An Implicit Interface Boundary Integral Method for Poisson's Equation on Arbitrary Domains

Catherine Kublik*, Nicolay M. Tanushev†, and Richard Tsai*

Abstract

We propose a simple formulation for constructing boundary integral methods to solve Poisson's equation on domains with piecewise smooth boundaries defined through their signed distance function. Our formulation is based on averaging a family of parameterizations of an integral equation defined on the boundary of the domain, where the integrations are carried out in the level set framework using an appropriate Jacobian. By the coarea formula, the algorithm operates in the Euclidean space and does not require any explicit parameterization of the boundaries. We present numerical results in two and three dimensions.

Keywords: Integral equations, level set methods, elliptic problems.

1 Introduction

We consider applications which involve the solution of Poisson's equation on evolving domains that can change shapes, merge, and split up. Such applications include multiphase fluid computations [5], Mullins-Sekerka type free boundary problems [37, 52] and iterative solutions to certain inverse problems, e.g. [7]. In these applications, one of the main challenges is to accurately capture the evolution of the domain boundaries which depends on the solution of Poisson's equation on the evolving domain with appropriate boundary conditions. For such type of applications, the level set method [39] is widely used to track the evolution of the boundary. With all these considerations in mind, we propose a novel technique based on integral equations for solving Poisson's equation with a class of boundary conditions defined on the interface. We concentrate on

$$\begin{cases} \Delta u(x) = \psi_0(x) & \text{in } \Omega \\ u(x) = f(x) \text{ or } \frac{\partial u(x)}{\partial n_x} = g(x) & \text{on } \partial\Omega, \end{cases} \tag{1}$$

for a fixed domain Ω, but we will keep in mind that Ω may depend on some other variables, for example time in our target applications. Various numerical methods have been proposed to solve elliptic problems such as (1), including finite element methods [4, 14, 23, 24, 26], finite difference techniques [5, 10, 19, 27, 28, 30], the immersed interface method [30, 31] and boundary integral methods [3, 25]. The theory of finite element methods for elliptic problems is well established, and there are sophisticated and highly accurate algorithms for solving such problems. In addition, finite element based methods can handle the variable coefficient version of (1) and therefore many other elliptic problems. On the other hand, finite element methods require an explicit representation (e.g. triangulation) of the domain which makes these types of

*Department of Mathematics and ICES, University of Texas at Austin, Austin, TX 78712.

†Department of Mathematics and ICES, University of Texas at Austin, Austin, TX 78712, and Z-Terra Inc., Houston, TX 77094.

methods less tractable in the case of an evolving domain. Indeed, if the domain is changing in time it will be necessary to constantly remesh it in order to track its shape and movement, and frequent remeshings can be costly, particularly if the domain is subject to large deformations or topological changes.

In [5], the authors present a second order accurate method for solving elliptic problems on irregular domains. Their approach is to use a hybrid finite difference-element method and embed the domain in a cartesian grid. The solution is then obtained by minimizing an energy functional. They also use a polygonal representation of the domain boundary which leads to extra computations when finding intersection points between the boundary and the grid. One of the main benefits of this technique, outside its second order accuracy, is its ability to handle variable coefficient elliptic equations. In addition, this numerical scheme is natural for Neumann boundary conditions in the sense that the resulting linear system can be solve efficiently. However, the extension of the method to Dirichlet boundary conditions is not as natural and extra care has to be taken to solve the linear system efficiently.

Finite difference techniques are also popular choices for solving (1) in applications of the level set methods [20, 32]. In the work of Gibou and Fedkiw [19], the domain boundary is described implicitly by a level set function and the elliptic operator discretized using a finite difference scheme. On grid nodes away from the interface a standard centered finite difference stencil is used. For grid nodes near the interface, the parts of the finite difference stencil that lie outside of the domain are replaced by values that are constructed by extrapolation using the boundary condition at the interface and the grid nodes inside the domain. The accuracy of this scheme depends on the order of the stencil used in the finite difference method and on the order of the extrapolation method. The authors can achieve fourth order accuracy by using a fourth order stencil and a cubic extrapolation. This scheme can solve elliptic equations with variable coefficients and is natural for the interior problem with Dirichlet boundary conditions but is likely more involved if Neumann boundary conditions need to be imposed or if the equation needs to be solved on the exterior of the domain. In the work of Min etal. [36] the authors propose a second-order finite difference scheme for solving the variable coefficient Poisson's equation on regular domains using non-graded adaptive grids (i.e., grids for which the difference in size between two adjacent cells is not constrained). For numerical efficiency, they use quadtrees (in 2D) and octrees (in 3D) to represent the Cartesian grid. This scheme is extended in the work of Chen etal. [8] to irregular domains and the heat equation. Using these schemes the solution of Poisson's equation and the heat equation is obtained efficiently on locally adaptive grids. We note that most of the finite difference based schemes described above (except for the work of Min etal. [36] and Chen etal. [8] which are already on locally adaptive grids) may be difficult to extend to local level set methods in which the grids are created only in narrow bands around the interface.

The Immersed Interface Method (IIM) [30, 31] is a popular technique for solving elliptic equations on arbitrary domains, particularly if the coefficients in the equation are discontinuous. This technique uses an adaptive finite difference scheme with a locally adaptive stencil. When the stencil is applied at points near the interface, it may be necessary to use points that lie outside of the domain. The method adaptively assigns values to these points based on the jump conditions of the coefficients or sources along the interface. Unlike finite element methods, the immersed interface method can be used with an implicit representation of the domain boundary.

In contrast with the schemes described above, boundary integral methods use an integral representation of the solution, namely the solution is defined by an integral of a suitable potential over the interface. Boundary integral methods could be restrictive however, since they are only practical on problems where the fundamental solution of the PDE can be conveniently calculated. Consequently, it is not convenient to solve variable coefficient elliptic problems using these formulations. Nevertheless, boundary integral methods provide a powerful and accurate technique for the solution of linear boundary value problems with constant coefficients, which

arise in many applications including sonar, cell phone and radio antenna design. In addition, these methods enable boundary conditions to be treated automatically, including boundary conditions at infinity. For instance, if Dirichlet boundary data are given, boundary integral methods reformulate the problem as an integral equation of the form

$$f(x) = \int_{\partial\Omega} \gamma(y(s))K(x, y(s))ds + \gamma(x), \quad x \in \partial\Omega, \tag{2}$$

where K is a kernel that relates to the fundamental solution of (1) and $y(s)$ is a parameterization of $\partial\Omega$. To obtain the solution of (2), we first solve for the unknown density function γ defined on the domain boundary $\partial\Omega$ and then construct the solution u as

$$u(x) = \int_{\partial\Omega} \gamma(y(s))\tilde{K}(x, y(s))ds, \quad x \in \Omega,$$

where $\tilde{K}$ may be a different kernel related to the fundamental solution of (1). Numerical schemes based on boundary integral equations typically use high order quadratures on smooth explicitly parameterized boundaries, and may be made computationally more efficient through various techniques such as Fast Multipole Methods, e.g. [17, 22, 41], the hierarchical matrix framework [6], wavelet based techniques [12, 34] and multidirectional algorithms [16].

In this paper we present a formulation for computing integrals of the form $\int_{\partial\Omega} v(x(s))ds$ in the level set framework, and with it we propose a boundary integral method where the domain boundary is described by its signed distance function to the boundary and an equivalent integral equation is formulated on a thin tubular neighborhood around the boundary. Within the tubular neighborhood of the boundary, the integral is discretized directly by the underlying grid. Typically in a level set method, to evaluate an integral of the form $\int_{\partial\Omega} v(x(s))ds$ where $\partial\Omega$ is the zero level set of a continuous function φ, it is necessary to extend the function v defined on the boundary $\partial\Omega$ to $\mathbb{R}^n$, such that its restriction onto $\partial\Omega$ coincides with v. The extension of v, denoted $\tilde{v}$, is typically a constant extension of v. The integral is then *approximated* by an integral involving a regularized Dirac-δ function concentrated on $\partial\Omega$, namely

$$\int_{\partial\Omega} v(x(s))ds \approx \int_{\mathbb{R}^n} \tilde{v}(x)\delta_\epsilon(\varphi(x))|\nabla\varphi(x)|dx.$$

Various numerical approximations of this delta function have been proposed, see e.g. [15, 45, 47, 51]. In this paper, by parameterizing integrals over $\partial\Omega$ using nearby level sets of φ and averaging over these different parameterizations, we derive the identity

$$\int_{\partial\Omega} v(x(s))ds = \int_{\mathbb{R}^n} \tilde{v}(x)J(x)\delta_\epsilon(\varphi(x))|\nabla\varphi(x)|dx,$$

where $J(x)$ on the right hand side is the Jacobian that accounts for the change of variables made in each parameterization of the integral on the left hand side. With this formulation, we propose a numerical method based on integral equation formulations for solving the Poisson problem with constant coefficients, subject to Dirichlet, Neumann, Robin or mixed boundary conditions. Our formulation involves projecting grid nodes located nearby the domain boundary onto their closest points on the boundary. As a result, our algorithm is simple, solves both the interior and exterior problem, handles moving boundaries easily and is applicable to various different meshes without the need to approximate the interface by finding the intersection points between the boundary and the grid. In addition, since our algorithm does not rely on uniform grids, it can be naturally used in applications that utilize different narrow banding, local level set, or adaptive gridding techniques [1, 35, 40, 46]. We note that the formulation we propose here gives an exact formulation for computing boundary integrals in the level set framework, and provides a natural way of defining and computing boundary integrals in applications using the closest point formulations [33, 43].

The paper is organized as follows. In Section 2, we present the integral equations we shall solve in this paper and give a brief description of previous numerical methods that have been developed for solving these types of integral equations. We describe our new formulation in Section 3 and introduce the corresponding algorithm in Section 4. We finish by presenting some numerical results in two and three dimensions in Section 5 and conclude in Section 6.

2 Boundary integral methods for the Poisson problem

We present below the boundary integral formulations most relevant to this paper. For simplicity, we limit our presentation to the solution of Poisson's equation in the interior of a bounded domain Ω. The solution of the exterior problem can be derived accordingly. We note that in this paper, the exterior problem describes Poisson's equation on an unbounded domain with adequate boundary conditions. In Appendix B, the reader can find a detailed derivation of these formulations.

2.1 Integral equation formulations for Poisson's equation

We begin by considering the Dirichlet problem for Poisson's equation,

$$\begin{cases} \Delta u(x) = \psi_0(x) & \text{in } \Omega, \\ u(x) = f(x) & \text{on } \partial\Omega. \end{cases} \tag{3}$$

Since Dirichlet boundary conditions are imposed, we introduce an unknown density β defined on the boundary $\partial\Omega$ and represent the solution u of (3) using the double layer potential formulation

$$u(x) = \int_{\partial\Omega} \beta(y(s)) \frac{\partial \Phi(x, y(s))}{\partial n_y} ds + \int_{\Omega} \Phi(x, y)\psi_0(y)dy, \quad x \in \Omega,$$

where Φ is the fundamental solution of Laplace's equation defined in (48) in Appendix B. The Dirichlet problem is solved as follows:

1. Find the density β defined on $\partial\Omega$ such that

$$\int_{\partial\Omega} \beta(y(s)) \frac{\partial \Phi(x, y(s))}{\partial n_y} ds + \frac{1}{2}\beta(x) = f(x) - \int_{\Omega} \Phi(x, y)\psi_0(y)dy, \qquad \text{for } x \in \partial\Omega. \tag{4}$$

2. Reconstruct the solution u in Ω using the double layer potential formulation

$$u(x) = \int_{\partial\Omega} \beta(y(s)) \frac{\partial \Phi(x, y(s))}{\partial n_y} ds + \int_{\Omega} \Phi(x, y)\psi_0(y)dy, \qquad \text{for } x \in \Omega.$$

We now consider the Neumann problem

$$\begin{cases} \Delta u(x) = \psi_0(x) & \text{in } \Omega, \\ \frac{\partial u(x)}{\partial n_x} = g(x) & \text{on } \partial\Omega \text{ such that } \int_{\partial\Omega} g(x(s))ds = \int_{\Omega} \psi_0(x)dx. \end{cases} \tag{5}$$

We observe that

$$\int_{\Omega} \Delta u(x)dx = \int_{\partial\Omega} \frac{\partial u(x(s))}{\partial n_x} ds = \int_{\Omega} \psi_0(x)dx.$$

Thus, in order for the Neumann problem (5) to have a solution, it is necessary to impose the compatibility condition

$$\int_{\partial\Omega} g(x(s))ds = \int_{\Omega} \psi_0(x)dx \tag{6}$$

on g. Let us note also that the solution to the Neumann problem is not unique. As a result, it is necessary to prescribe additional conditions on the solution u in order to make it unique. In fact the number of conditions that need to be imposed on u is the number of connected components of the domain Ω. In this work we choose to impose that the solution u takes specific values at a few chosen points inside Ω, namely if the domain Ω has m connected components, we pick m points such that each point lies inside a different connected component. We shall solve the Neumann problem using the single layer potential formulation. Unlike the Dirichlet problem where both formulations may be used (see Appendix B), the single layer potential formulation is the only practical representation of the solution of the Neumann problem. The steps for solving the Neumann problem are as follows:

1. Find the density α defined on the domain boundary $\partial\Omega$ such that

$$\int_{\partial\Omega} \alpha(y(s)) \frac{\partial\Phi(x,y(s))}{\partial n_x} ds - \frac{1}{2}\alpha(x) = g(x) - \int_{\Omega} \frac{\partial\Phi(x,y)}{\partial n_x} \psi_0(y) dy, \qquad \text{for } x \in \partial\Omega. \tag{7}$$

2. Reconstruct the solution u in Ω using the single layer potential formulation

$$u(x) = \int_{\partial\Omega} \alpha(y(s))\Phi(x,y(s)) ds + \int_{\Omega} \Phi(x,y)\psi_0(y) dy, \qquad \text{for } x \in \Omega.$$

Remark: Our algorithm also solves Poisson's equation subject to boundary conditions of the form

$$\sigma(x)u(x) + \rho(x)\frac{\partial u(x)}{n_x} = g(x), \tag{8}$$

where $x \in \partial\Omega$, and σ and ρ are functions in $L^1(\partial\Omega, \mathbb{R})$. Note that if σ and ρ are constant we recover Robin boundary conditions. In addition, if $\sigma(x) = \sigma_0(x)\mathbf{1}_{\Gamma_0}(x)$ and $\rho(x) = \rho_0(x)\mathbf{1}_{\partial\Omega\setminus\Gamma_0}(x)$ where Γ_0 is a subset of $\partial\Omega$ and $\mathbf{1}_{\Gamma_0}(\cdot)$ is the characteristic function of Γ_0, we recover the mixed boundary conditions $u(x)\big|_{\Gamma_0} = \sigma_0(x)$ and $\frac{\partial u(x)}{\partial n}\Big|_{\partial\Omega\setminus\Gamma_0} = \rho_0(x)$. For the general boundary conditions (8), the algorithm becomes

1. Find the density α defined on the domain boundary $\partial\Omega$ such that

$$\begin{aligned}\int_{\partial\Omega} \left(\sigma(x)\Phi(x,y(s)) + \rho(x)\frac{\partial\Phi(x,y(s))}{\partial n_x}\right)\alpha(y(s))ds - \frac{\rho(x)}{2}\alpha(x) \\ = g(x) - \int_{\Omega}\left(\sigma(x)\Phi(x,y) + \rho(x)\frac{\partial\Phi(x,y)}{\partial n_x}\right)\psi_0(y)dy, \qquad \text{for } x \in \partial\Omega.\end{aligned}$$

2. Reconstruct the solution u in Ω using the single layer potential formulation

$$u(x) = \int_{\partial\Omega} \alpha(y(s))\Phi(x,y(s)) ds + \int_{\Omega} \Phi(x,y)\psi_0(y) dy, \qquad \text{for } x \in \Omega.$$

2.2 A brief overview of numerical methods for Boundary Integral Methods

For each of the boundary integral equations obtained in the previous section, we need to solve a Fredholm equation of either the first or second kind. In other words, we need to find a function γ defined on $\partial\Omega$, such that

$$q(x) = \int_{\partial\Omega} \gamma(y(s))K(x,y(s))ds + C_0\gamma(x),$$

where C_0 is a constant and K is either the fundamental solution of Laplace's equation or its normal derivative to $\partial\Omega$. To solve these equations numerically it is necessary to discretize the

above integrals. Three discretization methods are typically used: the Nyström method [2, 38], the collocation method [2] and the Galerkin method [2, 11]. In the Nyström method, the boundary $\partial\Omega$ is described by a set of quadrature nodes, thus enabling the integral to be discretized using a quadrature rule. The resulting solution γ is first found at the set of quadrature nodes, and then extended to all points in Ω by means of an interpolation formula. The collocation method uses subspace approximations, namely a finite-dimensional space of basis functions defined on the boundary $\partial\Omega$. Additionally, a set of points on the boundary, called collocation points, are chosen such that the solution, expressed as a linear combination of the basis functions, satisfies the given equation at each of the collocation points. The Galerkin method is a collocation method with an orthogonal basis. Each of these discretization methods leads to a discrete system of the form

$$(I + \mathtt{K}\Lambda)\gamma = q,$$

where I is the identity matrix, $\mathtt{K}$ is a dense matrix, Λ is a diagonal matrix (for example containing the quadrature weights of the Nyström method), γ is the vector of unknowns, and q is a known vector obtained from the boundary conditions. Since $\mathtt{K}$ is dense this system is usually solved using an iterative procedure. In addition, low rank approximations may be constructed to improve the efficiency of the numerical solver. One very successful approach is the Fast Multipole Method introduced by Greengard and Rokhlin in 1987 [22]. The idea is to expand the fundamental solution using a multipole expansion in order to group sources that lie close together, and treat them as a single source. The use of hierarchical matrices [6] to solve this dense system is also popular. In this case the dense matrix is partitioned into subblocks based on a hierarchical subdivision of the points where the off-diagonal blocks are compressed in low rank forms, while the diagonal and the next-to-diagonal blocks are stored densely. Finally, a different approach to solving the dense system is to consider the dense matrix as a two dimensional image and compress it using wavelets [12, 34]. In the appropriate wavelet basis or frame, dense matrices may have sparse wavelet coefficients which can be used to perform matrix multiplications in the wavelet domain at a much lower cost and higher efficiency. The solution of the original system is then obtained by inverting the solution found in the wavelet domain. For more information on numerical methods for boundary integral equations we refer the reader to Atkinson's book [2]. Even though these numerical methods are quite efficient for solving integral equations, they all rely on a discretization of the explicitly parameterized interface.

In this paper, we propose an implicit boundary integral method that does not require an explicit parameterization of the domain boundary. Instead, the boundary is described by a level set function, thus enabling us to perform all the computations on a fixed grid regardless of the location of the boundary. Should the interface evolve in time, all computations will be performed on the mesh that is used by the level set function at each time step. This makes our algorithm easy to implement for evolving interfaces in two and three dimensions. In addition, computational techniques such as Fast Multipole Methods (FMM) [21, 22] may be incorporated into our algorithm to improve its computational speed.

3 Boundary Integral equations using signed distance functions

In this section we rewrite the boundary integral equations (4) and (7) in Section 2.1 as integrals over the embedding Euclidean space with appropriate delta measures, see (16).

3.1 Derivation

We use the signed distance function d defined as

$$d(x) := \begin{cases} \inf_{y\in\Omega^c} |x-y| & \text{if } x \in \Omega, \\ -\inf_{y\in\Omega} |x-y| & \text{if } x \in \bar{\Omega}^c. \end{cases}$$

We recall a few properties of the signed distance function that will be important in the implementation of our algorithm. These properties hold more generally in $\mathbb{R}^n$, (see e.g. [13, 18]). First, if $\partial\Omega$ is sufficiently smooth, then d is smooth in some tubular neighborhood T of $\partial\Omega$ and linear with slope one along the normals to the boundary:

$$|\nabla d| = 1 \text{ for all } x \in T, \text{ with boundary condition } d|_{x\in\partial\Omega} = 0. \tag{9}$$

Second, if $\partial\Omega$ is sufficiently smooth, the Laplacian of d at a point x gives up to a multiplicative constant the mean curvature of the isosurface of d passing through x:

$$\Delta d(x) = (1-n)H(x), \tag{10}$$

where $H(x)$ denotes the mean curvature of the level set $\{\xi : d(\xi) = d(x)\}$ and n is the number of dimensions. Given a general domain Ω, we will need to compute its signed distance function to its boundary. A common approach is to choose a level set function which is positive inside Ω and negative outside (e.g. $\mathbf{1}_\Omega(\cdot)$) and then apply a "redistancing" process to obtain d at least locally near the boundary $\partial\Omega$. Since we only need d near the boundary we will use this approach. This redistancing step is computationally efficient since there exist fast algorithms for constructing signed distance functions ($O(N \log N)$ where N is the total number of grid points) such as fast marching, fast sweeping, etc. [9, 42, 44, 48, 49, 50].

Given a domain Ω described by its signed distance function constructed as explained above, we project all grid nodes located inside an ϵ tubular neighborhood of the boundary $\partial\Omega$ onto the boundary $\partial\Omega$. This operation is easily performed using the signed distance function and its gradient, see (11). We let T_ϵ be the ϵ tubular neighborhood of $\partial\Omega$ defined as

$$T_\epsilon := \{x : |d(x)| \le \epsilon\},$$

where $\epsilon > 0$. For x in the tubular neighborhood T_ϵ, we consider its projection x^* onto $\partial\Omega$ (its closest point on the boundary) obtained using the equation

$$x^* = x - d(x)\nabla d(x). \tag{11}$$

We note that the idea of using the closest point mapping for solving partial differential equations has been previously used by Ruuth etal. in [33, 43].

We now continue with the single layer potential formulation. The result for the double layer potential is obtained similarly. For clarity in the upcoming derivations, we define the following quantities: we let $\partial\Omega_\eta$ be the η level set of d defined as

$$\partial\Omega_\eta := \{x : d(x) = \eta\},$$

for $\eta \in [-\epsilon, \epsilon]$ and define

$$J_\eta := J_\eta(y(s_\eta)) = \begin{cases} 1 + \eta\kappa_\eta & \text{if } n = 2, \\ 1 + 2\eta H_\eta + \eta^2 G_\eta & \text{if } n = 3, \end{cases}$$

where κ_η is the curvature of the curve $\partial\Omega_\eta$ at $y(s_\eta)$, H_η is the mean curvature of the surface $\partial\Omega_\eta$ at $y(s_\eta)$ ($H_\eta = \frac{\kappa_\eta^{(1)} + \kappa_\eta^{(2)}}{2}$ with $\kappa_\eta^{(i)}$ its ith principal curvature), and G_η is the Gaussian curvature

of the surface $\partial\Omega_\eta$ at $y(s_\eta)$. Using the change of variable described in detail in Appendix A we write the single layer potential integral equation as

$$\int_{\partial\Omega} \alpha(z(s))\Phi(x,z(s))ds = \int_{\partial\Omega_\eta} \alpha(y^*(s_\eta))\Phi(x,y^*(s_\eta))J_\eta ds_\eta, \tag{12}$$

where $z(s)$ is a parameterization of $\partial\Omega$ and $y^*(s_\eta)$ is the projection of $y(s_\eta) \in \partial\Omega_\eta$ onto $\partial\Omega$. We remark that (12) still holds in the case where $\partial\Omega$ is a hypersurface in $\mathbb{R}^n$ ($n \in \mathbb{N}^*$). In that case the Jacobian J_η is an nth order polynomial in η (see (46) in Appendix (A.2)).

Let $\delta_\epsilon(\eta)$ be a regularized delta function (or averaging kernel) compactly supported in $[-\epsilon,\epsilon]$ satisfying the moment conditions

$$\int_{\mathbb{R}} \delta_\epsilon(\eta)d\eta = \int_{-\epsilon}^{\epsilon} \delta_\epsilon(\eta)d\eta = 1, \tag{13}$$

and

$$\int_{\mathbb{R}} \eta^j \delta_\epsilon(\eta)d\eta = 0 \quad \text{for } 1 \le j \le p, \tag{14}$$

for $p \in \mathbb{N}^*$. By the moment condition (13) we have

$$\int_{-\epsilon}^{\epsilon} \delta_\epsilon(\eta) \int_{\partial\Omega} \alpha(y^*(s))\Phi(x,y^*(s))ds d\eta = \int_{\partial\Omega} \alpha(y^*(s))\Phi(x,y^*(s))ds,$$

since the interior integral does not depend on η. Using such a delta function δ_ϵ as a weight we average (12) in η and obtain

$$\int_{\partial\Omega} \alpha(y^*(s))\Phi(x,y^*(s))ds = \int_{-\epsilon}^{\epsilon} \delta_\epsilon(\eta) \int_{\partial\Omega_\eta} \alpha(y^*(s_\eta))\Phi(x,y^*(s_\eta))J_\eta ds_\eta d\eta. \tag{15}$$

Using the coarea formula, and Equations (9) and (10), we rewrite the right-hand side of (15) as

$$\begin{aligned}\int_{-\infty}^{\infty} \delta_\epsilon(\eta) \int_{\{y:d(y)=\eta\}} \alpha(y^*(s_\eta))\Phi(x,y^*(s_\eta))J_\eta ds_\eta d\eta &= \int_{\mathbb{R}^n} \alpha(z^*)\Phi(x,z^*)\delta_\epsilon(d(z))J(z)|\nabla d(z)|dz \\ &= \int_{\mathbb{R}^n} \alpha(z^*)\Phi(x,z^*)\delta_\epsilon(d(z))J(z)dz, \end{aligned} \tag{16}$$

with $z^* = z - d(z)\nabla d(z)$ for $z \in \mathbb{R}^n$ and

$$J(z) = \begin{cases} 1 - d(z)\Delta d(z) & \text{if } n = 2, \\ 1 - d(z)\Delta d(z) + d(z)^2\langle \nabla d, adj(Hess(d))\nabla d\rangle & \text{if } n = 3, \end{cases} \tag{17}$$

where $\langle\cdot,\cdot\rangle$ is the Euclidean inner product and $adj(Hess(d))$ is the adjugate matrix of the Hessian of d. Combining (15) and (16) we obtain for $x \in \Omega$,

$$\int_{\partial\Omega} \alpha(z(s))\Phi(x,z(s))ds = \int_{\mathbb{R}^n} \alpha(z^*)\Phi(x,z^*)\delta_\epsilon(d(z))J(z)dz. \tag{18}$$

For the double layer potential formulation similar calculations can be made to obtain the identity

$$\int_{\partial\Omega} \beta(z(s))\frac{\partial\Phi(x,z(s))}{\partial n_y}ds = \int_{\mathbb{R}^n} \beta(z^*)\frac{\partial\Phi(x,z^*)}{\partial n_{z^*}}\delta_\epsilon(d(z))J(z)dz, \tag{19}$$

In fact, the result is general and can be summarized in the following theorem:

Theorem 3.1 *Consider a C^2 compact hyper surface $\Gamma \subset \mathbb{R}^n$ and let d be the signed distance function to Γ. Define $\delta_\epsilon : \mathbb{R} \mapsto \mathbb{R}$ to be a regularized delta function compactly supported in $[-\epsilon, \epsilon]$ satisfying the moment conditions* (13) *and* (14)*. If v is a continuous function defined on Γ, then for sufficiently small $\epsilon > 0$ we have*

$$\int_\Gamma v(x(s))ds = \int_{\mathbb{R}^n} v(z - z\nabla d(z))\delta_\epsilon(d(z))J(z)dz,$$

where $J(z)$ is defined in (17) *for $n = 2, 3$ and in* (46) *in higher dimensions.*

Equations (18) and (19) are particular cases of Theorem 3.1 for the single and double layer potentials.

3.2 Truncation of the Jacobian polynomials

In this section we investigate the error made when evaluating (16) using a truncated Jacobian. We assume that $x \in \mathbb{R}^n$ is sufficiently distant from the boundary so that α, β, Φ, and $\frac{\partial \Phi}{\partial n}$ are smooth and bounded. Therefore in the following discussion, we shall replace the integrant $\alpha\Phi$ or $\beta\frac{\partial \Phi}{\partial n}$ by a smooth function f. As shown in the Appendix A.3 $\mathbb{R}^n$, the Jacobian J_η is a $d-1$ degree polynomial in η. We look at the error made when J_η is replaced by the polynomial $J_\eta^{(m)}$ where only the first m terms in J_η are kept. We have

$$\begin{aligned}
& \int_{-\epsilon}^{\epsilon} \delta_\epsilon(\eta) \int_{\partial\Omega_\eta} f(y^*(s_\eta))J_\eta ds_\eta d\eta - \int_{-\epsilon}^{\epsilon} \delta_\epsilon(\eta) \int_{\partial\Omega_\eta} f(y^*(s_\eta))J_\eta^{(m)} ds_\eta d\eta \\
= & \int_{-\epsilon}^{\epsilon} \delta_\epsilon(\eta) \int_{\partial\Omega_\eta} f(y^*(s_\eta))\eta^{m+1} Q_\eta^{(n-(m+1))} ds_\eta d\eta \\
= & \int_{-\epsilon}^{\epsilon} \eta^{m+1}\delta_\epsilon(\eta)\mathcal{I}(\eta)d\eta,
\end{aligned}$$

where $Q_\eta^{n-(m+1)}$ is a polynomial in η of degree $n-(m+1)$ and $\mathcal{I}(\eta) = \int_{\partial\Omega_\eta} f(y^*(s_\eta))Q_\eta^{(n-(m+1))}ds_\eta$. Writing $\mathcal{I}(\eta)$ in its Taylor series around zero

$$\mathcal{I}(\eta) = a_0(x) + a_1(x)\eta + a_2(x)\eta^2 + \cdots,$$

it follows that if the kernel, i.e. δ_ϵ, has p vanishing moments, we have

$$\begin{aligned}
\left|\int_{-\epsilon}^{\epsilon} \eta\delta_\epsilon(\eta)\mathcal{I}(\eta)d\eta\right| &= \left|\int_{-\epsilon}^{\epsilon} \eta\delta_\epsilon(\eta)\sum_{i=0}^{\infty} a_i(x)\eta^i d\eta\right| \\
&= \left|\sum_{i=\max(0,p-m)}^{\infty} a_i(x)\int_{-\epsilon}^{\epsilon} \underbrace{\delta_\epsilon(\eta)}_{=\mathcal{O}(\frac{1}{\epsilon})}\eta^{i+1}d\eta\right| \\
&= \begin{cases} \mathcal{O}(\epsilon^{m+1}) & \text{if } p < m, \\ \mathcal{O}(\epsilon^{p+1}) & \text{if } p \geq m. \end{cases}
\end{aligned}$$

Thus if the kernel has the same (or a higher) number of vanishing moments than the order of the approximation of the Jabobian, the error will be governed by the number of vanishing moments in ϵ. In this case, it actually does not matter which approximations of the Jacobian is used as long as its order m is smaller than the number of vanishing moments. On the other hand, if the number of vanishing moments is smaller than the order of the approximation, the error is

dominated by the order of approximation of J_η. In this case, it is advantageous to use the best approximation to J_η.

The above estimates suggest that in the two dimensional case the maximum error made by the use of $J_\eta^{(0)}$ scales like ϵ^{p+1} for any moment p of the averaging kernel. In other words, for symmetric δ_ϵ, it suffices to use $J_\eta^{(0)}$ in the computation. In the three dimensional case the maximum error resulting from the use of $J_\eta^{(0)}$ scales similarly to the two dimensional case, namely ϵ^{p+1}. On the other hand, if $J_\eta^{(1)}$ is used, the error scales like ϵ^{p+1} if $p \geq 1$ and ϵ^2 if $p = 0$. Consequently in three dimensions, if the averaging kernel has moment 1 or higher, the error incurred by the either approximation $J_\eta^{(0)}$ or $J_\eta^{(1)}$ will be the same.

3.3 Algorithms

The Dirichlet problem (3) with the double layer potential formulation can be solved using the following procedure:

Solution of the Dirichlet problem (3) **with the double layer potential formulation.** Let Ω be a bounded set in $\mathbb{R}^n$ $(n = 2, 3)$ with boundary $\partial\Omega$ defined through its signed distance function $d(x)$.

1. Find the density β defined on $\partial\Omega$ such that

$$\int_{\mathbb{R}^n} \beta(z^*) \frac{\partial \Phi(x, z^*)}{\partial n_{z^*}} \delta_\epsilon(d(z)) J(z) dz + \frac{1}{2}\beta(x) = f(x) - \int_\Omega \Phi(x, y)\psi_0(y) dy, \tag{20}$$

   where $z^* = z - d(z)\nabla d(z)$ and J is defined in (17).

2. Reconstruct u in Ω using

$$u(x) = \int_{\mathbb{R}^n} \beta(z^*) \frac{\partial \Phi(x, z^*)}{\partial n_{z^*}} \delta_\epsilon(d(z)) J(z) dz + \int_\Omega \Phi(x, y)\psi_0(y) dy.$$

The Neumann problem can be solved with the single layer potential formulation by the following procedure:

Solution of the Neumann problem (5) **with the single layer potential formulation.** Let Ω be a bounded set in $\mathbb{R}^n$ $(n = 2, 3)$ with boundary $\partial\Omega$ defined through its signed distance function $d(x)$.

1. Find the density α defined on $\partial\Omega$ such that

$$\int_{\mathbb{R}^n} \alpha(z^*) \frac{\partial \Phi(x, z^*)}{\partial n_x} \delta_\epsilon(d(z)) J(z) dz - \frac{1}{2}\alpha(x) = g(x) - \int_\Omega \frac{\partial \Phi(x, y)}{\partial n_x} \psi_0(y) dy, \tag{21}$$

   where $z^* = z - d(z)\nabla d(z)$ and J is defined in (17).

2. Reconstruct u in Ω using

$$u(x) = \int_{\mathbb{R}^n} \alpha(z^*) \Phi(x, z^*) \delta_\epsilon(d(z)) J(z) dz + \int_\Omega \Phi(x, y)\psi_0(y) dy.$$

Note that the integrals in (20) and (21) are now over a tubular neighborhood around $\partial\Omega$ rather than over the boundary $\partial\Omega$. These integrals are very easily discretized on a mesh that embeds the boundary $\partial\Omega$.

4 Discretization

In this section we present the discretization of (20) and (21) and introduce the full algorithm. We focus mainly on (20) and (21) since the double layer potential formulation for the Dirichlet problem leads to a discrete system with a better condition number than the one obtained with the single layer potential. However, the single layer potential is needed to solve the Dirichlet problem when the more general boundary conditions (8) are used.

We embed the domain Ω into the rectangle $R = [a, b]^n$, where $n = 2, 3$, and $a, b \in \mathbb{R}$ are chosen so that Ω lies completely inside R. The rectangle R constitutes our computational domain. For simplicity in the presentation of our algorithm, we work with a uniform discretization of R and let $h = \frac{b-a}{N}$ denote the grid size in each of the coordinate directions, however we note that our algorithm can be used on any non uniform discretization of the computational grid. We compute the projected points $x_i^* \in \partial\Omega$ as

$$x_i^* = x_i - d_i \nabla_h d_i,$$

where $x_i \in T_\epsilon$, $d_i = d(x_i)$ and ∇_h is the centered discrete gradient operator operating on d at grid node i, namely

$$\nabla_h d_i = (D^c_{1,h} d_i, \cdots, D^c_{n,h} d_i),$$

where

$$D^c_{j,h} d_i = \frac{d_{i+e_j} - d_{i-e_j}}{h} = \frac{d_{i_1,\cdots,i_j+1,\cdots,i_n} - d_{i_1,\cdots,i_j-1,\cdots,i_n}}{2h}$$

is the central difference quotient in the jth coordinate direction, for $1 \le j \le n$. We define the following quantities

$$f_i^* := f(x_i^*) - h^n \sum_j \Phi(x_i^*, y_j)\psi_0(y_j)H(d_j),$$

where H is the Heaviside function,

$$\Phi^*_{i,j} := \Phi(x_i^*, x_j^*),$$

$$\delta_i := \delta_\epsilon(d(x_i)),$$

$$\alpha_i^* := \alpha(x_i^*),$$

and

$$J_i = \begin{cases} 1 - d_i \Delta_h d_i & \text{if } n = 2, \\ 1 - d_i \Delta_h d_i + d_i^2 \langle \nabla_h d_i, adj(Hess(d_i)) \nabla_h d_i \rangle & \text{if } n = 3, \end{cases}$$

where Δ_h is the discrete Laplacian operator defined as

$$\Delta_h d_i := \sum_{j=1}^{n} D^+_{j,h} D^-_{j,h} d_i,$$

with

$$D^+_{j,h} d_i := \frac{d_{i+e_j} - d_i}{h} = \frac{d_{i_1,\cdots,i_j+1,\cdots,i_n} - d_{i_1,\cdots,i_n}}{h},$$

and

$$D^-_{j,h} d_i := \frac{d_i - d_{i-e_j}}{h} = \frac{d_{i_1,\cdots,i_n} - d_{i_1,\cdots,i_j-1,\cdots,i_n}}{h},$$

are forward and backward difference quotients respectively in the jth coordinate direction, for $1 \le j \le n$. In the rest of this section, $\nabla_{m,h}$ ($m = 1, 2$) will denote the discrete centered gradient with respect to the mth variable. We discretize the integral in (20) using the Riemann sum

$$h^n \sum_j \frac{\partial \Phi^*_{i,j}}{\partial n_j^*} \delta_j J_j \beta_j^*,$$

where $\frac{\partial \Phi^*_{i,j}}{\partial n^*_j} = \nabla_{2,h}\Phi(x^*_i, x^*_j) \cdot n_{x^*_j}$. Since d is the signed distance function to $\partial\Omega$, we can express the normal $n_{x^*_j}$ as $-\nabla_h d(x^*_j) = -\nabla_h d^*_j$. It follows that $\frac{\partial \Phi^*_{i,j}}{\partial n^*_j} = -\nabla_{2,h}\Phi^*_{i,j} \cdot \nabla_h d^*_j$. However, since d (and thus $\nabla_h d_j$) is only known on the regular grid, we compute $\nabla_h d^*_j$ at the projected points by interpolating the values of $\nabla_h d_j$ from the regular nodes to the projected points. We use a bilinear interpolation technique in two dimensions and a trilinear interpolation in three dimensions. It follows that the discretization of (20) becomes

$$f^*_i = -h^n \sum_j \left(\nabla_{2,h}\Phi^*_{i,j} \cdot \nabla_h d^*_j\right) \delta_j J_j \beta^*_j + \frac{1}{2}\beta^*_i,$$

or in matrix form

$$\left(B + \frac{1}{2}I\right)\beta^* = f^*, \tag{22}$$

where $B_{i,j} = -h^n \left(\nabla_{2,h}\Phi^*_{i,j} \cdot \nabla_h d^*_j\right) \delta_j J_j$ and I is the identity matrix. The final solution u is obtained by computing

$$u = \tilde{A}\beta^* + h^n \sum_j \Phi(x_i, y_j)\psi_0(y_j)H(d_j),$$

where $\tilde{A}_{i,j} = -h^n \frac{\partial \Phi_{i,j}}{\partial n^*_j}\delta_j J_j$ and $\frac{\partial \Phi_{i,j}}{\partial n^*_j} = \frac{\partial \Phi(x_i, y^*_j)}{\partial n^*_j}$.

In a similar fashion, we discretize the integral in the left-hand side of (21) as

$$h^n \sum_j \left(\nabla_{1,h}\Phi^*_{i,j} \cdot n_{x^*_i}\right) \delta_j J_j \alpha^*_j.$$

We remark that

$$\begin{aligned}
\nabla_{1,h}\Phi^*_{i,j} \cdot n_{x^*_i} &= \nabla_{1,h}\Phi(x^*_i, x^*_j) \cdot n_{x^*_i} \\
&= -\nabla_{2,h}\Phi(x^*_i, x^*_j) \cdot n_{x^*_i}, \quad \text{using Theorem (B.1) in Appendix B} \\
&= \nabla_{2,h}\Phi(x^*_j, x^*_i) \cdot n_{x^*_i} \\
&= \left(\nabla_{2,h}\Phi^*_{j,i} \cdot n_{x^*_i}\right) \\
&= \left(\left(\nabla_{2,h}\Phi^*_{i,j} \cdot n_{x^*_j}\right)_{i,j}\right)^T.
\end{aligned}$$

Thus, we can write the discrete system for (21) as

$$g^*_i = -h^n \sum_j \left(\nabla_{2,h}\Phi^*_{j,i} \cdot \nabla_h d^*_i\right) \delta_j J_j \alpha^*_j - \frac{1}{2}\alpha^*_i,$$

or in matrix form

$$\left(C - \frac{1}{2}I\right)\alpha^* = g^*, \tag{23}$$

where $C_{i,j} = -h^n \left(\left(\nabla_{2,h}\Phi^*_{i,j} \cdot n_{x^*_j}\right)_{i,j}\right)^T \delta_j J_j$ and $g^*_i = g(x^*_i) - h^n \sum_j \frac{\partial \Phi(x^*_i, y_j)}{\partial n_{x^*_i}} \psi_0(y_j)H(d_j)$, with H the Heaviside function. The solution u is constructed using

$$u = \bar{A}\alpha^* + h^n \sum_j \Phi(x_i, y_j)\psi_0(y_j)H(d_j),$$

where $\bar{A}_{i,j} = -h^n \Phi(x_i, y^*_j)\delta_j J_j$.

The matrix $\bar{A}$ derived above typically has a very bad condition number that increases as the density of the projected points increases. This is caused by the singularity of the gradient of $\Phi(x, y)$ as x approaches y. It is therefore necessary to regularize $\frac{\partial \Phi}{\partial n}$ when x and y are too close.

Remark: When solving the Neumann problem, it is necessary to impose additional conditions on the solution in order to make it unique. As described in Section 2.1, we impose that the solution takes specific values at a few chosen points inside Ω, where the number of points selected depends on the number of connected components of Ω. For convenience, we write Ω as the union of its connected components $\Omega = \bigcup_{i=1}^{m} \Omega_i$ such that any two $\bar{\Omega}_i$ and $\bar{\Omega}_j$ $(i \neq j)$ are disjoint. For each Ω_i we impose an extra condition by selecting a point inside Ω_i and a value $v_i \in \mathbb{R}$, and prescribe

$$u(x_i) = \int_{\partial\Omega} \alpha(y(s))\Phi(x_i, y(s))ds + \int_{\Omega} \Phi(x_i, y)\psi_0(y)dy = v_i, \qquad \text{for } i = 1, \cdots, m,$$

which discretized becomes

$$h^n \sum_j \Phi(x_i, y_j^*)\delta_j J_j \alpha_{j^*} = v_i - h^n \sum_j \Phi(x_i, y_j)\psi_0(y_j)H(d_j), \qquad \text{for } i = 1, \cdots, m. \tag{24}$$

These conditions are used to replace m rows of the matrix $C - \frac{1}{2}I$ defined in (23). For the best condition number, we select the rows that correspond to the m farthest grid points x_i to the interface. Additionally, we scale (24) in order to keep $C - \frac{1}{2}I$ diagonally dominant. To be explicit, if the r_i-th row of the matrix $C - \frac{1}{2}I$ is to be replaced (i ranging from 1 to m), we replace it using the left-hand side of the following equation

$$\mathcal{S}_i h^n \sum_j \Phi(x_i, y_j^*)\delta_j J_j \alpha_{j^*} = \mathcal{S}_i \left(v_i - h^n \sum_j \Phi(x_i, y_j)\psi_0(y_j)H(d_j) \right), \tag{25}$$

where $\mathcal{S}_i = \frac{-1}{2h^n \Phi(x_i, y^*_{r_i})\delta_{r_i} J_{r_i}}$. The scaling factor $\mathcal{S}_i$ ensures that the r_ith term in the r_ith row (a diagonal term in the matrix) is still $-\frac{1}{2}$. This ensures that all diagonal terms in the modified version of $C - \frac{1}{2}I$ are $-\frac{1}{2}$ as they all were before the modification. This choice ensures that the new matrix is still diagonally dominant.

4.1 Regularization of the normal derivative of the fundamental solution

In this section we describe our regularization for $\frac{\partial \Phi(x,y)}{\partial n_y}$. The same regularization applies to $\frac{\partial \Phi(x,y)}{\partial n_x}$.

Two dimensions

In two dimensions we have

$$\frac{\partial \Phi(x, y)}{\partial n_y} = \nabla_y \Phi(x, y) \cdot n_y = -\frac{1}{2\pi} \frac{x - y}{|x - y|^2} \cdot n_y,$$

where n_y is the outward unit normal at the point y on the boundary $\partial\Omega$. To understand the behavior of $\frac{\partial \Phi(x,y)}{\partial n_y}$ we assume that x is on the osculating circle of $\partial\Omega$ at y. To further simplify our calculations, we consider a local frame such that the osculating circle is a circle of radius

R centered at $(0,0)$, y is the fixed point $(R,0)$ and $x = (R\cos\theta, R\sin\theta)$ for $\theta \in [0, 2\pi]$. In this case the normal derivative becomes

$$\frac{\partial \Phi(x,y)}{\partial n_y} = -\frac{1}{2\pi R}\frac{(\cos\theta - 1)}{(\cos\theta - 1)^2 + \sin^2\theta} = \frac{1}{4\pi R}.$$

Thus, regardless of the location of the point x on the circle, $\frac{\partial \Phi(x,y)}{\partial n_y}$ has a constant value when the two points x and y are both on the circle. For a general smooth boundary, we consider the approximation of the boundary locally around a point y by its osculating circle, and obtain, for sufficiently close $x, y \in \partial\Omega$,

$$\frac{\partial \Phi(x,y)}{\partial n_i} = \frac{\kappa_i}{4\pi} + \mathcal{O}(|x-y|^\ell),$$

where $i = x, y$ for $x, y \in \partial\Omega$, κ_i is the curvature of the osculating circle at i, $\ell = 1$ for a general curve and $\ell = 2$ if y is a vertex, namely the contact order between the curve at y and its osculating circle is at least 4 (see Appendix C.1).

Thus we regularize $\frac{\partial \Phi^*_{i,j}}{\partial n^*_j}$ as follows:

$$\frac{\partial \Phi^*_{i,j}}{\partial n^*_j}_{reg} = \begin{cases} \frac{1}{4\pi}\kappa^*_j & \text{if } |x^*_i - x^*_j| < \tau, \\ \frac{\partial \Phi^*_{i,j}}{\partial n^*_j} & \text{else }, \end{cases} \tag{26}$$

where κ^*_j is the curvature of the interface at the point x^*_j and τ is taken to be $\mathcal{O}(h)$, with h denoting the mesh size.

Three dimensions

In three dimensions the expression of the normal derivative of the fundamental solution is

$$\frac{\partial \Phi(x,y)}{\partial n_y} = \nabla_y \Phi(x,y) \cdot n_y = -\frac{1}{4\pi}\frac{x-y}{|x-y|^3} \cdot n_y,$$

where n_y is the outward unit normal at the point y on the boundary $\partial\Omega$. Unlike the two dimensional case, in three dimensions the pointwise limit of $\frac{\partial \Phi}{\partial n_y}(x,y)$ as $y \to x$ does not exist. We therefore look at the full integral

$$I(x;\alpha) := \int_{\partial\Omega} \frac{\partial \Phi}{\partial n_y}(x,y)\alpha(y)dS(y), \quad x \in \partial\Omega,$$

where $\alpha : \mathbb{R}^3 \mapsto \mathbb{R}$ is a smooth function. We write $I(x;\alpha)$ as

$$\int_{\partial\Omega \setminus U(x;\tau)} \frac{\partial \Phi}{\partial n_y}(x,y)\alpha(y)dS(y) + \int_{U(x;\tau)} \frac{\partial \Phi}{\partial n_y}(x,y)\alpha(y)dS(y),$$

and approximate $\frac{\partial \Phi}{\partial n_y}(x,y)$ weakly locally in a small neighborhood $U(x;\tau) \subset \partial\Omega$ of x and assume that

$$\sup_{y \in U(x;\tau)} m_{\partial\Omega}(x,y) \le \tau,$$

where $m_{\partial\Omega}(x,y)$ is the geodesic distance between x and y on $\partial\Omega$. We replace $\frac{\partial \Phi}{\partial n_y}(x,y)$ by a function $K(x,y) = K_{U(x;\tau)}(x,y)$ for $y \in U(x;\tau)$ such that

$$\int_{U(x;\tau)} \frac{\partial \Phi}{\partial n_y}(x,y)\alpha(y)dS(y) \approx \tilde{I}_\tau(x;\alpha) := \int_{U(x;\tau)} K_{U(x;\tau)}(x,y)\alpha(y)dS(y).$$

Expanding α around x we have

$$\begin{aligned}&\int_{U(x;\tau)} \frac{\partial \Phi}{\partial n_y}(x,y)\alpha(y)dS(y)\\ =\;& \alpha(x)\int_{U(x;\tau)} \frac{\partial \Phi}{\partial n_y}(x,y)dS(y) + \nabla\alpha(x)\cdot\int_{U_x} \frac{\partial \Phi}{\partial n_y}(x,y)\,(y-x)dS(y) + \cdots\end{aligned}$$

We may therefore seek a function $K(x,y)$ that satisfies the following conditions

$$\int_{U(x;\tau)} K(x,y)dS(y) = \int_{U(x;\tau)} \frac{\partial \Phi}{\partial n_y}(x,y)dS(y), \tag{27}$$

and

$$\int_{U(x;\tau)} K(x,y)y^\nu dS(y) = \int_{U(x;\tau)} \frac{\partial \Phi}{\partial n_y}(x,y)y^\nu dS(y), \tag{28}$$

where $y^\nu = \Pi_{j=1}^3 y_j^{\nu_j}$, for $y = (y_1,y_2,y_3) \in U(x;\tau)$ and $\nu = (\nu_1,\nu_2,\nu_3) \in \mathbb{R}^3$ with $\nu_j \geq 0$, $j = 1,2,3$. Since the interface $\partial\Omega$ is not known explicitly, it is difficult to carry out the integrations (27) and (28). Instead we approximate the interface near $x \in \partial\Omega$ by a surface, the equation of which is known, and carry out the above integrations analytically on that surface. In this paper we only use the first moment condition (27) and choose K to be

$$K(x,y) = K_\tau(x) := \frac{1}{|\tilde{U}(x;\tau)|}\int_{\tilde{U}(x;\tau)} \frac{\partial \Phi}{\partial n_y}(x,y)dS(y),$$

where $\tilde{U}(x;\tau)$ is a neighborhood of x on the approximate surface.

The simplest strategy is to approximate the interface near x by its tangent plane $\mathcal{T}$ at $x \in \partial\Omega$. In this case,

$$K_\tau(x) = \frac{1}{|\bar{U}(x;\tau)|}\int_{\bar{U}(x;\tau)} \frac{\partial \Phi}{\partial n_y}(x,y)dS(y) = 0,$$

(see Appendix C.2), where $\bar{U}(x;\tau)$ is a local neighborhood of x on the tangent plane $\mathcal{T}$. This regularization amounts to throwing out the points that are too close to x, for each $x \in \partial\Omega$. Even though this approximation gives decent results, the accuracy resulting from this regularization can be further ameliorated.

Here, we propose one convenient improvement: We approximate the interface near x by its osculating paraboloid at $x \in \partial\Omega$. In this case, we obtain

$$K_\tau(x) = \frac{1}{8\pi\tau}\left(\kappa_1+\kappa_2\right) - \frac{1}{\pi}\left(\frac{5}{512}\left(\kappa_1^3+\kappa_2^3\right) + \frac{25}{1536}\kappa_1\kappa_2\left(\kappa_1+\kappa_2\right)\right)\tau + \mathcal{O}(\tau^3), \tag{29}$$

where κ_1 and κ_2 are the two principal curvatures of the surface $\partial\Omega$ at $x \in \partial\Omega$ and τ is the Euclidean distance from x computed on the tangent plane to the surface $\partial\Omega$ at x. With this choice of kernel we have

$$\begin{aligned}\int_{U(x;\tau)} \frac{\partial \Phi(x,y)}{\partial n_y}\alpha(y)dS(y) =& \alpha(x)\left(\frac{1}{8\pi\tau}\left(\kappa_1+\kappa_2\right) - \frac{1}{\pi}\left(\frac{5}{512}\left(\kappa_1^3+\kappa_2^3\right) + \frac{25}{1536}\kappa_1\kappa_2\left(\kappa_1+\kappa_2\right)\right)\tau\right)\\ &+ \mathcal{O}(\tau^p),\end{aligned}$$

where $p = 2$ in general and $p = 3$ if x is a vertex. More details on the relevant calculations are presented in Appendix C.2. Using this regularization we implement the discrete normal

derivative $\dfrac{\partial \Phi^*_{i,j}}{\partial n^*_j}$ in 3D as follows:

$$\frac{\partial \Phi^*_{i,j}}{\partial n^*_j}_{reg} = \begin{cases} K_\tau(x^*_j) & \text{if } |x^*_i - x^*_j|_{P_{x_{i^*}}} < \tau, \\ \dfrac{\partial \Phi^*_{i,j}}{\partial n^*_j} & \text{else }, \end{cases} \tag{30}$$

where K_τ is given by (29), τ is a small tolerance and $|.|_{P_x}$ is the Euclidean distance computed on the tangent plane to the surface at $x \in \partial\Omega$. Once regularized, we solve the discrete systems (22) and (23) using a bi-conjugate gradient stabilized solver.

4.2 Algorithms

Algorithm 1 *Solution of Poisson's equation with Dirichlet boundary conditions on* Ω*, namely*

$$\begin{cases} \Delta u(x) = \psi_0(x), & \text{in } \Omega, \\ u(x) = f(x), & \text{on } \partial\Omega, \end{cases}$$

using the double layer potential.
Given Ω *defined through its signed distance function* $d(x)$ *and* $\epsilon > 0$*,*

1. *Define tubular neighborhood* T_ϵ *and project points from* T_ϵ *onto* $\partial\Omega$*:*

$$x^*_i = x_i - d_i \nabla_h d_i, \ x_i \in T_\epsilon.$$

2. *Form the matrix* $A = (B + \frac{1}{2} I)$ *where*

$$B_{i,j} = -h^n \delta_j J_j \frac{\partial \Phi^*_{i,j}}{\partial n^*_j}_{reg},$$

and the vector f^* *such that*

$$f^*_i = f(x^*_i) - h^n \sum_{j \in \mathbb{L}} \Phi(x^*_i, y_j) \psi_0(y_j) H(d_j), \quad x_i \in T_\epsilon.$$

Here $\frac{\partial \Phi^*_{i,j}}{\partial n^*_j}_{reg}$ *is defined in* (26) *and* (30) *for the two dimensional and three dimensions cases respectively.*

3. *Solve the system* $A\beta^* = f^*$ *using a bi-conjugate gradient method.*

4. *Form the matrix* $\tilde{A}$ *such that*

$$\tilde{A}_{i,j} = -h^n \delta_j J_j \frac{\partial \Phi_{i,j}}{\partial n^*_j}_{reg}, \quad x_i \in [a,b]^n, \ x_j \in T_\epsilon.$$

5. *Construct the solution* u *as* $u = \tilde{A}\beta^*$*.*

Algorithm 2 *Solution of Poisson's equation with Neumann boundary conditions on* Ω *where* $\Omega = \bigcup_{i=1}^{m} \Omega_i$ *such that the intersection of any two* $\bar{\Omega}_i$, $\bar{\Omega}_j$, $i \neq j$ *is empty,*

$$\begin{cases} \Delta u(x) = \psi_0(x), & \text{in } \Omega, \\ \frac{\partial u(x)}{\partial n_s} = g(x), & \text{on } \partial\Omega, \\ \int_{\partial\Omega} g(x(s))ds = \int_{\Omega} \psi_0(x)dx, \end{cases}$$

using the single layer potential.
Given Ω *defined through its signed distance function* $d(x)$ *and* $\epsilon > 0$,

1. *Define tubular neighborhood* T_ϵ *and project the points from* T_ϵ *onto* $\partial\Omega$*:*

$$x_i^* = x_i - d_i \nabla_h d_i, \ x_i \in T_\epsilon.$$

2. *Form the matrix* $A = (C - \frac{1}{2}I)$ *where*

$$C_{i,j} = -h^n \delta_j J_j \left(\frac{\partial \Phi_{i,j}^*}{\partial n_j^*}_{reg} \right)^T,$$

and the vector g^* *such that*

$$g_i^* = g(x_i^*) - h^n \sum_{j \in \mathbb{L}} \frac{\partial \Phi(x_i^*, y_j)}{\partial n_{x_i^*}} \psi_0(y_j) H(d_j), \quad x_i \in T_\epsilon.$$

Here $\frac{\partial \Phi_{i,j}^*}{\partial n_j^*}_{reg}$ *is defined in* (26) *and* (30) *for the two dimensional and three dimensions cases respectively.*

3. *Replace* m *rows of the matrix* A *according to* (25) *and form the new matrix* A_m*.*
4. *Solve the system* $A_m \alpha^* = g^*$ *using a bi-conjugate gradient method.*
5. *Form the matrix* $\tilde{A}$ *such that*

$$\bar{A}_{i,j} = -h^n \delta_j J_j \Phi(x_i, y_j^*) \quad \text{for } x_i \in [a,b]^n, \ x_j \in T_\epsilon.$$

6. *Construct the solution* u *as* $u = \bar{A}\alpha^*$*.*

Remark: In practice, the matrices A, A_m, $\tilde{A}$ and $\bar{A}$ are never assembled since their storage requires a significant amount of memory which will limit the size of the problem that can be computed. Instead, only the matrix-vector products $A\beta^*$ and $A_m\alpha^*$ are evaluated in the iterative solver for the inversion, and $\tilde{A}\beta^*$ and $\bar{A}\alpha^*$ in the reconstruction of the solution. These computations can be further sped up with the use of Fast Multipole Methods.

In the next section we present our numerical results on various domains in two and three dimensions.

5 Numerical results

In this section we investigate the convergence of our numerical quadrature in the integration of (19), as well as the convergence of the complete algorithm in two and three dimensions. In the computations we use two different averaging kernels:

$$\delta_\epsilon^{cos}(x) = \begin{cases} \frac{1}{2\epsilon}\left(1 + \cos(\frac{\pi x}{\epsilon})\right) & \text{if } |x| \leq \epsilon, \\ 0 & \text{else,} \end{cases} \tag{31}$$

and

$$\delta_\epsilon^{hat}(x) = \begin{cases} \frac{1}{\epsilon} - \frac{|x|}{\epsilon^2} & \text{if } |x| \le \epsilon, \\ 0 & \text{else.} \end{cases} \tag{32}$$

Both of these kernels have one vanishing moment.

5.1 Convergence studies

We start by presenting convergence studies of the numerical quadrature used in the evaluation of integral (19). The convergence of the numerical evaluation of integral (18) is similar. In the study we use the exact density β and compare the accuracy of the numerical integration using various approximations of the Jacobian J_η. We denote $J_\eta^{(0)} = 1$, $J_\eta^{(1)} = 1 + \eta H_\eta$ where H_η is the mean curvature at a point on the η level set of d. In three dimensions we should further consider $J_\eta^{(2)} = 1 + 2\eta H_\eta + \eta^2 G_\eta$, where G_η is the Gaussian curvature at a point on the η level set of d. We first present the convergence of the numerical integration for a fixed width ϵ of the tubular neighborhood T_ϵ. We then present the errors produced by the numerical integration on a fixed grid as the width of the tubular neighborhood increases. In the numerical integration the solution is evaluated at one point far away from the boundary and compared with the value of the exact solution at that same point.

Eventually we focus on the complete algorithm presented in Section 4. We present the convergence of the density β, which is the solution of the integral equation, as the grid size increases using various Jacobians. We also present the accuracy of computed solutions to a few Poisson's equations. To see the behavior of our algorithm using extremely thin tubular neighborhoods around the interface, in most of the computations listed below we use ϵ that scales as

$$\epsilon = 2|\nabla d|_1 h, \tag{33}$$

where d is the signed distance function to the interface, h is the grid size, and $|\cdot|_1$ denotes the ℓ_1 norm of a vector in $\mathbb{R}^n$. This choice of width for the tubular neighborhood is motivated by the results of Engquist etal. [15] on convergent approximation of surface integrals on Cartesian grids. We present our convergence studies in two and three dimensions. In each of the studies we measure the relative error between the exact and computed solution inside the domain Ω, as well as the relative error between the exact and computed density α or β. In all the computations we use the double layer potential formulation to obtain the solution of the Dirichlet problem. All the computations with the complete algorithm use the exact Jacobian in the integrations.

In the following presentation, the computational results are obtained using uniform grids on $[-1, 1]^n$ for $n = 2, 3$, and the relative errors computed by the proposed algorithm are reported.

Two dimensions

Most of the numerical experiments presented in this section involve the exact solution to Laplace's equation on a circle with Dirichlet and Neumann boundary conditions. For clarity in the exposition of our results, we describe the calculations of the exact solution and the exact density for Laplace's equation on a circle with Dirichlet and Neumann boundary conditions. We first start with the exact solution to Laplace's equation on a circle with Dirichlet boundary conditions. The exact solution is obtained using separation of variables and expressed in polar coordinates for a circle centered at $(0, 0)$ as

$$u_e(r, \theta) = a_0 + \sum_{n=1}^{\infty} r^n \left(a_n \cos(n\theta) + b_n \sin(n\theta) \right), \tag{34}$$

where a_n, b_n are real numbers. In the general case where the circle is centered at $c = (c_x, c_y)$ the solution for $x, y \in \mathbb{R}^2$ is obtained using (34) with $x = c_x + r\cos\theta$, and $y = c_y + r\sin\theta$. In

this simple case the double layer solution v_{dl} is obtained from the expression of u_e in (34) as

$$v_{dl}(r,\theta) = -\sum_{n=1}^{\infty} \left(\frac{R^2}{r}\right)^n (a_n \cos(n\theta) + b_n \sin(n\theta)),$$

for a_n, b_n given in (34). The exact density β_e is given by

$$\beta_e(x) = u_e(x) - v_{dl}(x). \tag{35}$$

In the computations presented in this paper, we use $a_0 = 0, a_1 = -7, b_1 = 2$, $a_2 = 15, b_2 = 13, a_3 = 19, b_3 = 16, a_4 = -14, b_4 = -9$, and $a_n = b_n = 0$ for $n > 4$.

The exact solution to the Neumann problem is also given by (34) but since we are solving the Neumann problem we use the single layer potential formulation. In this simple case the exact exterior solution v_{sl} is expressed as

$$v_{sl}(r,\theta) = \sum_{n=1}^{\infty} \left(\frac{R^2}{r}\right)^n (a_n \cos(n\theta) + b_n \sin(n\theta)) = -v_{dl}(r,\theta),$$

for a_n, b_n given in (34). The exact density α_e is given by

$$\alpha_e(x) = \frac{\partial v_{sl}(x)}{\partial n_x} - \frac{\partial u_e(x)}{\partial n_x}, \tag{36}$$

where n_x is the outward unit normal to the circle at the point x. In Example 5.5 we use the same values for the constant a_n and b_n as the ones chosen in Example 5.4.

Example 5.1 *Convergence of the numerical integration of* (19)

We present the convergence of our numerical quadrature when, (a) the width of the tubular neighborhood T_ϵ is fixed and the number of grid points is increasing, and (b) the grid size is fixed but the width of the tubular neighborhood is increasing. For this study we use a circle as the interface, and the solution is computed at one point away from the boundary. The solution is obtained using the exact value for the density β given in (35), as well as the exact normal derivative of the fundamental solution $\frac{\partial \Phi}{\partial n_y}$. In the computations we take δ_ϵ to be the cosine kernel given in (31). The results are displayed in Figure 1 and Table 1 where we compare the errors using $J_\eta^{(0)}$ and $J_\eta^{(1)}$. The errors are very similar between using $J_\eta^{(0)}$ and $J_\eta^{(1)}$ and this is due to the fact that the cosine kernel has moment one therefore making any first order contribution in η irrelevant. We have observed numerically that in two dimensions the contribution of the curvature correction does not make much of a difference, but it does lower the errors slightly in the complete algorithm.

Example 5.2 *Study of the condition numbers for the inversion step*

In this example we demonstrate the effect of the regularization of the normal derivative of the fundamental solution on the condition number of the matrices that are assembled in Algorithms 1 and 2. The results are displayed in Table 2 when the interface is a circle. We see that the regularization lowers the condition number of the matrix significantly.

Example 5.3 *Convergence of the density* β

We present the convergence of the density β obtained for a circle using the double layer potential formulation. For this study we use the exact normal derivative of the fundamental solution $\frac{\partial \Phi}{\partial n_y}$. In the computations we use a constant width of the tubular neighborhood ϵ and take the averaging kernel δ_ϵ to be the cosine function (31). In Figure 2 we see that the errors with $J_\eta^{(0)}$ and $J_\eta^{(1)}$ are very similar. This is due to the fact that the averaging kernel has moment one.

Example 5.4 *Solution of Laplace's equation on a circle with Dirichlet boundary conditions*

We present the convergence of Algorithm 1 for the solution of Laplace's equation on a circle subject to Dirichlet boundary conditions. The convergence results on this example are presented in Figure 3. Figure 5(a) shows the computed solution of Laplace's equation on a circle of radius $R = 0.7$ centered at $(0.0061, 0.0061)$ subject to Dirichlet boundary conditions.

Example 5.5 *Solution of Laplace's equation on a circle with Neumann boundary conditions*

We present the convergence of Algorithm 2 for the solution of Laplace's equation on a circle subject to Neumann boundary conditions. The convergence results are displayed in Figure 3.

Example 5.6 *Solution of Poisson's equation on a flower domain with Dirichlet boundary conditions*

We present the convergence of Algorithm 1 for the solution of Poisson's equation on a flower domain subject to Dirichlet boundary conditions where the exact solution is given by

$$u_e(x, y) = x^6 + y^6 + \sin(\pi x) + \sin(\pi y) + \cos(\pi x) + \cos(\pi y).$$

This example was used in the work of Gibou and Fedkiw in [19]. The convergence results from this example are displayed in Figure 4. In Figure 5(b) we show the computed solution of Poisson's equation on the flower domain subject to Dirichlet boundary conditions.

Example 5.7 *Solution of Laplace's equation with Dirichlet boundary conditions on a domain whose boundary contains cusps*

Figure 5(c) shows the computed solution of Laplace's equation on a domain whose boundary contains cusps. In these computations we used constant Dirichlet boundary conditions where the constant was equal to 1.

Example 5.8 *Solution of Laplace's equation on a circle with mixed boundary conditions*

Figure 6 shows the computed solution of Laplace's equation on a circle subject to boundary conditions of the form given in (8). In these computations we chose $g(x) = 1$, $\sigma(x) = \frac{1}{4}\mathbf{1}_{\Gamma_0}(x)$ and $\rho(x) = \frac{1}{10}\mathbf{1}_{\partial\Omega\setminus\Gamma_0}(x)$ where $\partial\Omega$ was the circle and Γ_0 the left half of the circle. This choice is equivalent to imposing mixed boundary conditions.

Three dimensions

As in the two dimensional case we first describe the calculations of the exact solution and the exact density for Laplace's equation on a sphere with Dirichlet and Neumann boundary conditions. We first start with the exact solution to Laplace's equation on a sphere with Dirichlet boundary conditions. The exact solution can be obtained using separation of variables and expressed in spherical coordinates, for a sphere centered at $(0, 0, 0)$, as

$$u_e(r, \theta, \varphi) = \sum_{l=0}^{\infty} r^l \sum_{m=0}^{l} \left(a_{lm}\cos(m\varphi) + b_{lm}\sin(m\varphi)\right) f_l^m(\cos\theta), \tag{37}$$

where $a_{lm}, b_{lm} \in \mathbb{R}$ and f_l^m are the Legendre functions satisfying the ODE

$$\frac{d}{dx}\left((1-x^2)f'(x)\right) + \left(l(l+1) - \frac{m^2}{1-x^2}\right) f(x) = 0, \qquad l > 0, m \in \mathbb{N},$$

with the conditions that f should remain finite at the end points $x = 1$ and $x = -1$ corresponding to $\theta = 0$ and $\theta = \pi$ through the change of variables $x = \cos\theta$. These finite conditions can only be satisfied if $l \in \mathbb{N}^*$ and $m \leq l$. The solutions f_l^m are derived from the Legendre polynomials P_l by the formula

$$f_l^m(x) = (-1)^m (1-x^2)^{\frac{m}{2}} \frac{d^m}{dx^m} P_l(x).$$

In the general case where the sphere is centered at (c_x, c_y, c_z) the exact solution of Laplace's equation in $(x, y, z) \in \mathbb{R}^3$ is obtained using (37) with $x = c_x + r\sin\theta\cos\varphi$, $y = c_y + r\sin\theta\sin\varphi$ and $z = c_z + r\cos\theta$. Since the boundary conditions are of Dirichlet type we use the double layer potential formulation with exact exterior solution v_{dl} given by

$$v_{dl}(r,\theta,\varphi) = -\sum_{l=0}^{\infty} \frac{l}{l+1} \frac{R^{2l+1}}{r^{l+1}} \sum_{m=0}^{l} \left(a_{lm}\cos(m\varphi) + b_{lm}\sin(m\varphi)\right) f_l^m(\cos\theta), \tag{38}$$

for a_{lm}, b_{lm} as in (37). The exact density β_e is given by (35), where u_e is given by (37) and v_{dl} by (38). In these computations we use $a_{00} = 0, a_{10} = -7, a_{11} = 3, b_{11} = 8, a_{20} = -5$, $a_{21} = 3, a_{22} = 5, b_{21} = -5, b_{22} = -4, a_{30} = 6, a_{31} = -9, a_{32} = 7, a_{33} = 1, b_{31} = 4$, $b_{32} = -4, b_{33} = 8$ and $a_{lm} = b_{lm} = 0$ for $l > 3, m \leq l$.

The exact solution to the Neumann problem is the same as for the Dirichlet problem and is given by (37). We use the single layer potential formulation. In this simple case the exact exterior solution v_{sl} is obtained from the interior solution u_e and expressed as

$$v_{sl}(r,\theta,\varphi) = \sum_{l=0}^{\infty} \frac{R^{2l+1}}{r^{l+1}} \sum_{m=0}^{l} \left(a_{lm}\cos(m\varphi) + b_{lm}\sin(m\varphi)\right) f_l^m(\cos\theta), \tag{39}$$

for a_{lm}, b_{lm} as in (37). The exact density α_e is given by (36), where v_{sl} is given by (39) and u_e by (37). We use the same values of a_{lm} and b_{lm} as for the Dirichlet problem given above.

Example 5.9 *Convergence of the numerical integration of* (19) *when x is away from the boundary*

We present the convergence of our numerical quadrature when, (a) the width of the tubular neighborhood is fixed and the number of grid points is increasing, and (b) the grid size is fixed but the width of the tubular neighborhood is increasing. For this study we use a sphere as the interface, and the solution is computed at one point away from the boundary. The solution is obtained using the exact value for the density β given in (35), where u_e is given by (37) and v_{dl} by (38), as well as the exact normal derivative of the fundamental solution $\frac{\partial \Phi}{\partial n_y}$. In the computations we take δ_ϵ to be the cosine kernel (31). The results are displayed in Table 4 and Figure 7 where we compare the errors using $J_\eta^{(0)}$, $J_\eta^{(1)}$ and $J_\eta^{(2)}$.

The results in Figure 7 show that the errors in the solution obtained with $J_\eta^{(0)}$ and $J_\eta^{(1)}$ quickly, and already at very coarse grids, saturate to a relatively small magnitude of the order of 10^{-4}. This is due to the fact that the errors are dominated by the analytical error (we are not using the correct Jacobian) which scales with ϵ. Since ϵ is fixed in these computations, the errors with $J_\eta^{(0)}$ and $J_\eta^{(1)}$ are also stationary. Indeed, as Table 4 shows, the error gets larger as ϵ increases. On the other hand the computations with the correct Jacobian $J_\eta^{(2)}$ display a decrease in the error in the solution as the resolution increases.

Example 5.10 *Convergence of the numerical integration of* (19) *when x is on the interface*

As in Example 5.9, we present the convergence of our numerical quadrature when, (a) the width of the tubular neighborhood is fixed and the number of grid points is increasing, and (b) the grid size is fixed but the width of the tubular neighborhood is increasing. For this study we use a sphere as the interface, and the integral is evaluated at one point on the boundary. In this example we take the density $\beta = 1$ and use the result of Theorem B.4 in Appendix B to compare the computed value with the exact value of $\frac{1}{2}$. The purpose of this study is to test the effect the regularization of the normal derivative of the fundamental solution on the result of the integration. In the computations, we take δ_ϵ to be the cosine kernel (31). The results are displayed in Figure 8 where we compare the errors using $J_\eta^{(0)}$, $J_\eta^{(1)}$ and $J_\eta^{(2)}$.

As in Example 5.9, the results in Figure 8 show that the errors in the solution obtained with $J_\eta^{(0)}$ and $J_\eta^{(1)}$ are basically constant as the grid spacing decreases. This is due to the fact that the errors are dominated by the analytical error which scales with ϵ. Since ϵ is fixed in these computations, the errors with $J_\eta^{(0)}$ and $J_\eta^{(1)}$ are also stationary. On the other hand, when the exact Jacobian $J_\eta^{(2)}$ is used, the errors become much smaller and seem to converge with a globally third order trend.

Example 5.11 *Study of the condition numbers for the inversion step when the interface is made of several connected components*

In this example we study the condition number of the matrices assembled in Algorithms 1 and 2 when the interface is made of several connected components. We compare the condition number of these matrices when the tangent and the paraboloid regularizations are used. We display the computed condition numbers in Table 3 in the case where the interface consists of two disjoint spheres.

Example 5.12 *Solution of Laplace's equation on a sphere with Dirichlet boundary conditions*

We present the convergence of Algorithm 1 for the solution of Laplace's equation on a sphere subject to Dirichlet boundary conditions. The convergence results are displayed in Figure 9.

Example 5.13 *Solution of Laplace's equation on a sphere with Neumann boundary conditions*

We present the convergence of Algorithm 2 for the solution of Laplace's equation on a sphere subject to Neumann boundary conditions. The convergence results are displayed in Figure 9.

Example 5.14 *Solution of Poisson's equation on an ellipsoid with Dirichlet boundary conditions*

We present the convergence of Algorithm 1 for the solution of Poisson's equation on an ellipsoid subject to Dirichlet boundary conditions. In our computations we use the ellipsoid described by the equation $\frac{(x-c_x)^2}{a^2} + \frac{(y-c_y)^2}{b^2} + \frac{(z-c_z)^2}{c^2} = 1$, with $c_x = 0.02$, $c_y = -0.026$, $c_z = 0.012$, $a = 0.784$, $b = 0.465$ and $c = 0.634$. The exact solution of Poisson's equation is taken to be

$$u_e(x, y, z) = x^4 + y^4 + z^4 + \cos x + \cos z.$$

The convergence results are displayed in Figure 10.

Table 1: Convergence of the numerical integration of (19) with the exact value of the density β. In these convergence studies we used a fixed resolution of 513^2 and took the averaging kernel to be the cosine function (31). The interface was chosen to be a circle and the error in the solution was measured at a point far away from the interface. This table refers to Example 5.1.

Epsilon	Error in the solution with $J_\eta^{(0)}$	Order	Error in the solution with $J_\eta^{(1)}$	Order
ϵ_0	$5.306304988 \times 10^{-8}$	–	$8.836347155 \times 10^{-8}$	–
$2\epsilon_0$	$2.861647354 \times 10^{-8}$	0.89	$3.504417392 \times 10^{-8}$	1.33
$4\epsilon_0$	$6.320256577 \times 10^{-9}$	2.18	$8.006345883 \times 10^{-9}$	2.13

Table 2: Condition number for the matrix built for solving Laplace's equation with Dirichlet and Neumann boundary conditions using the double layer potential. The exact curvature correction is used. We denote by C_0 the condition number without regularization of the fundamental solution and by C_{reg} the condition number with regularization. In these computations the interface is one circle. τ is to the tolerance used to determine the onset of the regularization. This table refers to Example 5.2.

τ	C_0,Dirichlet BC	C_{reg}, Dirichlet BC	C_0, Neumann BC	C_{reg},Neumann BC
$n = 128^2$				
$4dx$	11.6427	6.9807	13.7324	9.7054
dx	11.6427	6.9808	13.7324	9.7053
$n = 1024^2$				
$4dx$	107.4301	8.0026	103.6829	11.4458
dx	107.4301	8.0026	103.6829	11.4458

Table 3: Condition number for the matrix built for solving Laplace's equation with Dirichlet and Neumann boundary conditions using the double layer potential. The exact curvature correction is used. We denote by C_{regT} the condition number with the tangent regularization and C_{regP} the condition number with the paraboloid regularization. In these computations the interface consists of two disjoint spheres. This table refers to Example 5.2.

τ	C_{regT}, Dirichlet BC	C_{regP}, Dirichlet BC	C_{regT}, Neumann BC	C_{regP}, Neumann BC
$n = 50^3$				
$4dx$	6.0734	8.1287	12.6043	24.0906
dx	7.0869	8.0396	20.3432	24.3680
$n = 80^3$				
$4dx$	6.3995	7.9241	14.8948	23.5693
dx	7.1536	7.8566	21.1952	23.1743

Table 4: Convergence of the numerical integration of (19) with the exact value of the density β. In these convergence studies we used a fixed resolution of 200^3 and took the averaging kernel to be the cosine function (31). The interface was chosen to be a sphere and the error in the solution was measured at a point far away from the interface. This table refers to Example 5.9.

Epsilon	Error with $J_\eta^{(0)}$	Order	Error with $J_\eta^{(1)}$	Order	Error with $J_\eta^{(2)}$	Order
ϵ_0	0.000216219	–	0.000216315	–	$1.313118131 \times 10^{-7}$	–
$2\epsilon_0$	0.000865085	−2.00	0.000865090	−2.00	$5.425436281 \times 10^{-8}$	1.28
$4\epsilon_0$	0.003460350	−2.00	0.003460350	−2.00	$6.553840441 \times 10^{-9}$	3.05

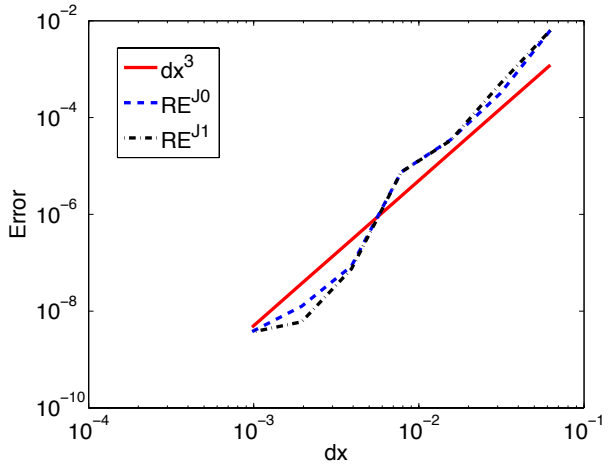


Figure 1: Convergence of the numerical integration of (19) with the exact value of the density β as described in Example 5.1. In these convergence studies we used a constant width of the tubular neighborhood ϵ and took the averaging kernel to be the cosine function (31). The interface was chosen to be a circle and the error in the solution was measured at a point far away from the interface. This is a loglog plot of the relative error in the solution computed using $J_\eta^{(0)}$ and $J_\eta^{(1)}$. This figure refers to Example 5.1.

Table 5: Convergence of the numerical integration of (19) with the exact value of the density β. In these convergence studies we used a fixed resolution of 200^3 and took the averaging kernel to be the cosine function (31). The interface was chosen to be a sphere and the error in the solution was measured at a point on the interface. This table refers to Example 5.10.

Epsilon	Error with $J_\eta^{(0)}$	Order	Error with $J_\eta^{(1)}$	Order	Error with $J_\eta^{(2)}$	Order
ϵ_0	0.003980277628161	–	0.003675494063089	–	$1.41307015 \times 10^{-4}$	–
$2\epsilon_0$	0.015319781365319	−1.94	0.015229122258640	−2.05	3.8369953×10^{-5}	1.88
$4\epsilon_0$	0.061092589758195	−2.00	0.061058214575044	−2.00	1.0870387×10^{-5}	1.82

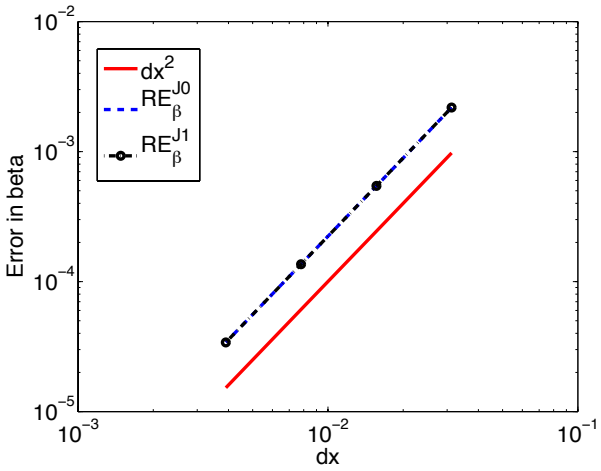


Figure 2: Convergence of the density β as described in Example 5.3. In these convergence studies we used a constant width of the tubular neighborhood ϵ and took the averaging kernel to be the cosine function (31). The interface was chosen to be a circle and the error in β was computed using $J_\eta^{(0)}$ and $J_\eta^{(1)}$. This is a loglog plot of the relative errors in the solution computed at a point away from the interface using $J_\eta^{(0)}$ and $J_\eta^{(1)}$. These two errors are so similar that they line up perfectly (dashed line). This figure refers to Example 5.3.

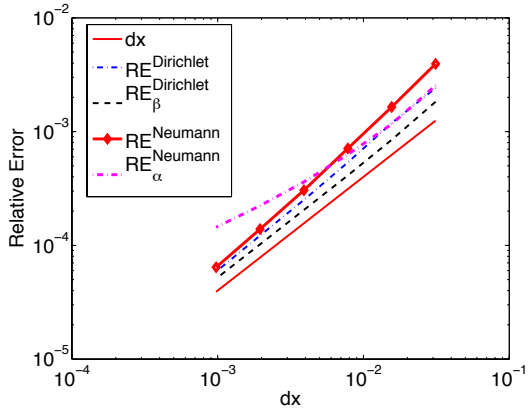


Figure 3: Convergence of Algorithm 1 and Algorithm 2 for the solution of Laplace's equation on a circle with Dirichlet and Neumann boundary conditions as presented in Example 5.4. In these computations the averaging kernel δ_ϵ was taken to be the hat function (32), the width of the tubular neighborhood was $\epsilon = 2|\nabla d|h$ and the tolerance for the regularization of the normal derivative of the fundamental solution was $\tau = \frac{h}{5}$. This loglog plot displays the relative errors in the solution, and in the densities β (with Dirichlet boundary conditions) and α (with Neumann boundary conditions)

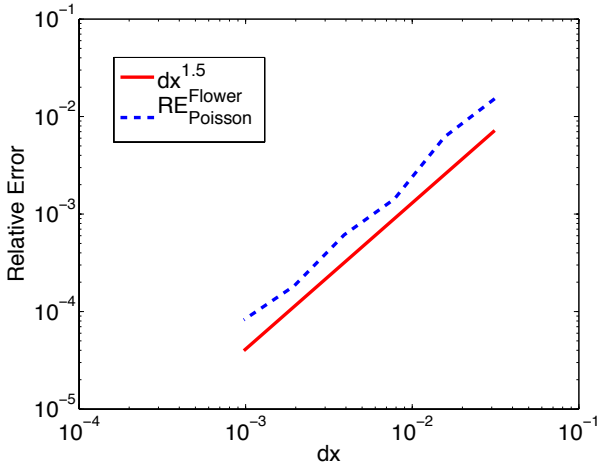
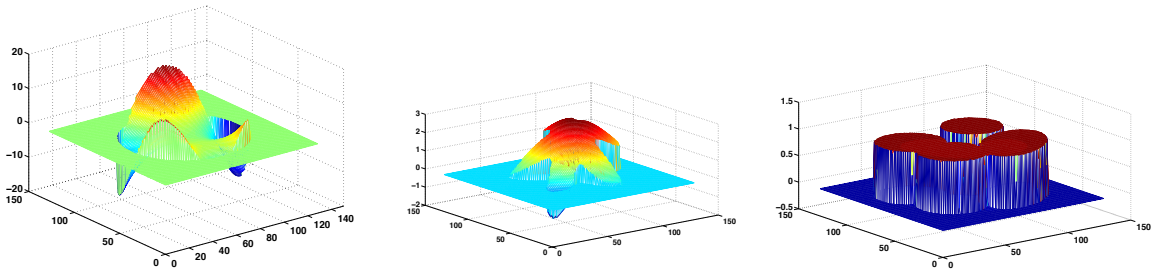


Figure 4: Convergence of Algorithm 1 on a flower domain. In these computations, the averaging kernel δ_ϵ was taken to be the hat function, the width of the tubular neighborhood was $\epsilon = 2|\nabla d|_1 h$ and the tolerance for the regularization of the normal derivative of the fundamental solution was $\tau = \frac{h}{5}$. This loglog plot shows the relative error in the solution. This figure refers to Example 5.6.



(a) Computed solution of Laplace's equation on a circle

(b) Computed solution of Poisson's equation on a flower

(c) Computed solution Laplace's equation on a domain with cusps

Figure 5: 5(a): Computed solution of Laplace's equation with Dirichlet boundary conditions on a circle. 5(b): Computed solution of Poisson's equation with Dirichlet boundary conditions on a flower domain. 5(c): Computed solution of Laplace's equation with constant Dirichlet boundary conditions on a domain containing cusps. In these computations the averaging kernel δ_ϵ was taken to be the hat function (32), the width of the tubular neighborhood was $\epsilon = 2|\nabla d|_1 h$ and the tolerance for the regularization of the normal derivative of the fundamental solution was $\tau = \frac{h}{5}$. The computations were performed on a 512 by 512 grid and the solution was reconstructed on a 128 by 128 grid. This figure refers to Examples 5.4, 5.6 and 5.7.

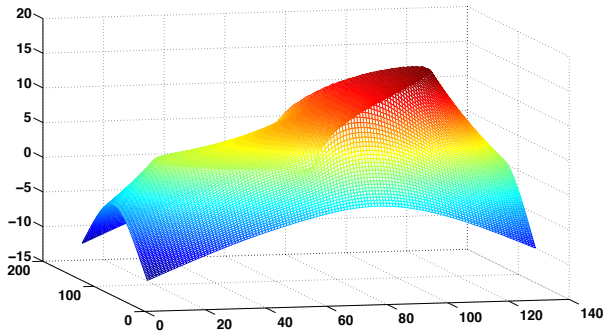


Figure 6: Computed solution of Laplace's equation with mixed boundary conditions. In these computations the averaging kernel δ_ϵ was taken to be the cosine function (31), the width of the tubular neighborhood was $\epsilon = 2h$ and the tolerance for the regularization of the normal derivative of the fundamental solution was $\tau = h$. The computations were performed on a 128 by 128 grid and the solution was reconstructed on a 128 by 128 grid. This figure refers to Example 5.8.

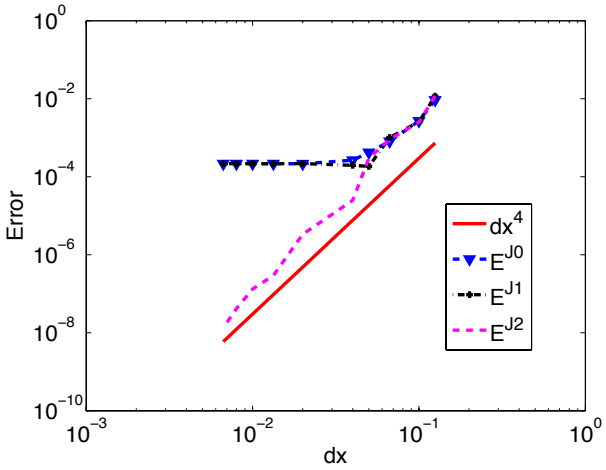


Figure 7: Convergence of the numerical integration of (19) with the exact value of the density β. In these convergence studies we used a constant width of the tubular neighborhood ϵ and took the averaging kernel to be the cosine function (31). The interface was chosen to be a sphere and the error in the solution was measured at a point far away from the interface. This figures shows the loglog plot of the error in the solution computed using $J_\eta^{(0)}$, $J_\eta^{(1)}$ and $J_\eta^{(2)}$. We see that when $J_\eta^{(0)}$ and $J_\eta^{(1)}$ are used, the error saturates, but is still quite small (around 10^{-4}). When the correct Jacobian $J_\eta^{(2)}$ is used, the error seems to be fourth order accurate. This figure refers to Example 5.9

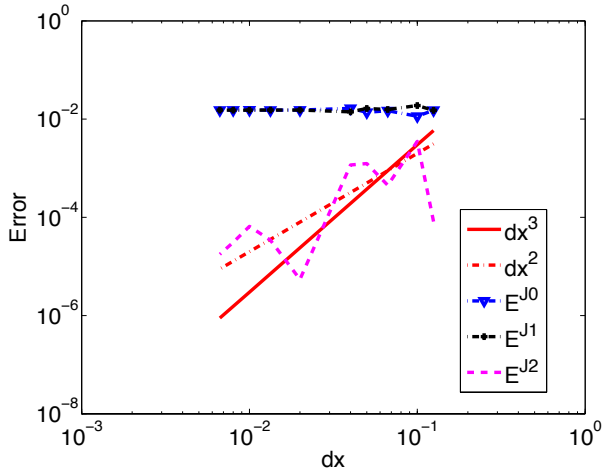


Figure 8: Convergence of the numerical integration of (19) with the exact value of the density β. In these convergence studies we used a constant width of the tubular neighborhood ϵ and took the averaging kernel to be the cosine function (31). The interface was chosen to be a sphere and the error in the solution was measured at a point on the interface. This figure is a loglog plot of the error in the solution computed at a point on the interface using $J_\eta^{(0)}$, $J_\eta^{(1)}$ and $J_\eta^{(2)}$. We see that if $J_\eta^{(0)}$ and $J_\eta^{(1)}$ are used the error remains stationary around 10^{-2}. On the other hand, if the correct Jacobian $J_\eta^{(2)}$ is used, the error seems to follow a third order accuracy trend. This figure refers to Example 5.10.

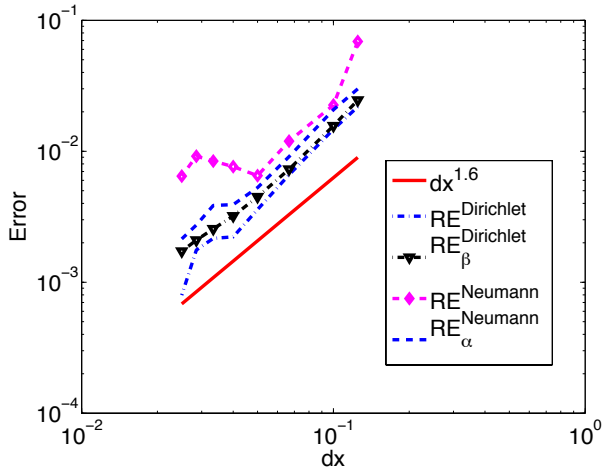


Figure 9: Convergence of Algorithm 1 for the solution of Laplace's equation on a sphere with Dirichlet boundary conditions and Algorithm 2 for the solution of Laplace's equation on a sphere with Neumann boundary conditions. In these computations the averaging kernel δ_ϵ was taken to be the hat function (32), the width of the tubular neighborhood was $\epsilon = 2|\nabla d|_1 h$ and the tolerance for the regularization of the normal derivative of the fundamental solution was $\tau = h$. This loglog plot displays the relative errors in the solution and in the densities β (with Dirichlet boundary conditions) and α (with Neumann boundary conditions). This figure refers to Examples 5.12 and 5.13.

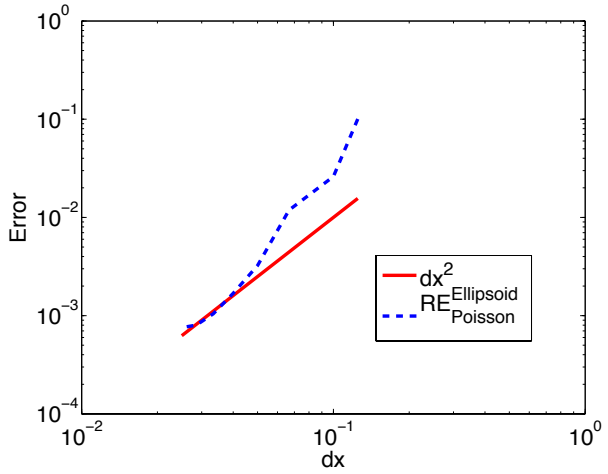


Figure 10: Convergence of Algorithm 1 for the solution of Poisson's equation on an ellipsoid. In these computations the averaging kernel δ_ϵ was taken to be the cosine function (31), the width of the tubular neighborhood was $\epsilon = 2h$ and the tolerance for the regularization of the normal derivative of the fundamental solution was $\tau = h$. This figure refers to Example 5.14.

6 Conclusion

We proposed a formulation for computing integrals of the form $\int_{\partial\Omega} v(x(s))ds$ in the level set framework and presented an implicit boundary integral method for solving Poisson's equation in domains of any shape. Our algorithm is based on the solution of an integral equation on the domain boundary, which is implicitly defined by a signed distance function. One of the main advantages of our proposed algorithm is its flexibility and simplicity of implementation. Indeed, our algorithm can solve Poisson's equation on any domain with various boundary conditions (i.e. Neumann, Dirichlet, Robin and mixed boundary conditions) and can also solve the interior and exterior problem with no additional changes. The other main advantage of our proposed algorithm is that it is grid independent, thus eliminating the need to compute intersection points between the domain boundary and the grid. One immediate consequence is that this algorithm handles complicated domains and moving interfaces easily. The other consequence is that local level set techniques can be incorporated into our algorithm with almost no modification. Furthermore, our algorithm is compatible with Fast Multipole Methods and other established computational techniques that can be used to further improve its numerical efficiency.

Acknowledgments

The authors gratefully acknowledge the support of NSF grants DMS-0914465 and DMS-0914840. Catherine Kublik was in part supported by a Bing Fellowship. The authors thank Dan Knopf and Mar González for helpful discussions.

A Jacobian for the integration over an offset hypersurface

Let Ω be a n dimensional domain ($n = 2, 3$) such that its boundary $\partial\Omega$ is of smoothness of class C^2. Then for each $x \in \partial\Omega$ there is a neighborhood $\mathcal{N}(x)$ of x on which the signed distance function to the boundary $\partial\Omega$, denoted by $d(x)$, is a C^2 function. Thus, at any point $x \in \partial\Omega$, we can define the unit normal vector (outward by convention) $n(x)$. Moreover we have the following property:

Proposition A.1 *If d is differentiable at a point $x \in \mathbb{R}^n$, then there exists a unique $x^* \in \partial\Omega$, such that $d(x) = |x - x^*|$, and*

$$\nabla d(x) = \frac{x - x^*}{d(x)}.$$

x^* is called the projection of x onto $\partial\Omega$ and the projection map $x \mapsto x^*$ is a diffeomorphism.

Let $\epsilon > 0$ and consider $\partial\Omega_\eta$ for $\eta \in [-\epsilon, \epsilon]$, where $\partial\Omega_\eta := \{x : d(x) = \eta\}$. We assume that any $x \in \partial\Omega_\eta$ for all $\eta \in [-\epsilon, \epsilon]$ is included in a neighborhood on which the signed distance function d is C^2. In other words at any point $x \in T_\epsilon := \{x : |d(x)| \leq \epsilon\}$, the characteristics are straight lines and are normal to $\partial\Omega_\eta$ for any $\eta \in [-\epsilon, \epsilon]$.

A.1 Two dimensions

Consider the two integrals

$$\int_{\partial\Omega_\eta} \alpha(y^*(s))\zeta(x, y^*(s))ds, \tag{40}$$

and

$$\int_{\partial\Omega} \alpha(y^*(s))\zeta(x, y^*(s))ds, \tag{41}$$

where α is a continuous function defined on $\partial\Omega$ and ζ is a continuous function defined on $\mathbb{R}^2 \times \mathbb{R}^2$. Without loss of generality we assume that the length of the interface $\partial\Omega_\eta$ is 1 and let $s_\eta \in [0, 1] \mapsto \mathbb{R}$ be its arc length parameterization. Then

$$\int_{\partial\Omega_\eta} \alpha(y^*(s))\zeta(x, y^*(s))ds = \int_0^1 \alpha(y^*(s_\eta))\zeta(x, y^*(s_\eta))ds_\eta,$$

and

$$\int_{\partial\Omega} \alpha(y^*(s))\zeta(x, y^*(s))ds = \int_0^1 \alpha(y^*(s_\eta))\zeta(x, y^*(s_\eta))|y'(s_\eta)|ds_\eta.$$

The pointwise projection map can be written as

$$y^*(s_\eta) = y(s_\eta) - d(y(s_\eta))\nabla d(y(s_\eta)) = y(s_\eta) - d(y(s_\eta))n_{y(s_\eta)},$$

where $y(s_\eta) \in \partial\Omega_\eta$ and $n_{y(s_\eta)}$ is the inward unit normal to the curve $\partial\Omega$ ($n_{y(s_\eta)}$ is also the normal unit vector in the Frenet frame). Since s_η is the arc length parameterization of the curve $\partial\Omega$ it follows that $\tau_{s_\eta} = \tau'(y(s_\eta)) = \kappa(s_\eta)n_{y(s_\eta)} = \kappa(s_\eta)n(s_\eta)$, where $\tau(y(s))$ is the tangent vector to the curve $\partial\Omega$ at $y(s_\eta)$ and $n(s_\eta) = n_{y(s_\eta)}$. Moreover, since $y(s_\eta) \in \partial\Omega_\eta$, we have $d(y(s_\eta)) = \eta$, which gives

$$y^*(s_\eta) = y(s_\eta) - \eta\frac{\tau_{s_\eta}}{\kappa(s_\eta)}. \tag{42}$$

Differentiating (42) with respect to s_η we obtain

$$(y^*)'(s_\eta) = y'(s_\eta) - \eta\frac{\tau_{s_\eta s_\eta}\kappa(s_\eta) + \tau_{s_\eta}\kappa'(s_\eta)}{\kappa(s_\eta)^2},$$

which, using $\tau_{s_\eta} = \kappa(s_\eta)n(s_\eta)$ and $n_{s_\eta} = -\kappa(s_\eta)\tau(s_\eta)$, can be simplified as

$$(y^*)'(s_\eta) = y'(s_\eta) + \eta\kappa(s_\eta)\tau(s_\eta). \tag{43}$$

Since s_η is the arc length parameterization of $\partial\Omega_\eta$ it follows that $y'(s_\eta) = \tau_{s_\eta}$ and thus (43) can be rewritten as

$$(y^*)'(s_\eta) = (1 + \eta\kappa(s_\eta))\tau(s_\eta).$$

Consequently if η is chosen such that $\eta < \min_{x\in\partial\Omega_\epsilon} \frac{1}{\kappa_\epsilon(x)}$, we have

$$|(y^*)'(s_\eta)| = 1 + \eta\kappa(s_\eta) = 1 + \eta\kappa_\eta.$$

Thus

$$\begin{aligned}
\int_{\partial\Omega} \alpha(y^*(s))\zeta(x, y^*(s))ds &= \int_0^1 \alpha(y^*(s_\eta))\zeta(x, y^*(s_\eta))|y'(s_\eta)|ds_\eta \\
&= \int_0^1 \alpha(y^*(s_\eta))\zeta(x, y^*(s_\eta))\,(1 + \eta\kappa_\eta)\,ds_\eta \\
&= \int_{\partial\Omega_\eta} \alpha(y^*(s_\eta))\zeta(x, y^*(s_\eta))\,(1 + \eta\kappa_\eta)\,ds_\eta.
\end{aligned} \tag{44}$$

Using the signed distance function $d(z)$, we compute the curvature $\kappa(z)$ at a point $z \in \mathbb{R}^2$ sitting on $\partial\Omega_\eta$ as

$$\kappa(z) = \kappa_\eta = \Delta d(z).$$

A.2 Three dimensions

In this section we provide the reader with a simple and intuitive derivation of the change of variable for surfaces. We consider the two integrals

$$\int_{\partial\Omega_\eta} \alpha(y^*(s_\eta, \lambda_\eta)\zeta(x, y^*(s_\eta, \lambda_\eta))ds_\eta d\lambda_\eta,$$

and

$$\int_{\partial\Omega_\eta} \alpha(y^*(s, \lambda)\zeta(x, y^*(s, \lambda))dsd\lambda.$$

By a simple calculation we will relate the surface element $dsd\lambda$ to the surface element $ds_\eta d\lambda_\eta$. Pick a point x on the zero level set surface and consider its two principal directions. We assume without loss of generality that s is the parameterization of the first principal direction and λ the parameterization of the second. We assume also that the curvature along the first principal direction at x is κ_1 and that the curvature along the second principal direction at x is κ_2. In this situation we call R_1 the radius of the osculating circle associated to the first principal direction and R_2 the radius of the osculating circle associated to the second principal direction. From x we now consider a surface element $dsd\lambda$, where $ds = R_1\theta_1$ and $R_2\theta_2$. See Figure 11. Since the offset surface is defined as $\{x : d(x) = \eta\}$, the two principal curvatures of the offset surface element at $x_\eta = x - \eta\nabla d(x)$ (x_η is the projection of x onto the offset surface) are $\kappa_1^\eta = \frac{1}{R_1-\eta}$ and $\kappa_2^\eta = \frac{1}{R_2-\eta}$. Therefore the offset surface element can be expressed as $ds_\eta d\lambda_\eta$,

where $ds_\eta = (R_1 - \eta)\theta_1$ and $d\lambda_\eta = (R_2 - \eta)\theta_2$. Relating the two surface elements we have

$$\begin{aligned}
\frac{ds d\lambda}{ds_\eta d\lambda_\eta} &= \frac{R_1 R_2}{(R_1 - \eta)(R_2 - \eta)} \\
&= \frac{(R_1 - \eta)(R_2 - \eta) + \eta(R_1 - \eta + R_2 - \eta)}{(R_1 - \eta)(R_2 - \eta)} + \frac{\eta^2}{(R_1 - \eta)(R_2 - \eta)} \\
&= 1 + \left(\frac{1}{R_1 - \eta} + \frac{1}{R_1 - \eta}\right)\eta + \frac{\eta^2}{(R_1 - \eta)(R_2 - \eta)} \\
&= 1 + (\kappa_1^\eta + \kappa_2^\eta)\eta + \frac{\eta^2}{(R_1 - \eta)(R_2 - \eta)} \\
&= 1 + 2H_\eta \eta + O(\eta^2),
\end{aligned}$$

where $H_\eta = \dfrac{\kappa_\eta^{(1)} + \kappa_\eta^{(2)}}{2}$ is the mean curvature of the η level set at x_η. So it follows that if η is chosen such that $\eta < \min\limits_{x \in \partial\Omega_\epsilon} \dfrac{1}{H_\epsilon(x)}$, we have

$$\int_{\partial\Omega} \alpha(y^*(s,\lambda))\zeta(x, y^*(s,\lambda)) ds d\lambda = \int_{\partial\Omega_\eta} \alpha(y^*(s_\eta, \lambda_\eta))\zeta(x, y^*(s_\eta, \lambda_\eta)) \left((1 + 2\eta H_\eta)\, ds_\eta d\lambda_\eta + O(\eta^2)\right),$$

where H_η is the mean curvature of the offset surface at $y^*(s_\eta, \lambda_\eta)$.

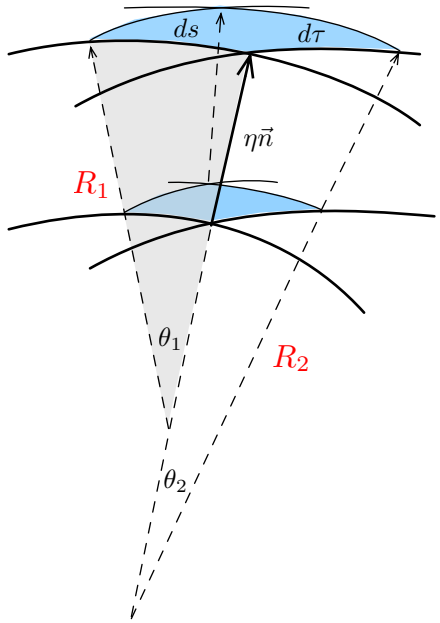


Figure 11: The two surface elements on the zero and the η level set surfaces.

The exact Jacobian in three dimensions is actually of degree 2 in η and so that the integral

becomes

$$\int_{\partial\Omega} \alpha(y^*(s,\lambda))\zeta(x,y^*(s,\lambda))dsd\lambda = \int_{\partial\Omega_\eta} \alpha(y^*(s_\eta,\lambda_\eta))\zeta(x,y^*(s_\eta,\lambda_\eta))\left(1+2\eta H_\eta+\eta^2 G_\eta\right)ds_\eta d\lambda_\eta, \tag{45}$$

where G_η is the Gaussian curvature of the offset surface at $y^*(s_\eta,\lambda_\eta)$.

Using the signed distance function $d(z)$, we compute the mean curvature $H(z)$ and the Gaussian curvature $\Gamma(z)$ at a point $z\in\mathbb{R}^3$ sitting on $\partial\Omega_\eta$ as

$$H(z) = H_\eta = \frac{1}{2}\Delta d(z),$$

and

$$\begin{aligned} G(z) = G_\eta &= \langle \nabla d, adj(Hess(d))\nabla d\rangle \\ &= d_x^2(d_{yy}d_{zz}-d_{yz}^2)+d_y^2(d_{xx}d_{zz}-d_{xz}^2)+d_z^2(d_{xx}d_{yy}-d_{xy}^2) \\ &\quad +2[d_xd_y(d_{xz}d_{yz}-d_{xy}d_{zz})+\ d_yd_z(d_{xy}d_{xz}-d_{yz}d_{xx})+d_xd_z(d_{xy}d_{yz}-d_{xz}d_{yy})], \end{aligned}$$

where $adj(Hess(d))$ is the adjugate matrix of the Hessian of d.

In the next section we present a detailed derivation of the closed formula for the change of variable in any dimension.

A.3 Closed formula for the Jacobian in any dimension

In this section we provide the reader with a sketch of the derivation of the complete change of variable in dimension $(n+1)$, $n\in\mathbb{N}^*$. The proof was obtained by Dan Knopf.

Consider the hypersurfaces $\partial\Omega$ and $\partial\Omega_\eta$ in $\mathbb{R}^{n+1}$, and a domain $\mathcal{U}\subset\mathbb{R}^{n+1}$ such that $\upsilon:\mathcal{U}\mapsto\mathbb{R}^n$ is a parameterization of $\partial\Omega_\eta$. Then the induced metric g on $\partial\Omega_\eta$ has components

$$g_{ij} = \langle D_i\upsilon, D_j\upsilon\rangle,$$

where $i,j=1,\cdots,n$, and $D_i\upsilon = D_i\upsilon(x) = (\partial_i\upsilon_1(x),\ldots,\partial_i\upsilon_{n+1}(x))$. If ν is the inward unit normal to $\partial\Omega_\eta$ we define the second fundamental form h as

$$h_{ij} = \langle D_i\nu, D_j\upsilon\rangle.$$

The area of $\partial\Omega_\eta$ is then computed by

$$A(\partial\Omega_\eta) = \int_{\mathcal{U}} \sqrt{\det\, g}dx,$$

where dx is a Lebesgue measure on $\mathbb{R}^n$. From the parameterization of $\partial\Omega_\eta$ we obtain a parameterization of $\partial\Omega$ as

$$\tilde{\upsilon}(x) = \upsilon(x)+\eta\nu(\upsilon(x)),$$

with induced metric

$$\tilde{g}_{ij} = \langle D_i\upsilon+\eta D_i\nu, D_j\upsilon+\eta D_j\nu\rangle = g_{ij}+2\eta h_{ij}+\eta^2\langle D_i\nu, D_j\nu\rangle.$$

Using the fact that $D_i\nu = h_{ik}g^{kl}D_l\upsilon$, (using Einstein summation convention), we obtain that

$$\begin{aligned} \langle D_i\nu, D_j\nu\rangle &= \langle h_{ik}g^{kl}D_l\upsilon, h_{jm}g^{mr}D_r\upsilon\rangle \\ &= h_{ik}g^{kl}h_{jm}g^{mr}\underbrace{\langle D_l\upsilon, D_r\upsilon\rangle}_{=g_{lr}} \\ &= h_{ik}g^{kl}h_{jm}\underbrace{g^{mr}g_{lr}}_{=\delta_l^m} \\ &= h_{ik}g^{kl}h_{jl} = h*g*h. \end{aligned}$$

Thus we obtain the tensor identity

$$\tilde{g} = g + 2\eta h + \eta^2 h * g * h.$$

By applying g^{-1} to the above equation we obtain

$$(g^{-1}\tilde{g})_i^j = \delta_i^j + 2\eta h_i^j + \eta^2 h_i^k h_k^j,$$

in which we can now diagonalize the Weingarten map induced by h to obtain

$$\begin{aligned} P^{-1}\left(g^{-1}\tilde{g}\right)P &= I_n + 2\eta \begin{pmatrix} \kappa_\eta^{(1)} & 0 & \cdots & 0 \\ 0 & \kappa_\eta^{(2)} & 0 & 0 \\ 0 & \cdots & \cdots & 0 \\ 0 & \cdots & 0 & \kappa_\eta^{(n)} \end{pmatrix} + \eta^2 \begin{pmatrix} (\kappa_\eta^{(1)})^2 & 0 & \cdots & 0 \\ 0 & (\kappa_\eta^{(2)})^2 & 0 & 0 \\ 0 & \cdots & \cdots & 0 \\ 0 & \cdots & 0 & (\kappa_\eta^{(n)})^2 \end{pmatrix} \\ &= \begin{pmatrix} (1+\eta\kappa_\eta^{(1)})^2 & 0 & \cdots & 0 \\ 0 & (1+\eta\kappa_\eta^{(2)})^2 & 0 & 0 \\ 0 & \cdots & \cdots & 0 \\ 0 & \cdots & 0 & (1+\eta\kappa_\eta^{(n)})^2 \end{pmatrix}, \end{aligned}$$

where P is the change of basis matrix and $\kappa_\eta^{(i)}$ are the eigenvalues of the Weingarten map induced by h and the ith principal curvature of $\partial\Omega_\eta$. Then

$$\sqrt{\det\,\tilde{g}} = \sqrt{\det\,g}\left(\prod_{i=1}^{n}(1+\eta\kappa_\eta^{(i)})\right) = 1 + \sum_{i=1}^{n}\sigma_i(h)\eta^i, \tag{46}$$

where $\sigma_i(h)$ is the ith symmetric polynomial in the eigenvalues of the Weingarten map induced by h, and $\sigma_1(h) = 2H_\eta$ is the non averaged mean curvature ($H_\eta = \frac{1}{n}\sum_{i=1}^{n}\kappa_\eta^{(i)}$ with $\kappa_\eta^{(i)}$ its ith principal curvature) and $\sigma_n(h) = G_\eta = \prod_{i=1}^{n}\kappa_\eta^{(i)}$ is the Gaussian curvature. When $n = 1$ we recover the Jacobian obtained for curves in two dimensions in (44) and when $n = 2$ we obtain the Jacobian obtained in (45).

Remark: These changes of variable only hold if η is the constant distance between the two level sets $\partial\Omega$ and $\partial\Omega_\eta$, and η is sufficiently small, i.e. less than the focal distance. We also note that the first order curvature correction (in η) obtained in the above changes of variable is the first variation of arc length (for curves) and area (for surfaces) from Riemannian geometry.

B Single and double layer potentials for Laplace's equation

Let Ω be a bounded set of $\mathbb{R}^n$, $n \in \mathbb{N}^*$, and consider Laplace's equation

$$\Delta u(x) = 0, \tag{47}$$

in the bounded set Ω subject to Dirichlet, Neumann or the general Robin as in (8) boundary conditions on $\partial\Omega$. We refer to (47) as the *interior* Laplace problem. We define the fundamental solution of Laplace's equation on $\mathbb{R}^n$ to be the solution $\Phi(x, y)$ of

$$\Delta_y\Phi(x, y) = \delta(x - y), \tag{48}$$

for $x, y \in \mathbb{R}^n$, where $n \in \mathbb{N}^*$ is the dimension. By noticing that Laplace's equation is invariant under rotations, (47) can be solved by searching for radial solutions. The fundamental solution

of Laplace's equation can therefore be expressed as

$$\Phi(x,y) = \begin{cases} \frac{1}{2\pi}\ln|x-y| & \text{for } n=2, \\ -\dfrac{1}{n(n-2)\rho_n|x-y|^{n-2}} & \text{for } n \geq 3, \end{cases} \tag{49}$$

where ρ_n is the volume of the unit ball in $\mathbb{R}^n$. A few properties of the fundamental solution Φ are summarized in the following theorem:

Theorem B.1 (Properties of Φ) *For all $x, y \in \mathbb{R}^n$, $x \neq y$, we have*

1. *Symmetry:* $\Phi(x,y) = \Phi(y,x)$,
2. $\nabla_y \Phi(x,y) = -\nabla_x \Phi(x,y)$,
3. $\nabla_y \Phi(x,y) = -\nabla_y \Phi(y,x)$.

Since the fundamental solution Φ satisfies (48) we can express the solution u of (47) as an integral involving Φ. Using Green's identity for u and Φ defined in Equations (47) and (48) respectively, it follows that

$$\int_\Omega (u(y)\Delta_y\Phi(x,y) - \Phi(x,y)\Delta_y u(y))\,dy = \int_{\partial\Omega} \left(u(y(s))\frac{\partial\Phi(x,y(s))}{\partial n_y} - \Phi(x,y(s))\frac{\partial u(y(s))}{\partial n_y} \right) ds, \tag{50}$$

where n_y is the *outward* unit normal to Ω at the point $y(s) \in \partial\Omega$. Since u is harmonic on Ω Equation (50) simplifies to

$$\int_\Omega u(y)\Delta_y\Phi(x,y)dy = \int_{\partial\Omega} \left(u(y(s))\frac{\partial\Phi(x,y(s))}{\partial n_y} - \Phi(x,y(s))\frac{\partial u(y(s))}{\partial n_y} \right) ds.$$

We consider two cases:

- $x \in \Omega$
  In this case, since both x and y in the left-hand side of the above equation are in Ω, we obtain
  $$u(x) = \int_{\partial\Omega} \left(u(y(s))\frac{\partial\Phi(x,y(s))}{\partial n_y} - \Phi(x,y(s))\frac{\partial u(y(s))}{\partial n_y} \right) ds.$$
- $x \in \bar{\Omega}^c$
  In this case, we have $\Delta_y\Phi(x,y) = 0$ since $x \in \bar{\Omega}^c$ and $y \in \Omega$, and thus Equation (50) further simplifies to
  $$0 = \int_{\partial\Omega} \left(u(y(s))\frac{\partial\Phi(x,y(s))}{\partial n_y} - \Phi(x,y(s))\frac{\partial u(y(s))}{\partial n_y} \right) ds.$$

We therefore obtain the following identity:

$$\int_{\partial\Omega} \left(u(y(s))\frac{\partial\Phi(x,y(s))}{\partial n_y} - \Phi(x,y(s))\frac{\partial u(y(s))}{\partial n_y} \right) ds = \begin{cases} u(x) & \text{if } x \in \Omega, \\ 0 & \text{if } x \in \bar{\Omega}^c. \end{cases} \tag{51}$$

We now define the *exterior* Laplace problem as the equation

$$\Delta v(x) = 0, \tag{52}$$

satisfied in $\bar{\Omega}^c$. The boundary condition for v on $\partial\Omega$ will be described later. Using Green's identity for v we can write

$$\int_{\bar{\Omega}^c} (v(y)\Delta_y\Phi(x,y) - \Phi(x,y)\Delta_y v(y))\,dy = \int_{\partial\Omega} \left(v(y(s))\frac{\partial\Phi(x,y(s))}{\partial n_y^-} - \Phi(x,y(s))\frac{\partial v(y(s))}{\partial n_y^-} \right) ds, \tag{53}$$

where n_y^- is the *outward* unit normal to $\bar{\Omega}^c$ at the point $y(s) \in \partial\Omega$. Noticing that $n_y^- = -n_y$, we rewrite (53) as

$$\int_{\bar{\Omega}^c} (v(y)\Delta_y\Phi(x,y) - \Phi(x,y)\Delta_y v(y))\, dy = \int_{\partial\Omega} \left(\Phi(x,y(s)) \frac{\partial v(y(s))}{\partial n_y} - v(y(s)) \frac{\partial \Phi(x,y(s))}{\partial n_y} \right) ds.$$

Simplifying the above left-hand side in the same manner as we simplified the left-hand side of (50), we obtain the identity

$$\int_{\partial\Omega} \left(\Phi(x,y(s)) \frac{\partial v(y(s))}{\partial n_y} - v(y(s)) \frac{\partial \Phi(x,y(s))}{\partial n_y} \right) ds = \begin{cases} 0 & \text{if } x \in \Omega, \\ v(x) & \text{if } x \in \bar{\Omega}^c. \end{cases} \tag{54}$$

Adding (51) and (54) we arrive at the following result:

$$\int_{\partial\Omega} \left(\left(\frac{\partial v(y(s))}{\partial n_y} - \frac{\partial u(y(s))}{\partial n_y} \right) \Phi(x,y(s)) + (u(y(s)) - v(y(s))) \frac{\partial \Phi(x,y(s))}{\partial n_y} \right) ds = \begin{cases} u(x) & \text{if } x \in \Omega, \\ v(x) & \text{if } x \in \bar{\Omega}^c. \end{cases} \tag{55}$$

We now define the following two boundary value problems:

Definition B.1 (Single layer boundary value problem) *Let Ω be a bounded set in $\mathbb{R}^n$ and define the single layer boundary value problem as*

$$\begin{cases} \Delta v_{sl}(x) = 0 \text{ in } \bar{\Omega}^c \\ v_{sl}(x) = u(x) \text{ on } \partial\Omega, \\ \lim_{|x|\to\infty} v_{sl}(x) = 0, \end{cases} \tag{56}$$

where u is the solution of (47) *with Dirichlet, Neumann, or the general Robin as in* (8) *boundary conditions.*

Similarly we define the double layer boundary value problem in the following way:

Definition B.2 (Double layer boundary value problem) *Let Ω be a bounded set in $\mathbb{R}^n$ and define the double layer boundary value problem as*

$$\begin{cases} \Delta v_{dl}(x) = 0 \text{ in } \bar{\Omega}^c \\ \frac{\partial v_{dl}(x)}{\partial n_x} = \frac{\partial u(x)}{\partial n_x} \text{ on } \partial\Omega, \\ \lim_{|x|\to\infty} v_{dl}(x) = 0, \end{cases} \tag{57}$$

where n_x is the outward unit normal to Ω at the point $x \in \partial\Omega$, and u is the solution of (47) *with Dirichlet, Neumann, the general Robin as in* (8) *boundary conditions.*

Thus if we choose the function v to be the solution v_{sl} of the single layer boundary value problem (56), (55) becomes

$$\int_{\partial\Omega} \alpha(y(s))\Phi(x,y(s))ds = \begin{cases} u(x) & \text{if } x \in \Omega, \\ v_{sl}(x) & \text{if } x \in \bar{\Omega}^c, \end{cases} \tag{58}$$

where $\alpha(y) = \left(\dfrac{\partial v_{sl}(y)}{\partial n_y} - \dfrac{\partial u(y)}{\partial n_y} \right)$ for $y \in \partial\Omega$. The function $\displaystyle\int_{\partial\Omega} \alpha(y(s))\Phi(x,y(s))ds$ in (58) is referred to as the single layer potential with density α.

If we choose v to be the solution v_{dl} of the double layer boundary value problem (57), (55) becomes

$$\int_{\partial\Omega} \beta(y(s)) \frac{\partial \Phi(x,y(s))}{\partial n_y} ds = \begin{cases} u(x) & \text{if } x \in \Omega, \\ v_{dl}(x) & \text{if } x \in \bar{\Omega}^c, \end{cases} \tag{59}$$

where $\beta(y) = u(y) - v_{dl}(y)$ for $y \in \partial\Omega$. The function $\displaystyle\int_{\partial\Omega} \beta(y(s)) \frac{\partial \Phi(x, y(s))}{\partial n_y} ds$ in (59) is known as the double layer potential with density β. It follows that in the bounded set Ω the solution u of (47) can be represented by either the single layer or the double layer potential. So far the above single layer and double layer potential functions are only defined on $\mathbb{R}^n \setminus \partial\Omega$. It is therefore interesting (and also useful for practical applications) to look at their respective limits as x approaches the boundary $\partial\Omega$. Standard results in potential theory [29] give the following theorems:

Theorem B.2 *Let $\partial\Omega$ be of class C^2 and $\alpha \in C(\partial\Omega)$. Then the single layer potential with density α is continuous throughout $\mathbb{R}^n$, namely*

$$\lim_{h \to 0^+} \int_{\partial\Omega} \alpha(y) \Phi(x \pm h n_x, y) ds(y) = \int_{\partial\Omega} \alpha(y(s)) \Phi(x, y(s)) ds,$$

where $x \in \partial\Omega$ and n_x is the outward unit normal to Ω at the point $x \in \partial\Omega$, and the integral exists as an improper integral.

Theorem B.3 *Let $\partial\Omega$ be of class C^2, and $\beta \in C(\partial\Omega)$. Then the double layer potential with density β can be continuously extended from Ω to $\bar{\Omega}$, and from $\bar{\Omega}^c$ to Ω^c with*

$$\lim_{h \to 0^+} \int_{\partial\Omega} \beta(y(s)) \frac{\partial \Phi(x \pm n_x, y(s))}{\partial n_y} ds = \int_{\partial\Omega} \beta(y(s)) \frac{\partial \Phi(x, y(s))}{\partial n_y} ds \mp \frac{1}{2} \beta(x),$$

where $x \in \partial\Omega$ and n_x is the outward unit normal to Ω, and the integral exists as an improper integral.

We also have the following result regarding the normal derivative of the fundamental solution of Laplace's equation:

Theorem B.4 *Let Φ be the fundamental solution of Laplace's equation defined in* (49)*. Then we have the following:*

$$\int_{\partial\Omega} \frac{\partial \Phi(x, y(s))}{\partial n_y} ds = \begin{cases} 1 & \text{if } x \in \Omega, \\ \frac{1}{2} & \text{if } x \in \partial\Omega, \\ 0 & \text{if } x \in \bar{\Omega}^c. \end{cases}$$

This result is used in Example 5.10. This result also shows that any non zero constant function defined on $\partial\Omega$ is an eigenvector for the operator $K : v \mapsto \displaystyle\int_{\partial\Omega} \frac{\partial \Phi(x, y(s))}{\partial n_y} v(y(s)) ds$ associated to the eigenvalue $\dfrac{1}{2}$.

B.1 Integral equations for Poisson's equations

For Ω a bounded set of $\mathbb{R}^n$ and a real function ψ_0 defined on Ω, we consider Poisson's equation

$$\Delta u(x) = \psi_0(x), \tag{60}$$

for $x \in \Omega$, subject to either Dirichlet, Neumann, or the general Robin as in (8) boundary conditions. The solution to this problem can be obtained using the fundamental solution of Laplace's equation through integral equations. By following the procedure described in Section B.1 for Laplace's equation, we obtain the following single and double layer potential formulations for the Poisson problem respectively:

$$\int_{\partial\Omega} \alpha(y(s)) \Phi(x, y(s)) ds + \int_{\Omega} \Phi(x, y) \psi_0(y) dy = \begin{cases} u(x) & \text{if } x \in \Omega, \\ v_{sl}(x) & \text{if } x \in \bar{\Omega}^c, \end{cases} \tag{61}$$

where $\alpha(y) = \left(\dfrac{\partial v_{sl}(y)}{\partial n_y} - \dfrac{\partial u(y)}{\partial n_y} \right)$ for $y \in \partial\Omega$ and v_{sl} is the solution of the single layer boundary value problem (56), and

$$\int_{\partial\Omega} \beta(y(s)) \frac{\partial \Phi(x, y(s))}{\partial n_y} ds + \int_{\Omega} \Phi(x, y) \psi_0(y) dy = \begin{cases} u(x) & \text{if } x \in \Omega, \\ v_{dl}(x) & \text{if } x \in \bar{\Omega}^c, \end{cases} \tag{62}$$

where $\beta(y) = u(y) - v_{dl}(y)$ for $y \in \partial\Omega$ and v_{dl} is the solution of the double layer boundary value problem (57). Similarly to Laplace's equation, the limit of the single layer potential (61) and the double layer potential (62) as x approaches the boundary $\partial\Omega$ are obtained using standard results in potential theory.

We now consider the Dirichlet problem for Poisson's equation (60),

$$\begin{cases} \Delta u(x) = \psi_0(x) & \text{on } \Omega \\ u(x) = f(x) & \text{on } \partial\Omega, \end{cases} \tag{63}$$

and the Neumann problem for Laplace's equation

$$\begin{cases} \Delta u(x) = \psi_0(x) & \text{on } \Omega \\ \frac{\partial u(x)}{\partial n_x} = g(x) & \text{on } \partial\Omega \text{ such that } \int_{\partial\Omega} g(x(s)) ds = \int_{\Omega} \psi_0(x) dx. \\ \lim_{|x| \to \infty} u(x) = 0. \end{cases} \tag{64}$$

Contrary to the Dirichlet problem which is well-posed, the Neumann problem stated in (64) is ill-posed since a solution to (64) might not always exist for any function g defined on the boundary $\partial\Omega$. It is therefore useful to observe that since u satisfies Poisson's equation on Ω, it follows from the divergence theorem that

$$\int_{\Omega} \Delta u(x) dx = \int_{\partial\Omega} \frac{\partial u(x(s))}{\partial n_x} ds = \int_{\Omega} \psi_0(x) dx.$$

Thus in order for the Neumann problem (64) to have a solution it is necessary to impose some constraints on the function g prescribed on the boundary. In particular g should satisfy the compatibility condition

$$\int_{\partial\Omega} g(x(s)) ds = \int_{\Omega} \psi_0(x) dx. \tag{65}$$

Note also the solution of the Neumann problem can only be obtained up to a constant.

Let us now first consider the Dirichlet problem (63). To obtain its solution u in Ω, we can use two different approaches: one using the single layer potential formulation (58) or one using the double layer potential formulation (59).

Single layer potential formulation

In this case we represent u in Ω as

$$u(x) = \int_{\partial\Omega} \alpha(y(s)) \Phi(x, y(s)) ds + \int_{\Omega} \Phi(x, y) \psi_0(y) dy,$$

where α is the unknown density on $\partial\Omega$. We remark that u is entirely determined by the knowledge of the single layer density α defined on the boundary $\partial\Omega$. Letting x go to the boundary $\partial\Omega$, we use the result of Theorem B.2 to obtain

$$f(x) = \int_{\partial\Omega} \alpha(y(s)) \Phi(x, y(s)) ds + \int_{\Omega} \Phi(x, y) \psi_0(y) dy, \qquad \text{for } x \in \partial\Omega, \tag{66}$$

which is an integral equation involving only boundary quantities α and f. This equation is a Fredholm equation of the first kind (see [29]). Since f and Φ are both known quantities it is possible to use (66) to solve for α. The steps to solve the Dirichlet problem using the single layer potential formulation are thus as follows:

1. Find the density α defined on $\partial\Omega$ such that

$$\int_{\partial\Omega} \alpha(y(s))\Phi(x, y(s))ds = f(x) - \int_{\Omega} \Phi(x, y)\psi_0(y)dy, \qquad \text{for } x \in \partial\Omega.$$

2. Reconstruct u in Ω using the single layer potential formulation

$$u(x) = \int_{\partial\Omega} \alpha(y(s))\Phi(x, y(s))ds + \int_{\Omega} \Phi(x, y)\psi_0(y)dy, \qquad \text{for } x \in \Omega.$$

Double layer potential formulation

In this case we represent u in Ω as

$$u(x) = \int_{\partial\Omega} \beta(y(s))\frac{\partial\Phi(x, y(s))}{\partial n_y}ds + \int_{\Omega} \Phi(x, y)\psi_0(y)dy,$$

where β is the unknown density on $\partial\Omega$. Again we remark that u is entirely determined by the knowledge of the double layer density β defined on the boundary $\partial\Omega$. Letting x go to the boundary $\partial\Omega$ from the inside of Ω, we use the result of Theorem B.3 to obtain

$$f(x) = \int_{\partial\Omega} \beta(y(s))\frac{\partial\Phi(x, y(s))}{\partial n_y}ds + \int_{\Omega} \Phi(x, y)\psi_0(y)dy + \frac{1}{2}\beta(x), \qquad \text{for } x \in \partial\Omega, \qquad (67)$$

which is a Fredholm equation of the second kind. Note that if we let x go the boundary from the outside of Ω we obtain another Fredholm equation of the second kind, namely

$$\lim_{h\to 0^+} v_{dl}(x + hn_x) = \int_{\partial\Omega} \beta(y(s))\frac{\partial\Phi(x, y(s))}{\partial n_y}ds + \int_{\Omega} \Phi(x, y)\psi_0(y)dy - \frac{1}{2}\beta(x), \qquad \text{for } x \in \partial\Omega.$$

Unfortunately we do not know the function v_{dl} nor its limit as x approaches the boundary $\partial\Omega$. This equation is therefore not useful in practice. Nevertheless, (67) can be used to solve for β. The steps to solve the Dirichlet problem using the double layer potential formulation are as follows:

1. Find the density β defined on $\partial\Omega$ such that

$$\int_{\partial\Omega} \beta(y(s))\frac{\partial\Phi(x, y(s))}{\partial n_y}ds + \frac{1}{2}\beta(x) = f(x) - \int_{\Omega} \Phi(x, y)\psi_0(y)dy, \qquad \text{for } x \in \partial\Omega.$$

2. Reconstruct u in Ω using the double layer potential formulation

$$u(x) = \int_{\partial\Omega} \beta(y(s))\frac{\partial\Phi(x, y(s))}{\partial n_y}ds + \int_{\Omega} \Phi(x, y)\psi_0(y)dy, \qquad \text{for } x \in \Omega.$$

Between these two approaches the double layer potential formulation is often preferred since the Fredholm equation of the second kind leads to a numerical system with a better condition number than the system obtained from the Fredholm equation of the first kind.

Single layer formulation for the Neumann problem

We consider the Neumann problem

$$\begin{cases} \Delta u(x) = \psi_0(x) & \text{on } \Omega \\ \frac{\partial u(x)}{\partial n_x} = g(x) & \text{on } \partial\Omega \text{ such that } \int_{\partial\Omega} g(x(s))ds = \int_{\Omega} \psi_0(x)dx. \\ \lim_{|x|\to\infty} u(x) = 0. & \end{cases}$$

In this case it is necessary to use the single layer potential formulation (58) and represent u in Ω as

$$u(x) = \int_{\partial\Omega} \alpha(y(s))\Phi(x, y(s))ds + \int_{\Omega} \Phi(x, y)\psi_0(y)dy, \tag{68}$$

where α is the unknown density on $\partial\Omega$. This formulation however cannot be used directly since the boundary condition of the Neumann problem is prescribed on the normal derivative of u on $\partial\Omega$ instead of being specified on the function u. It is therefore necessary to compute the normal derivative of u in (68) and then take its limit as x approaches the boundary $\partial\Omega$. Here again standard results in potential theory apply and give the following theorem:

Theorem B.5 *Let $\partial\Omega$ be of class C^2. Then for the single layer potential u with continuous density α we have*

$$\lim_{h\to 0^+} \frac{\partial u(x \pm hn_x)}{\partial n_x} = \int_{\partial\Omega} \alpha(y(s)) \frac{\partial\Phi(x, y(s))}{\partial n_x} ds \pm \frac{1}{2}\alpha(x),$$

where $\lim_{h\to 0^+} \frac{\partial u(x\pm hn_x)}{\partial n_x}$ is to be understood in the sense of uniform convergence on $\partial\Omega$ and where the integral exists as an improper integral.

Using this result we can solve the Neumann problem as follows:

1. Find the density α defined on the domain boundary $\partial\Omega$ such that

$$\int_{\partial\Omega} \alpha(y(s)) \frac{\partial\Phi(x, y(s))}{\partial n_x} ds - \frac{1}{2}\alpha(x) = g(x) - \int_{\Omega} \frac{\partial\Phi(x, y)}{\partial n_x} \psi_0(y)dy, \qquad \text{for } x \in \partial\Omega.$$

2. Reconstruct u in Ω using the single layer potential formulation

$$u(x) = \int_{\partial\Omega} \alpha(y(s))\Phi(x, y(s))ds + \int_{\Omega} \Phi(x, y)\psi_0(y)dy, \qquad \text{for } x \in \Omega.$$

C Accuracy of the regularizations of $\frac{\partial\Phi}{\partial n}$

In this section we estimate the errors incurred by our regularizations of the normal derivative of the fundamental solution in two and three dimensions.

C.1 Two dimensions

Let $\mathcal{C}$ be a C^2 curve in $\mathbb{R}^2$ and let z be a point on $\mathcal{C}$. We assume that we have a parameterization $(x(t), y(t))$ of $\mathcal{C}$ and consider the Frenet frame associated to $\mathcal{C}$ and centered at $z = (x(t_0), y(t_0) \in \mathcal{C}$ for some $t_0 > 0$. In that frame z is the point $(0, 0)$, the x-axis is the tangent and the y-axis the normal. For simplicity we denote by O the origin of the frame (which is also z). See Figure 12. Locally around the origin, the equation of the curve can be written as a function $y = f(x)$. As a result we have $f(0) = 0$, $f'(0) = 0$ and $f''(0) = \kappa(0) = \frac{1}{R}$ is the curvature of the curve at O. This curvature is also the curvature of $\mathcal{C}$ at z.

Now we consider the osculating circle of the curve $\mathcal{C}$ at O. In the Frenet frame the osculating circle is centered at $(0, R)$. The equation of the circle (bottom portion) can be written as

$$y = R - \sqrt{R^2 - x^2},$$

for $|x| < R$. For $|h| < R$ a small parameter we consider a point M on the osculating circle with coordinates $(h, R - \sqrt{R^2 - h^2})$ and a point P on the curve with coordinates $(h, f(h))$. We

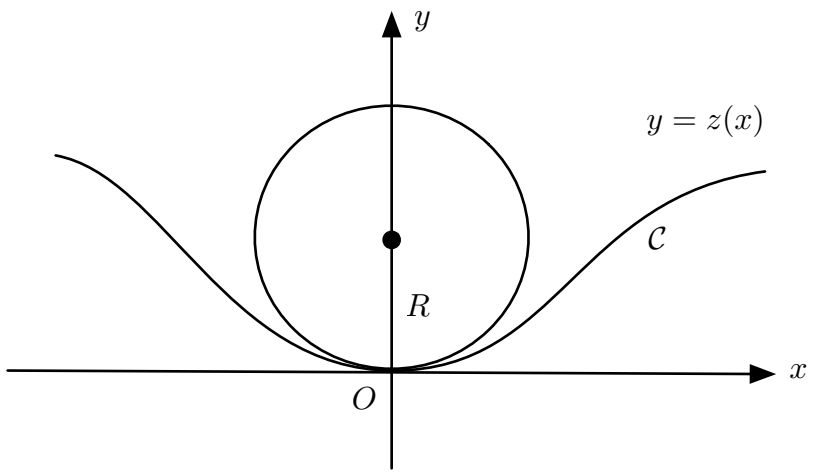


Figure 12: The curve and its osculating circle in the Frenet frame.

compute the difference of their y-coordinates:

$$\begin{aligned} f(h) - (R - \sqrt{R^2 - h^2}) &= f(0) + hf'(0) + \frac{h^2}{2} f''(0) + \frac{h^3}{6} f'''(0) + \mathcal{O}(h^4) \\ &\quad -R + R\left(1 - \frac{h^2}{2R^2} + \frac{h^4}{8R^4} + \mathcal{O}(h^6)\right) \\ &= \frac{h^2}{2R} + \frac{h^3}{6} f'''(0) - \frac{h^2}{2R} + \mathcal{O}(h^4) \\ &= \mathcal{O}(h^3). \end{aligned}$$

If the point O is a vertex, then $f'''(0) = 0$ and the circle is called overosculating. In this case the contact point between the curve and its osculating circle is of order ≥ 4. We recall that a vertex of a curve in $\mathbb{R}^2$ is a point where the contact order of the curve with its osculating circle is at least 4 (i.e. $\mathcal{O}(h^4)$).

Estimate for the normal derivative Let M be a point on the osculating circle such that its coordinates are $(x, R - \sqrt{R^2 - x^2})$ for $|x| \ll R$. Let P be a point on the curve such that its coordinates are $(x, f(x))$. We compare the two quantities $\dfrac{\partial \Phi(M,O)}{\partial n_O}$ and $\dfrac{\partial \Phi(P,O)}{\partial n_O}$.

$$\begin{aligned} \frac{\partial \Phi(M,O)}{\partial n_O} &= -\frac{1}{2\pi} \frac{(M-O)}{|M-O|^2} \cdot n_O = -\frac{1}{2\pi} \frac{\sqrt{R^2 - x^2} - R}{x^2 + (R - \sqrt{R^2 - x^2})^2} = \frac{1}{4\pi R}, \\ \frac{\partial \Phi(P,O)}{\partial n_O} &= -\frac{1}{2\pi} \frac{(P-O)}{|P-O|^2} \cdot n_O = \frac{1}{2\pi} \frac{f(x)}{x^2 + f(x)^2} = \frac{1}{2\pi} \frac{R - \sqrt{R^2 - x^2} + \mathcal{O}(x^3)}{x^2 + (R - \sqrt{R^2 - x^2} + \mathcal{O}(x^3))^2} \\ &= \frac{1}{4\pi R} + \mathcal{O}(x). \end{aligned}$$

We note that if the curve is locally convex we have

$$\frac{\partial \Phi(P,O)}{\partial n_O} = \frac{1}{4\pi R} + \mathcal{O}(x^2).$$

Consequently we have in general

$$\frac{\partial \Phi(P,O)}{\partial n_O} = \frac{\partial \Phi(M,O)}{\partial n_O} + \mathcal{O}(x^p) = \frac{\partial \Phi(M,O)}{\partial n_O} + \mathcal{O}(|O-P|^p),$$

since $|O-P| = \mathcal{O}(x)$ and where $p = 1$ in general and $p = 2$ if the origin O (or $z \in \mathcal{C}$) is a vertex.

C.2 Three dimensions

Let $\mathcal{S}$ be a C^2 surface in $\mathbb{R}^3$, and let P be a point on $\mathcal{S}$. For simplicity in the calculations we consider a local coordinate system centered at P with axes x, y and z such that the tangent plane to surface $\mathcal{S}$ at P is the xy plane with the principal directions being the x-axis and the y-axis. Note that P is also the origin (denoted by O). Locally around the origin, the equation of the surface can be written as a function $z = f(x,y)$. As a result, we have $f(0,0) = 0$, $f_x(0,0) = 0$, $f_y(0,0) = 0$, $f_{xx}(0,0) = \kappa_1$, $f_{yy}(0,0) = \kappa_2$ and $f_{xy}(0,0) = f_{yx}(0,0) = 0$, where κ_1 and κ_2 are the two principal curvatures of $\mathcal{S}$ at P.

If we use the tangent plane at P, (equivalently the origin O), then a point on the tangent plane can be written as $(x,y,0)$ so the normal derivative of the fundamental solution becomes (for x and y small):

$$\frac{\partial \Phi(P,O)}{\partial n_O} \quad = \quad -\frac{1}{4\pi}\frac{(P-O)\cdot(0,0,-1)}{|P-O|^3} = -\frac{1}{4\pi}\frac{(x,y,0)\cdot(0,0,-1)}{(x^2+y^2)^{\frac{3}{2}}} = 0.$$

This is equivalent to throwing out points on the interface that are too close to P. As pointed out in Section 4.1, the accuracy resulting from this regularization can be further improved by using the osculating paraboloid instead of the tangent plane.

We consider the osculating paraboloid of the surface $\mathcal{S}$ at O. We can write the equation of the paraboloid as

$$z(x,y) = \frac{1}{2}\left(\kappa_1 x^2 + \kappa_2 y^2\right).$$

Near the the origin, we consider a point M on the osculating paraboloid with coordinates $(x, y, \frac{1}{2}\left(\kappa_1 x^2 + \kappa_2 y^2\right))$ and a point P on the surface with coordinate $(x, y, f(x,y))$ and compute the difference in their z-coordinates:

$$\begin{aligned}
& f(x,y) - \frac{1}{2}\left(\kappa_1 x^2 + \kappa_2 y^2\right)) \\
= \quad & f(0,0) + f_x(0,0)x + f_y(0,0)y + \frac{1}{2}\left(f_{xx}(0,0) + 2f_{xy}(0,0) + f_{yy}(0,0)\right) + \mathcal{O}(x^3, x^2y, xy^2, y^3) \\
& -\frac{1}{2}\left(\kappa_1 x^2 + \kappa_2 y^2\right)) \\
= \quad & \mathcal{O}(x^3, x^2y, xy^2, y^3).
\end{aligned}$$

The osculating paraboloid has contact of order 3 in general. If the point is a vertex, then the contact order is at least 4.

Estimate for the normal derivative Let M be a point on the osculating paraboloid with coordinates $(x, y, \frac{1}{2}\left(\kappa_1 x^2 + \kappa_2 y^2\right)))$ and let P be a point on $\mathcal{S}$ with coordinates $(x, y, f(x,y))$. We compare the two quantities $\frac{\partial \Phi(M,O)}{\partial n_O}$ and $\frac{\partial \Phi(P,O)}{\partial n_O}$. We compute these quantities using cylindrical coordinates. The point M on the paraboloid can be described as

$$p(r,\theta) = \left(r\cos(\theta), r\sin(\theta), \frac{1}{2}\left(\kappa_1\left(r\cos(\theta)\right)^2 + \kappa_2\left(r\sin(\theta)\right)^2\right)\right),$$

for $r \in [0,\tau]$ and $\theta \in [0,2\pi]$. Then

$$\frac{\partial p}{\partial r} \times \frac{\partial p}{\partial \theta} = \left(-r^2\kappa_1\cos(\theta), -r^2\kappa_2\sin(\theta), r\right).$$

Thus $||\frac{\partial p}{\partial r} \times \frac{\partial p}{\partial \theta}|| = r\sqrt{1+r^2\left(\left(\kappa_1\cos(\theta)\right)^2 + \left(\kappa_2\sin(\theta)\right)^2\right)}$. Computing the normal derivative of the fundamental solution at the point P on the paraboloid we obtain

$$\begin{aligned}
\frac{\partial\Phi(P,O)}{\partial n_O} &= -\frac{1}{4\pi}\frac{(P-O)\cdot(0,0,-1)}{|P-O|^3} \\
&= -\frac{1}{4\pi}\frac{(x,y,\frac{1}{2}\left(\kappa_1x^2+\kappa_2y^2\right))\cdot(0,0,-1)}{\left(x^2+y^2+\frac{1}{4}\left(\kappa_1x^2+\kappa_2y^2\right)^2\right)^{\frac{3}{2}}} \\
&= \frac{1}{8\pi}\frac{\kappa_1x^2+\kappa_2y^2}{\left(x^2+y^2+\frac{1}{4}\left(\kappa_1x^2+\kappa_2y^2\right)^2\right)^{\frac{3}{2}}} \\
&= \frac{1}{8\pi}\frac{r^2\left(\kappa_1\cos^2(\theta)+\kappa_2\sin^2(\theta)\right)}{r^3\left(1+\frac{r^2}{4}\left(\kappa_1\cos^2(\theta)+\kappa_2\sin^2(\theta)\right)^2\right)^{\frac{3}{2}}} \\
&= \frac{1}{\pi r}\frac{\kappa_1\cos^2(\theta)+\kappa_2\sin^2(\theta)}{\left(4+r^2\left(\kappa_1\cos^2(\theta)+\kappa_2\sin^2(\theta)\right)^2\right)^{\frac{3}{2}}}.
\end{aligned}$$

We now compute the average value of the normal derivative of the fundamental solution over the piece of paraboloid defined as $\mathcal{P} := \left\{\frac{r^2}{2}\left(\kappa_1\cos^2\theta+\kappa_2\sin^2\theta\right) : \theta\in[0,2\pi], r\in[0,\tau]\right\}$. Using a Taylor expansion for small τ and Maple to simplify the result, the first few terms of the integral of the normal derivative of the fundamental solution over the piece of paraboloid $\mathcal{P}$ are

$$\begin{aligned}
&\frac{1}{\pi}\int_0^\tau\int_0^{2\pi}\frac{\kappa_1\cos^2(\theta)+\kappa_2\sin^2(\theta)}{r\left(4+r^2\left(\kappa_1\cos^2(\theta)+\kappa_2\sin^2(\theta)\right)^2\right)^{\frac{3}{2}}}r\sqrt{1+r^2\left(\left(\kappa_1\cos(\theta)\right)^2+\left(\kappa_2\sin(\theta)\right)^2\right)}drd\theta \\
= &\frac{1}{\pi}\int_0^\tau\int_0^{2\pi}\frac{\kappa_1\cos^2(\theta)+\kappa_2\sin^2(\theta)}{\left(4+r^2\left(\kappa_1\cos^2(\theta)+\kappa_2\sin^2(\theta)\right)^2\right)^{\frac{3}{2}}}\sqrt{1+r^2\left(\left(\kappa_1\cos(\theta)\right)^2+\left(\kappa_2\sin(\theta)\right)^2\right)}drd\theta \\
= &\frac{\kappa_1+\kappa_2}{8}\tau+\left(\frac{3}{512}\left(\kappa_1^3+\kappa_2^3\right)-\frac{1}{1536}\kappa_1\kappa_2\left(\kappa_1+\kappa_2\right)\right)\tau^3+\mathcal{O}(\tau^5).
\end{aligned}$$

Similarly, the first few terms of the surface area of the piece of paraboloid $\mathcal{P}$:

$$\begin{aligned}
&\int_0^\tau\int_0^{2\pi}r\sqrt{1+r^2\left(\left(\kappa_1\cos(\theta)\right)^2+\left(\kappa_2\sin(\theta)\right)^2\right)}drd\theta \\
= &\pi\tau^2+\pi\left(\frac{\kappa_1^2+\kappa_2^2}{8}\right)\tau^4-\pi\left(\frac{\kappa_1^4+\kappa_2^4}{64}+\frac{\kappa_1^2\kappa_2^2}{96}\right)\tau^6+\mathcal{O}(\tau^8).
\end{aligned}$$

Consequently, the first few terms in the average of the normal derivative over $\mathcal{P}$ are

$$\begin{aligned}
Avg &= \frac{\frac{1}{\pi}\int_0^\tau \int_0^{2\pi} \frac{\kappa_1\cos^2(\theta)+\kappa_2\sin^2(\theta)}{r\left(4+r^2(\kappa_1\cos^2(\theta)+\kappa_2\sin^2(\theta))^2\right)^{\frac{3}{2}}} r\sqrt{1+r^2\left((\kappa_1\cos(\theta))^2+(\kappa_2\sin(\theta))^2\right)}drd\theta}{\int_0^\tau\int_0^{2\pi} r\sqrt{1+r^2\left((\kappa_1\cos(\theta))^2+(\kappa_2\sin(\theta))^2\right)}drd\theta} \\
&= \frac{1}{8\pi\tau}(\kappa_1+\kappa_2) - \frac{1}{\pi}\left(\frac{5}{512}\left(\kappa_1^3+\kappa_2^3\right)+\frac{25}{1536}\kappa_1\kappa_2(\kappa_1+\kappa_2)\right)\tau \\
&\quad +\frac{1}{\pi}\left(\frac{3379}{983040}\left(\kappa_1^2\kappa_2^3+\kappa_1^3\kappa_2^2\right)+\frac{6487}{1966080}\left(\kappa_1^4\kappa_2+\kappa_1\kappa_2^4\right)+\frac{13}{131072}\left(\kappa_1^5+\kappa_2^5\right)\right)\tau^3+\mathcal{O}(\tau^5).
\end{aligned}$$

Now, we look at the error made by approximating the normal derivative of the fundamental solution using the paraboloid instead of the surface. Let P be the point $(x,y,z(x,y))$ on the paraboloid and S the point on the surface $(x,y,f(x,y))$. We have

$$\frac{\partial\Phi(P,O)}{\partial n_O} = \frac{1}{8\pi}\frac{\kappa_1x^2+\kappa_2y^2}{\left(x^2+y^2+\frac{1}{4}\left(\kappa_1x^2+\kappa_2y^2\right)^2\right)^{\frac{3}{2}}},$$

and thus

$$\begin{aligned}
\frac{\partial\Phi(S,O)}{\partial n_O} &= -\frac{1}{4\pi}\frac{(S-O)\cdot(0,0,-1)}{|S-O|^3} \\
&= \frac{1}{4\pi}\frac{f(x,y)}{(x^2+y^2+f(x,y)^2)^{\frac{3}{2}}} \\
&= \frac{1}{4\pi}\frac{z(x,y)+\mathcal{O}(x^3,x^2y,xy^2,y^3)}{\left(x^2+y^2+(z(x,y)+\mathcal{O}(x^3,x^2y,xy^2,y^3))^2\right)^{\frac{3}{2}}} \\
&= \frac{1}{4\pi}\frac{z(x,y)+\mathcal{O}(x^3,x^2y,xy^2,y^3)}{(x^2+y^2+z(x,y)^2)^{\frac{3}{2}}}\left(1+\frac{\mathcal{O}(x^5,x^4y,x^3y^2,x^2y^3,xy^4,y^5)}{x^2+y^2+z(x,y)^2}\right)^{-\frac{3}{2}}.
\end{aligned}$$

Using polar coordinates for the $\mathcal{O}$ term with $x=r\cos(\theta)$ and $y=r\sin(\theta)$, we have

$$\begin{aligned}
\frac{\partial\Phi(S,O)}{\partial n_O} &= \frac{1}{4\pi}\frac{z+\mathcal{O}(r^3)}{(x^2+y^2+z^2)^{\frac{3}{2}}}\left(1+\mathcal{O}(r^3)\right)^{-\frac{3}{2}} \\
&= \frac{1}{4\pi}\left(\frac{z}{(x^2+y^2+z^2)^{\frac{3}{2}}}+\mathcal{O}(1)\right)\left(1+\mathcal{O}(r^3)\right) \\
&= \frac{1}{4\pi}\frac{z}{(x^2+y^2+z^2)^{\frac{3}{2}}}+\mathcal{O}(1),
\end{aligned}$$

since $\frac{z}{(x^2+y^2+z^2)^{\frac{3}{2}}} = \mathcal{O}\left(\frac{1}{r}\right)$. Thus

$$\frac{\partial\Phi(S,O)}{\partial n_0} = \frac{\partial\Phi(P,O)}{\partial n_O}+\mathcal{O}(1).$$

Now

$$\begin{aligned}\int_{U(O;\tau)} \frac{\partial \Phi(S,O)}{\partial n_O} dS(y) &= \int_{\tilde{U}(0;\tau)} \frac{\partial \Phi(P,O)}{\partial n_O} dS(y) \\ &\quad + \int_0^{\tau} \int_0^{2\pi} \mathcal{O}(1)\, r \sqrt{1 + r^2 \left((\kappa_1 \cos(\theta))^2 + (\kappa_2 \sin(\theta))^2 \right)} dr d\theta \\ &= \int_{\tilde{U}(O;\tau)} \frac{\partial \Phi(P,O)}{\partial n_O} dS(y) + \mathcal{O}(\tau^2),\end{aligned}$$

where $\tilde{U}(O;\tau)$ is a neighborhood of O on the tangent plane to the surface at O (also P). If the point $P \in \partial\Omega$ is a vertex, namely if the paraboloid is overosculating, we would have at least a third order accuracy in τ. So in general we can write

$$\int_{U(O;\tau)} \frac{\partial \Phi(S,O)}{\partial n_O} dS(y) = \int_{\tilde{U}(O;\tau)} \frac{\partial \Phi(P,O)}{\partial n_O} dS(y) + \mathcal{O}(\tau^p),$$

where $p = 2$ in general and $p = 3$ if P (also the origin O on the tangent plane) is a vertex. Now we estimate the error made when we approximate the normal derivative of the fundamental solution weakly using the osculating paraboloid as the approximate surface. We have the following

$$\begin{aligned}\int_{U(x;\tau)} \frac{\partial \Phi(x,y)}{\partial n_y} \alpha(y) dS(y) &= \alpha(x) \int_{U(x;\tau)} \frac{\partial \Phi(x,y)}{\partial n_y} dS(y) + \nabla \alpha(x) \cdot \int_{U(x;\tau)} \frac{\partial \Phi(x,y)}{\partial n_y} (y - x) dS(y) + \cdots \\ &= \alpha(x) \left(\int_{\tilde{U}(x;\tau)} \frac{\partial \Phi(x,y)}{\partial n_y} dS(y) + \mathcal{O}(\tau^p) \right) \\ &\quad + \nabla \alpha(x) \cdot \int_{U(x;\tau)} \frac{\partial \Phi(x,y)}{\partial n_y} (y - x) dS(y) + \cdots . \\ &= \alpha(x) \left(\frac{1}{8\pi\tau} (\kappa_1 + \kappa_2) - \frac{1}{\pi} \left(\frac{5}{512} \left(\kappa_1^3 + \kappa_2^3 \right) + \frac{25}{1536} \kappa_1 \kappa_2 (\kappa_1 + \kappa_2) \right) \tau \right) + \mathcal{O}(\tau^p),\end{aligned}$$

where $p = 2$ in general and $p = 3$ if x is a vertex.

References

[1] D. Adalsteinsson and J. A. Sethian. A fast level set method for propagating interfaces. *J. of Comput. Phys.*, 118(2):269–277, 1995.

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

[3] K. E. Atkinson and G. Chandler. Boundary integral equation methods for solving laplace's equation with nonlinear boundary conditions: the smooth boundary case. *Mathematics of Computation*, 55(191):451–472, 1990.

[4] I. Babuška. The finite element method for elliptic equations with discontinuous coefficients. *Computing*, 5:207–213, 1970.

[5] J. Berossian, J. J. von Brecht, S. Zhu, E. Sifakis, and J. Teran. A second order virtual node method for elliptic problems with interfaces and irregular domains. *J. Comput. Phys.*, 229:6405–6426, 2010.

[6] S. Börn, L. Grasedyck, and W. Hackbusch. Hierarchical matrices. Technical report, Max-Planck Institut fur Mathematik in den Naturwissenschaften, Leipzig, 2003.

[7] M. Burger and S. Osher. A survey on level set methods for inverse problems and optimal design. *Euro J. of Appl. Math.*, 16(2):263–301, 2005.

[8] H. Chen, C. Min, and F. Gibou. A supra-convergence finite difference scheme for the Poisson and heat equations on irregular domains and non-graded adaptive cartesian grids. *J. Comput. Phys.*, 31(1/2):19–60, 2007.

[9] L.-T. Cheng and Y.-H. Tsai. Redistancing by flow time dependent Eikonal equation. *J. Comput. Phys.*, 227(2):4002–4017, 2008.

[10] I.-L. Chern and Y.-C. Shu. A coupling interface method for elliptic interface problems. *J. of Comput. Physics*, 225:2138–2174, 2007.

[11] P. G. Ciarlet. *The Finite Element Method for Elliptic Problems*. SIAM, 1978.

[12] I. Daubechies. *Ten lectures on wavelets*, volume 61 of *CBMS-NSF Regional Conference Series in Applied Mathematics*. SIAM, 1992.

[13] M. C. Delfour and J.-P. Zolesio. Shapes and geometries. analysis, differential calculus and optimization. *Advances in Design and Control, SIAM*, 2001.

[14] J. Dolbow and I. Harari. An efficient finite element method for embedded interface problems. *J. Numer. Methods Eng.*, 78:229–252, 2009.

[15] B. Engquist, A.-K. Tornberg, and R. Tsai. Discretization of dirac delta functions in level set methods. *J. Comput. Phys.*, 207(1):28–51, 2005.

[16] B. Engquist and L. Ying. Fast directional multilevel algorithms for oscillatory kernels. *SIAM J. Sci. Comput.*, 29(4):1710–1737, 2007.

[17] F. Ethridge and L. Greengard. A new fast-multipole accelerated poisson solver in two dimensions. *SIAM J. Sci. Comput.*, 23(3):741–760, 2001.

[18] H. Federer. Curvature measures. *Transactions of the American Mathematical Society*, 93:418–491, 1959.

[19] F. Gibou and R. Fedkiw. A fourth order accurate discretization for the laplace and heat equations on arbitrary domains, with applications to the stefan problem. *J. Comput. Phys.*, 202:577–601, 2005.

[20] F. Gibou, R. Fedkiw, L. Cheng, and M. Kang. A second order accurate symmetric discretization of the poisson equation on irregular domains. *J. Comput. Phys.*, 176:1–23, 2002.

[21] L. Greengard. *The rapid evaluation of potential fields in particle systems*. PhD thesis, MIT, 1988.

[22] L. Greengard and V. Rokhlin. A fast algorithm for particle simulations. *J. Comput. Phys.*, 73(2):325–348, 1987.

[23] A. Hansbo and P. Hansbo. An unfitted element method, based on nitsche's method for elliptic interface problems. *Comput. Methods Appl. Mech. Eng.*, 191:5537–5552, 2002.

[24] A. Hansbo and P. Hansbo. A finite element method for the simulation of strong and weak discontinuities in solid mechanics. *Comput. Methods Appl. Mech. Eng.*, 193:3523–3540, 2004.

[25] J. Helsing. Integral equation methods for elliptic problems with boundary conditions of mixed type. *J. Comput. Phys.*, 228(23), 2009.

[26] J. Huang and J. Zou. A mortar element method for elliptic problems with discontinous coefficients. *IMA J. Numer. Anal.*, 22:549–576, 2002.

[27] H. Johansen. *Cartesian grid embedded boundary finite difference methods for elliptic and parabolic differential equations on irregular domains*. PhD thesis, University of California, Berkeley, 1997.

[28] H. Johansen and P. Colella. A cartesian grid embedded boundary method for poisson's equation on irregular domains. *J. Comput. Phys.*, 147:60–85, 1998.

[29] R. Kress. *Linear Integral Equations*. Springer-Verlag, New York, second edition, 1999.

[30] R. Leveque and Z. Li. The immersed interface method for elliptic equations with discontinuous coefficients and singular sources. *SIAM J. Numer. Anal.*, 31:1019–1044, 1994.

[31] Z. Li and K. Ito. The immersed interface method: numerical solutions of pdes involving interfaces and irregular domains (frontiers in applied mathematics). *Society for Industrial and Applied Mathematics*, 2006.

[32] X. Liu, R. Fedkiw, and M. Kang. A boundary condition capturing method for poisson's equation on irregular domains. *J. Comput. Phys.*, 160(1):151–178, 2000.

[33] C. B. Macdonald and S. J. Ruuth. Level set equations on surfaces via the Closest Point Method. *J. Sci. Comput.*, 35(2–3):219–240, June 2008. doi:10.1007/s10915-008-9196-6.

[34] S. Mallat. *A wavelet tour of signal processing*. Academic Press Inc., San Diego, CA, 1998.

[35] C. Min and F. Gibou. A second order accurate level set method on non-graded adaptive cartesian grids. *J. Comput. Phys.*, 225(1):300–321, 2007.

[36] C. Min, F. Gibou, and H. D. Ceniceros. A supra-convergent finite difference scheme for the variable coefficient poisson equation on non-graded grids. *J. Comput. Phys.*, 218(1):123–140, 2007.

[37] W. W. Mullins and R. F. Sekerka. Morphological stability of a particle growing by diffusion or heat flow. *J. of Applied Physics*, 34:323–329, 1963.

[38] E. Nyström. Über die praktische Auflösung von Integralgleichungen mit Anwendungen auf Randwertaufgaben. *Acta Math.*, 54:185–204, 1930.

[39] S. Osher and J. A. Sethian. Fronts propagating with curvature dependent speed: Algorithms based on hamilton-jacobi formulations. *J. Comp. Phys.*, 79:12–49, 1988.

[40] D. Peng, B. Marriman, S. Osher, and H. Zhao. A PDE-based fast local level set method. *J. of Comput. Phys.*, 155(2):410–438, 1999.

[41] V. Rokhlin. Rapid solution of integral equations of classical potential theory. *J. Comput. Phys.*, 60:185–207, 1985.

[42] G. Russo and P. Smereka. A remark on computing distance functions. *J. Comput. Phys.*, 163:51–67, 2000.

[43] S. J. Ruuth and B. Merriman. A simple embedding method for solving partial differential equations on surfaces. *J. Comput. Phys.*, 227(3):1943–1961, 2008.

[44] J. Sethian. A fast marching level set method for monotonically advancing fronts. *Proceedings of the National Academy of Sciences*, 93(4):1591–1595, 1996.

[45] P. Smereka. The numerical approximation of a delta function with application to level set methods. *J. Comput. Phys.*, 211(1):77–90, 2006.

[46] J. Strain. Tree methods for moving interfaces. *J. Comput. Phys.*, 151:616–648, 1999.

[47] J. D. Towers. Two methods for discretizing a delta function supported on a level set. *J. Comput. Phys.*, 220(2):915–931, 2007.

[48] Y.-H. Tsai, L. Cheng, S. Osher, and H.-K. Zhao. Fast sweeping methods for a class of hamilton-jacobi equations. *SIAM Journal on Numerical Analysis*, 41(2):673–694, 2003.

[49] Y.-H. R. Tsai. Rapid and accurate computation of the distance function using grids. *J. Comput. Phys.*, 178(1):175–195, 2002.

[50] J. Tsitsiklis. Efficient algorithms for globally optimal trajectories. *IEEE Transactions on Automatic Control*, 40:1528–1538, 1995.

[51] S. Zahedi and A.-K. Tornberg. Delta function approximations in level set methods by distance function extension. *J. Comput. Phys.*, 229(6):2199–2219, 2010.

[52] J. Zhu, X. Chen, and T. Y. Hou. An efficient boundary integral method for the Mullins-Sekerka problem. *J. Comput. Phys.*, 127:246–267, 1996.