interpolation nodes $t_1, \dots, t_k$ and $\Phi_2$ is the matrix whose columns consist of the values of the basis functions at the points $y_1, \dots, y_m$, in a least squares sense.

Remark 6.1. *Interpolation nodes chosen via the pivoted Gram-Schmidt procedure do not necessary lead to a stable interpolation scheme. However, in practice it performs reliably (see* [3] *for a detailed discussion of the use of Gram-Schmidt algorithms in numerical analysis). If difficulties do arise, then an RRQR algorithm, like that described in* [12]*, can be substituted for the Gram-Schmidt procedure; stability bounds can be easily derived in this case.*

**Stage four: construction of the "far" quadrature formula.**

Since the wedges on which we solved the integral equation are discretized over the interval $[-1, 1]$, we can regard the basis functions $\sigma_1, \dots, \sigma_k$ as being defined on the interval $[-1, 1]$. In this stage, the procedure of [4] is used to construct either a generalized Chebyshev or generalized Gaussian quadrature formula for integrals of the form

$$\int_{-1}^{1} P(y)\sigma_j(y)\, dS(y),$$

where the $P(y)$ is a function on $[-1, 1]$ whose restrictions to the subintervals $[-1, 0)$ and $(0, 1]$ are Legendre polynomials of degree $l$ (i.e., $P(y)$ is a piecewise Legendre polynomial). Since the kernel $K(x, y)$ is smooth when the point $x \in \mathbb{R}^2$ is removed from $y \in \mathbb{R}^2$ — and therefore can be approximated by polynomials — this quadrature serves as the "far" quadrature formula required by the Nyström method described in subsection 4.1.

**Stage five: construction of the "diagonal" quadrature formulae.**

Just as we regarded the the basis functions as being given on the interval $[-1, 1]$, we can regard the kernel function $K_\theta(x, y)$ for the wedge of angle $\theta$ as being given on $[-1, 1] \times [-1, 1]$. In this stage, for each of the $k$ interpolation nodes $t$, the procedure of [4] is used to construct either a generalized Chebyshev or generalized Gaussian procedure for all integrals of the form

$$\int_{-1}^{1} K_{\theta_i}(t, y)\sigma_j(y) dy,$$

where $\theta_i$ varies over the set of sampled angles $\theta_1, \dots, \theta_r$ and $\sigma_j$ varies over the basis functions $\sigma_1, \dots, \sigma_k$. The $k$ resulting quadrature rules are, of course, the "diagonal" quadrature formulae required by the Nyström method described in subsection 4.1.

## 7. Numerical Results

We now present the results of a number of numerical experiments. All code was written in Fortran 77 and compiled with the Lahey/Fujitsu Linux64 Fortran Compiler Release 8.10a. Timings were performed on a PC with an Intel Core i7 2.67 GHz processor and 12GB of memory. No attempt was made to parallelize any of the code.

An algorithm for the construction of generalized Chebyshev and Gaussian quadratures for very general classes of functions was implemented. The procedure of Section 6 was implemented by coupling that code with a simple direct solver for boundary integral equations which is asymptotically $O(n^2)$ in the number of discretization nodes $n$.

| | Range of angles (radians) | Interpolation nodes | Far quadrature nodes | Largest diagonal quadrature |
|---|---|---|---|---|
| Int. Dir. | 0.706 - 0.785 | 28 | 78 | 27 |
| | 1.492 - 1.571 | 28 | 78 | 24 |
| | 2.228 - 2.356 | 28 | 76 | 22 |
| Ext. Dir. | 0.706 - 0.785 | 26 | 76 | 42 |
| | 1.492 - 1.571 | 27 | 76 | 40 |
| | 2.228 - 2.356 | 27 | 74 | 39 |
| Ext. Neu. | 0.706 - 0.785 | 35 | 90 | 29 |
| | 1.492 - 1.571 | 36 | 92 | 28 |
| | 2.228 - 2.356 | 36 | 93 | 29 |
| Int. Neu. | 0.706 - 0.785 | 36 | 97 | 52 |
| | 1.492 - 1.571 | 30 | 86 | 44 |
| | 2.228 - 2.356 | 32 | 88 | 45 |

Table 1: The number of interpolation and quadrature nodes for selected polygonal quadrature formulae.

For each of the boundary integral formulations discussed in Section 3, a collection of universal quadrature formulas for polygonal domains was constructed. More specifically, the interval $(0, 2\pi)$ was partitioned into 80 equispaced subintervals and for each such subinterval $[\theta_1, \theta_2]$ a set of interpolatory quadrature formula for wedges with angles between $\theta_1$ and $\theta_2$ was constructed using the algorithm of Section 6. On each interval $[\theta_1, \theta_2]$, 8 angles were sampled and the far quadrature formulae were constructed for piecewise Legendre polynomials of order 20. In all cases, we opted to construct generalized Chebyshev rather than generalized Gaussian quadrature formulae in order to maintain simplicity. Table 1 shows the results for selected quadrature formulae.

7.1. **A domain with a single corner point.** Figure 4 shows a simply-connected domain $\Omega_1$ whose boundary $\partial\Omega_1$ is $C^\infty$ except for a single corner point of angle $\pi/4$ radians. In this experiment, we solve the interior Neumann boundary value problem

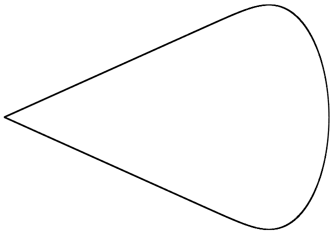


Figure 4: The domain $\Omega_1$ with a single corner of angle $\pi/4$.

$$\begin{gathered}\Delta u(x) = 0 \text{ for } x \in \Omega_1 \\ \lim_{\substack{x \to p \\ x \in \Omega_1}} \frac{\partial u}{\partial \nu_x}(x) = f(p) \text{ for } p \in \partial\Omega_1,\end{gathered} \tag{7.1}$$

where $f(p)$ is the normal derivative on $\partial\Omega_1$ of a potential function $v(x)$ generated by 5 point charges randomly placed in the exterior of the domain $\Omega_1$, via the technique outlined in Section 3 — namely by inverting integral equation

$$-\frac{1}{2}\sigma(p) + \frac{1}{2\pi}\int_{\partial\Omega_1} \sigma(y)\frac{\partial}{\partial \nu_p}\log|p-y|\,dS(y) - \sigma(p^*) = f(p) \tag{7.2}$$

in order to obtain a solution $u$ of (7.1) in the form of a single layer potential

$$u(x) = \frac{1}{2\pi}\int_{\partial\Omega_1} \log|x-y|\,\sigma(y)\,dS(y). \tag{7.3}$$

Note that $p^*$ in (7.2) denotes an arbitrary point on the boundary curve; see Section 3.

| | N | $T_{\text{solve}}$ | $E_{\text{abs}}$ | $E_{\text{circ}}$ |
|---|---|---|---|---|
| Universal quadrature | 216 | 0.006 | $3.12\times10^{-14}$ | $8.82\times10^{-15}$ |
| Simply-graded mesh | 520 | 0.081 | $4.07\times10^{-06}$ | $2.30\times10^{-07}$ |
| | 800 | 0.275 | $1.52\times10^{-10}$ | $8.58\times10^{-11}$ |
| | 1440 | 1.713 | $8.82\times10^{-12}$ | $2.76\times10^{-12}$ |

Table 2: Comparative performance of the universal quadrature formulas in solving an interior Neumann problem on the domain $\Omega_1$.

Table 2 compares the results obtained by discretizing the integral equation (7.2) using a wedge quadrature formula with those obtained using the discretizations described in subsection 4.3. In both cases, the smooth portion of the curve were discretized using 180 piecewise Legendre interpolation nodes. The discrete systems were inverted using a simple direct solver for boundary integral equations which has asymptotic running time $O(n^2)$ in the number of discretization nodes $n$.

- $N$ is the total number of interpolation nodes used to discretize the integral equation (7.2);
- $T_{\text{solve}}$ is the time required to invert the discrete linear system in seconds;
- $E_{\text{abs}}$ is the largest absolute error observed while measuring the difference between the computed solution $\tilde{u}(x)$ and the true potential function $u(x)$ at a collection of 100 randomly placed points in the interior of the domain $\Omega_1$; and
- $E_{\text{circ}}$ is an approximation (obtained with a high order Gaussian quadrature) of the relative $L^2(\Gamma)$ error $\|u-v\|_2/\|u\|_2$, where $\Gamma$ is a circle of small radius located at the center of mass of $\partial\Omega_1$.

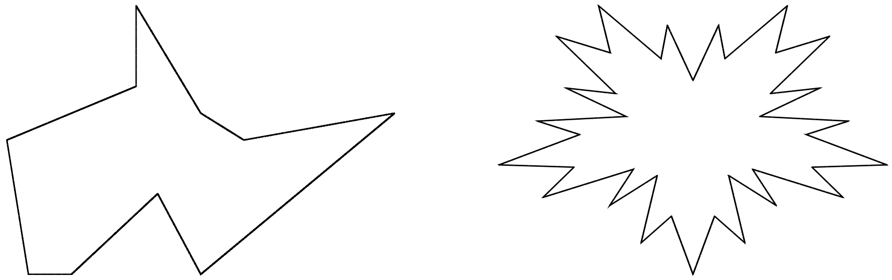
Remark 7.1. *The boundary integral formulation (7.2) used here to solve the interior Neumann problem is not necessary an optimal approach for domains with corners. There are many possible ways to modify a boundary integral formulation in order to reduce the complexity of such a problem; however, a notable advantage of the algorithm of this paper is that efficient quadratures can be obtained even using suboptimal boundary integral formulations. The precomputations can be performed, leisurely, in extended precision if need be and the resulting quadratures produce highly efficient and accurate formulas, regardless of the underlying integral equation.*

7.2. **Two polygonal domains.** In this subsection, we present results pertaining to the two polygonal domains shown in Figure 5. The domain $\Omega_2$ (pictured on the left) has 10 corner points of various angles, while the domain $\Omega_3$ (pictured on the right) has 38 corner points.

For each of the Laplace boundary value problems discussed in Section 3, the corresponding integral equation, which has the form

$$\lambda\sigma(x) + \int_\Gamma K(x,y)\sigma(y)\,dy = u(x),$$

was formed and discretized; the appropriate precomputed quadrature formula was used at each corner point and piecewise Legendre quadratures were used to discretize the smooth portions of the curve. The right hand side $u(x)$ was taken to be the restriction to the boundary of the domain of either a potential generated by a collection of 5 charges placed randomly either in the interior or exterior of the domain (depending on the boundary value problem under consideration), or the normal derivative of such a potential function on the boundary of the domain. The discrete systems were inverted using a simple direct solver (with asymptotic complexity $O(n^2)$ in the number



(a) The domain $\Omega_2$.

(b) The domain $\Omega_3$.

Figure 5: The polygonal domains under consideration in subsection 7.2.

of discretization nodes $n$). Figure 6 shows the dipole charge distribution on $\Omega_3$ obtained in the solution of the interior Dirichlet problem. Table 3 presents the results for the polygonal domain $\Omega_2$ and Table 4 for the domain $\Omega_3$; the notation used is the same as that of the preceding section.

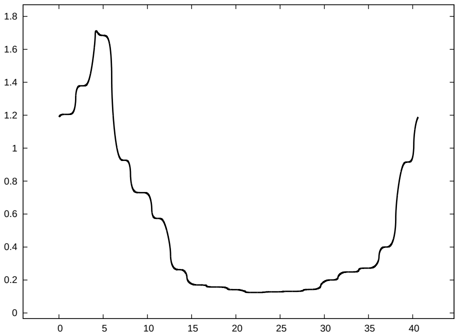


Figure 6: The continuous, but not smooth, dipole distribution on $\partial\Omega_3$ resulting from the solution of the interior Dirichlet problem.

| | N | $T_{\text{Solve}}$ | $E_{\text{abs}}$ | $E_{\text{circ}}$ |
|---|---|---|---|---|
| Interior Dirichlet | 847 | 0.364 | $4.48\times10^{-15}$ | $1.17\times10^{-14}$ |
| Exterior Dirichlet | 840 | 0.327 | $7.08\times10^{-15}$ | $3.01\times10^{-14}$ |
| Exterior Neumann | 926 | 0.437 | $5.50\times10^{-15}$ | $2.74\times10^{-14}$ |
| Interior Neumann | 900 | 0.391 | $3.71\times10^{-15}$ | $2.98\times10^{-14}$ |

Table 3: Computational results obtained for the polygonal domain $\Omega_2$.

| | N | $T_{\text{Solve}}$ | $E_{\text{abs}}$ | $E_{\text{circ}}$ |
|---|---|---|---|---|
| Interior Dirichlet | 2202 | 1.78 | $1.48\times10^{-14}$ | $4.22\times10^{-15}$ |
| Exterior Dirichlet | 2197 | 1.77 | $2.23\times10^{-14}$ | $3.41\times10^{-13}$ |
| Exterior Neumann | 2484 | 2.42 | $2.95\times10^{-14}$ | $2.31\times10^{-13}$ |
| Interior Neumann | 2821 | 2.92 | $1.08\times10^{-14}$ | $6.22\times10^{-15}$ |

Table 4: Computational results obtained for the polygonal domain $\Omega_3$.

7.3. **A polygonal domain with 250 vertices.** In this example, we solve the exterior Neumann problem

$$\begin{aligned} \Delta u(x) &= 0 \text{ for } x \in \Omega_4^c \\ \lim_{\substack{x\to p \\ x\in\Omega_4^c}} \frac{\partial u}{\partial \nu_x}(x) &= f(p) \text{ for } p \in \partial\Omega_4 \end{aligned} \tag{7.4}$$

on the polygonal domain $\Omega_4$ shown in Figure 7. The boundary of the domain $\Omega_4$ was obtained by sampling the smooth closed curve defined by the polar equation

$$r(\theta) = 1 + \frac{1}{2}\cos(n\theta)\sin(m\theta),$$

where $n = 4$ and $m = 4$, at 250 points. More specifically, the coordinates $(x_j, y_j)$ of the vertices of the polygon $\partial\Omega_4$ are given by

$$\begin{aligned} x_j &= \left[1 + \frac{1}{2}\cos(n\theta_j)\sin(m\theta_j)\right]\cos(\theta_j) \\ y_j &= \left[1 + \frac{1}{2}\cos(n\theta_j)\sin(m\theta_j)\right]\sin(\theta_j) \end{aligned}$$

where $\theta_j = 2\pi j/250$, $j = 1, 2, \ldots, 250$.

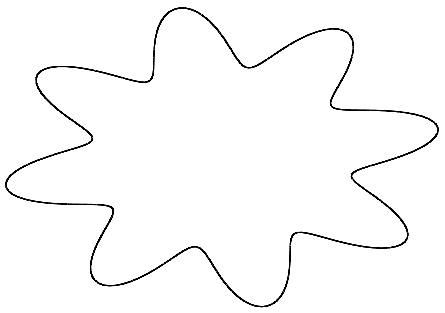


Figure 7: The polygonal domain $\Omega_4$, which was obtained by sampling a smooth curve at 250 points.

The boundary data $f(p)$ was taken to be the normal derivative on $\partial\Omega_4$ of a potential function $v(x)$ generated by a collection of 10 point charges randomly placed in the interior of the domain $\Omega_4$. The boundary integral equation

$$\frac{1}{2}\sigma(p) + \frac{1}{2\pi}\int_{\partial\Omega_4} \frac{\partial}{\partial \nu_x} \log|p - y|\sigma(y)\, dS(y) = f(p) \tag{7.5}$$

corresponding to the problem (7.4) was discretized as before; a total of 16341 discretization nodes were required. The resulting discrete system was inverted in 61.0 seconds using a direct solver which is asymptotically $O(n^2)$ in the number of discretization nodes $n$.

The true potential function $v(x)$ was compared to the single layer potential

$$u(x) = \frac{1}{2\pi}\int_{\partial\Omega_4} \log|x - y|\, \sigma(y)\, dS(y) \tag{7.6}$$

arising from the charge distribution $\sigma$ obtained by inverting (7.5) at 100 randomly chosen points in the exterior of $\Omega_4$. The largest absolute error was found to be $2.23\times10^{-14}$.

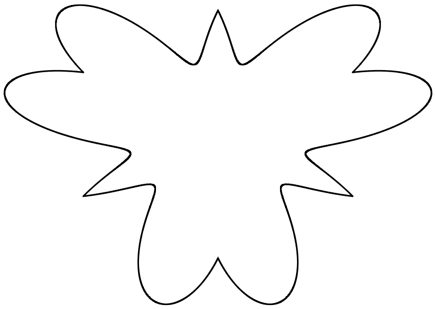


Figure 8: The nonpolygonal domain $\Omega_5$ under consideration in subsection 7.4.

7.4. **A nonpolygonal domain.** In this final example, we turn our attention to the nonpolygonal domain $\Omega_5$ with 6 corner points shown in Figure 8. Unlike the domains in previous examples, the boundary curve $\partial\Omega_5$ has nonzero curvature near corner points. This means that the wedge quadrature formulas described in subsection 7 are not applicable. Instead, specialized quadratures were constructed for each of the corner points.

Once again, as in the previous subsection, an instance of each of the boundary value problems discussed in Section 3 was solved on $\Omega_4$. In particular, for each such Laplace boundary value problem, the corresponding integral equation, which has the form

$$\lambda\sigma(x) + \int_\Gamma K(x,y)\sigma(y)\,dy = u(x),$$

was formed and discretized using the appropriate quadratures. The right hand sides $u(x)$ were once again taken to be the restriction to the boundary of the domain of either a potential generated by a collection of 5 charges placed randomly either in the interior or exterior of the domain (depending on the boundary value problem under consideration), or the normal derivative of such a potential function. The discrete systems were inverted using a simple direct solver (with asymptotic complexity $O(n^2)$ in the number of discretization nodes $n$). Table 5 presents the results for the domain $\Omega_5$; the notation is the same as that of subsection 7.

| | N | $T_{\text{Solve}}$ | $E_{\text{abs}}$ | $E_{\text{circ}}$ |
|---|---|---|---|---|
| Interior Dirichlet | 469 | 0.065 | $6.00\times10^{-13}$ | $1.74\times10^{-13}$ |
| Exterior Dirichlet | 448 | 0.067 | $3.98\times10^{-13}$ | $1.24\times10^{-13}$ |
| Exterior Neumann | 512 | 0.072 | $7.88\times10^{-13}$ | $2.03\times10^{-13}$ |
| Interior Neumann | 522 | 0.080 | $5.21\times10^{-13}$ | $9.23\times10^{-13}$ |

Table 5: Computational results obtained for the nonpolygonal domain $\Omega_5$.

## 8. Conclusions and Future Work

To summarize, in this paper, a method for the construction of quadrature formulae suitable for the Nyström discretization of Laplace boundary integral equations over more-or-less arbitrary two-dimensional curve segments was introduced. That scheme was then used to generate collections of quadrature formulae for the efficient discretization of certain Laplace boundary integral equations on polygonal domains. We close this paper by mentioning the following obvious applications and generalizations of the procedure of this paper:

- The generalization of the algorithm of Section 6 to boundary integral equations in three dimensions — that is, to equations of the form

  $$\lambda\sigma(x) + \int_{\Sigma} K(x,y)\sigma(y)\,dS(y) = u(x) \tag{8.1}$$

  where $\Sigma$ is a surface in $\mathbb{R}^3$ rather than a curve in $\mathbb{R}^2$ — is extremely straightforward. In particular, integral equations of this type can be discretized using generalizations of Gaussian polynomials for triangles. Given a small surface $\Sigma_0$ contained in $\Sigma$ discretized using a large number of quadrature nodes, the integral equation (8.1) can be discretized and solved for right-hand-sides consisting of the terms of a three dimensional multipole expansion on $\Sigma_0$ in order to form a local basis for charge distributions on the surface $\Sigma_0$. Efficient quadratures can then be constructed in essentially the same manner as described in Section 6.
- The approach detailed here for Laplace's equation can be readily adapted to the case of boundary value problems for the Helmholtz equation

  $$\Delta u + \omega^2 u = 0,$$

  assuming $\omega$ is not too large. Because the integral kernels for boundary integral equations arising from the Helmholtz equation are weakly-singular instead of continuous, additional care must be taken in discretizing the integral equations. Moreover, the lack of scale invariance in the Helmholtz equation makes the construction of universal quadratures slightly more difficult. Otherwise, however, the algorithm for the Helmholtz case is analogous to that described here with the role of the multipoles now played by the terms of the J-expansion

  $$\sum_k J_k(\omega r)\left(\alpha_k \sin(k\theta) + \beta_k \cos(k\theta)\right),$$

  where $J_k$ is the Bessel function of the first kind of order $k$.
- The algorithm presented here for the construction of universal quadrature rules for polygonal domains can be extended to any setting for which the pathological behavior of the domains can be classified efficiently. In particular, we expect to be able to extend the construction to much more general classes of domains with corner points.
- Similarly, in many engineering applications it is convenient to approximate boundary curves via $C^2$ splines. The discretization of a boundary integral equation

  $$\lambda\sigma(x) + \int_{\Gamma} K(x,y)\sigma(y)\,dS(y) = u(x) \tag{8.2}$$

  over such a contour is, however, problematic — the lack of smoothness of spline functions severely limits the order of convergence of discrete approximations of (8.2). But, by classifying the behavior of $C^2$ spline functions near singular points, it should be possible to construct a collection of quadrature formulae for domains bounded by splines analogous to those constructed here for polygonal domains.
- The quadrature construction algorithm can be used as a local solver for boundary integral equations, allowing for a divide-and-conquer approach to the solution of boundary integral equations on extremely complicated domains. This can be, for instance, exploited to solve problems on domains which are sufficiently complicated that the resulting discrete systems might not otherwise fit in memory.
- Finally, we reiterate that the numerical experiments of Section 7 were conducted using a direct solver for boundary integral equations which is asymptotically $O(n^2)$ in the number of discretization nodes $n$. Coupling the approach of this paper with a faster technique for the inversion of boundary integral equations (for instance, with the $O(n)$ fast direct solver of [19]) will allow for the solution of boundary integral equations on tremendously complicated domains.

## 9. Acknowledgments

The first author was supported by the Office of Naval Research under contract N00014-09-1-0318. The second author was supported in part by the ONR under contract N0014-07-1-0711, in part by the AFOSR under contract FA9550-09 -1-0241, and in part by DARPA-AFOSR contract FA9550-07-1-0541.

## References

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

[2] K. E. Atkinson and I. Graham, *Iterative variants of the Nyström method for second kind boundary integral operators*, SIAM J. Sci. Stat. Comp., 13 (1990), pp. 694–722.

[3] A. Björck, *Numerical Methods for Least Squares Problems*, SIAM, Philadephia, 1996.

[4] J. Bremer, Z. Gimbutas, and V. Rokhlin, *A nonlinear optimization procedure for generalized Gaussian quadratures*, Yale University, Department of Computer Science Tech Report TR1406, 2008.

[5] G. Chandler, *Galkerin's method for boundary integral equations on polygonal domains.*, J. Austral. Math. Soc., Series B, 26 (1984), pp. 1–13.

[6] H. Cheng, V. Rokhlin, and N. Yarvin, *Nonlinear optimization, quadrature, and interpolation*, SIAM J. Optim, 9 (1999), pp. 901–923.

[7] R. Coifman and Y. Meyer, *Wavelets: Calderon-Zygmund and multilinear Operators*, Cambridge University Press, 1997.

[8] E. Fabes, M. Jodeit, and N. Riviére, *Potential theoretic techniques for boundary value problems on $C^1$ domains*, Acta. Math., 141 (1978), pp. 165–186.

[9] G. Folland, *Introduction to Partial Differential Equations*, Princeton University Press, Princeton, N.J., 1976.

[10] P. Grisvard, *Elliptic Problems in Nonsmooth Domains*, Pitman, Boston, 1985.

[11] ———, *Singularities in Boundary Value Problems*, Springer-Verlag, 1992.

[12] M. Gu and S. Eisenstat, *Efficient algorithms for computing a strong rank-revealing QR factorization*, SIAM J. Sci. Comput., 17 (1996), pp. 848–869.

[13] J. Helsing and R. Ojala, *Corner singularities for elliptic problems: integral equations, graded meshes, quadrature, and compressed inverse preconditioning*, Journal of Computational Physics, 227 (2008), pp. 8820–8840.

[14] O. Kellog, *Foundations of Potential Theory*, Dover, New York, 1953.

[15] C. Kenig, *Elliptic boundary value problems on Lipschitz domains*, Beijing Lectures in Harmonic Analysis, Ann. of Math. Stud., 112 (1986), pp. 131–183.

[16] R. Kress, *A Nyström method for boundry integral equations in domains with corners*, Numerische Mathematik, 58 (1991).

[17] ———, *Integral Equations*, Springer-Verlag, New York, 1999.

[18] J. Ma, V. Rokhlin, and S. Wandzura, *Generalized Gaussian quadrature rules for systems of arbitrary functions*, SIAM J. Numer. Anal., 33 (1996), pp. 971–996.

[19] P. Martinsson and V. Rokhlin, *A fast direct solver for boundary integral equations in two dimensions*, Journal of Computational Physics, 205 (2006).

[20] P.-G. Martinsson, V. Rokhlin, and M. Tygert, *On interpolation and integration in finite-dimensional spaces of bounded functions*, Communications in Applied Mathematics and Computational Science, 1 (2006), pp. 133–142.

[21] S. Mikhlin, *Integral Equations*, Pergamon Press, New York, 1957.

[22] G. Verchota, *Layer potentials and boundary value problems for Laplace's equation in Lipschitz domains*, Journal of Functional Analysis, 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.