FMM-Accelerated Solvers for the Laplace–Beltrami Problem on Complex Surfaces in Three Dimensions

Dhwanit Agarwal1 · Michael O’Neil2 · Manas Rachh3

Received: 10 May 2023 / Revised: 16 August 2023 / Accepted: 23 August 2023 / Published online: 15 September 2023
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2023

Abstract
The Laplace–Beltrami problem on closed surfaces embedded in three dimensions arises in many areas of physics, including molecular dynamics (surface diffusion), electromagnetics (harmonic vector fields), and fluid dynamics (vesicle deformation). Using classical potential theory, the Laplace–Beltrami operator can be pre-/post-conditioned with an integral operator whose kernel is translation invariant, resulting in well-conditioned Fredholm integral equations of the second-kind. These equations have the standard 1/r kernel from potential theory, and therefore the equations can be solved rapidly and accurately using a combination of fast multipole methods (FMMs) and high-order quadrature corrections. In this work we detail such a scheme, presenting two alternative integral formulations of the Laplace–Beltrami problem, each of whose solution can be obtained via FMM acceleration. We then present several applications of the solvers, focusing on the computation of what are known as harmonic vector fields, relevant for many applications in electromagnetics. A battery of numerical results are presented for each application, detailing the performance of the solver in various geometries.

Keywords Nyström method · Laplace–Beltrami · Harmonic vector field · Fast multipole method · Potential theory

Michael O’Neil
oneil@cims.nyu.edu
Dhwanit Agarwal
dhwanit@oden.utexas.edu
Manas Rachh
mrachh@flatironinstitute.org

1 Oden Institute, University of Texas at Austin, Austin, TX 78712, USA
2 Courant Institute, NYU, New York, NY 10012, USA
3 Center for Computational Mathematics, Flatiron Institute, New York, NY 10010, USA
1 Introduction

The Laplace–Beltrami problem along surfaces in three dimensions appears in many fields, including plasma physics [33], fluid mechanics [37, 41], electromagnetics [11, 13, 14, 34], surface diffusion, pattern formation [24, 36], and computational geometry [28]. In general, the Laplace–Beltrami operator is a second-order variable coefficient elliptic operator whose behavior is highly dependent on the nature of the surface on which it is defined. Its eigenfunctions are useful in describing band-limited functions defined along the associated manifold or surface, and more generally, the related harmonic vector fields provide a basis for important classes of functions appearing in electromagnetics (e.g. non-radiating currents, etc.).

To describe the problem, suppose that \( \Gamma \) is at least a twice-differentiable surface without boundary embedded in \( \mathbb{R}^3 \). Let \( \nabla_{\Gamma} \) denote the intrinsic surface gradient, and let its adjoint acting on the space of continuously differentiable \( L^2 \) tangential vector fields on the boundary, denoted by \( \nabla_{\Gamma}^{\ast} \), be the intrinsic surface divergence. Given a mean-zero function \( f \) defined along \( \Gamma \), the Laplace–Beltrami problem is to find a mean-zero function \( u \) satisfying \( \Delta_{\Gamma} u = f \), where \( \Delta_{\Gamma} \) is the Laplace–Beltrami operator defined as \( \Delta_{\Gamma} = \nabla_{\Gamma} \cdot \nabla_{\Gamma} \). The mean-zero condition on \( f \) and \( u \) is required for unique solvability due to a rank-deficiency of the Laplace–Beltrami operator [11, 34, 35].

In many of the applications discussed above, the Laplace–Beltrami problem naturally arises for the computation of the Hodge decomposition of tangential vector fields. Any smooth tangential vector field \( F \) on the surface \( \Gamma \) can be written as

\[
F = \nabla_{\Gamma} \alpha + \mathbf{n} \times \nabla_{\Gamma} \beta + H,
\]

where \( \alpha, \beta \) are smooth functions on \( \Gamma \), \( H \) is a harmonic field satisfying \( \nabla_{\Gamma} \cdot H = \nabla_{\Gamma} \cdot \mathbf{n} \times H = 0 \), and \( \mathbf{n} \) is the unit normal to the surface \( \Gamma \). The curl free component \( \nabla_{\Gamma} \alpha \) and the divergence free component \( \mathbf{n} \times \nabla_{\Gamma} \beta \) can be computed via the solution of the following two Laplace–Beltrami problems:

\[
\Delta_{\Gamma} \alpha = \nabla_{\Gamma} \cdot F, \\
\Delta_{\Gamma} \beta = -\nabla_{\Gamma} \cdot (\mathbf{n} \times F).
\]

A related problem is that of determining a basis for harmonic vector fields on non-contractible surfaces, which plays a crucial role for several integral representations in computational electromagnetics. On a genus \( g \) object, the space of harmonic vector fields forms a \( 2g \)-dimensional vector space. One can compute elements of this basis via the solution of the two Laplace–Beltrami problems above for a randomly generated smooth vector field \( F \), which with high probability tends to have a non-trivial projection onto the subspace of harmonic vector fields.

Numerical methods for the Laplace–Beltrami problem can be typically classified into two categories. The first category of methods rely on the direct discretization of the differential operator on surface. Examples include finite element methods [2, 5, 7, 10], virtual element methods [3, 17], differencing methods [44], and level set methods [4, 8, 21, 31, 32]. The level set methods differ from the rest of the direct discretizations of the PDE as they rely on using an embedded finite difference grid in the volume \( \mathbb{R}^3 \) to discretize a thickening of the surface.

Most recently, a general purpose spectral elements method for surface PDEs, coupled with a fast direct solver, was introduced in [15]. Section 5.1.1 of that work applies the discretization scheme and fast direct solver to the Laplace–Beltrami problem on smooth surfaces, and Section 5.1.2 discusses solving the same problem on surfaces with corners and edges. The discretization scheme and fast solver of [15] results in an \( O(N^{3/2}) \) algorithm for
solving the Laplace–Beltrami problem, where $N$ is the total number of unknowns on the surface. When the discretization is relatively isotropic, the linear systems arising from spectral element methods are relatively well-conditioned and very high-accuracy can be achieved quite efficiently—likely more efficiently than the method presented in this work. However, if the surface has multiscale features which require a highly adaptive discretization, the conditioning of the spectral elements linear system will likely suffer and the overall obtainable accuracy will degrade. In this regime—large-scale, multiscale geometries—methods based on second-kind integral equation formulations are immune from ill-conditioning due to adaptive discretizations and would likely perform better when coupled with fast algorithms such as fast multipole methods.

On the other hand, the other general category of methods is to reformulate the Laplace–Beltrami problem as an integral equation along the surface. For example, on a flat surface the Laplace–Beltrami operator reduces to the standard Laplace operator and therefore the Green’s function is known to be $G(x) = -\log|x|/(2\pi)$. This Green’s function $G$ can be used as a parametrix for the fundamental solution to the Laplace–Beltrami operator on surfaces as well [29]. This idea has also been used for the solution of the related Yukawa–Beltrami problem on general surfaces, including surfaces with boundary in [30]. A separation of variables approach, combined with an integral reformulation of an ODE, is presented in [18]; this approach was also shown to be applicable in geometries with edges. An alternate integral equation-based approach is to use the Green’s function of the free-space Laplace equation in three dimensions as either left or right preconditioners for the Laplace–Beltrami problem [35].

In this work, we present a fast multipole method (FMM) accelerated iterative solver for the solution of the Laplace–Beltrami problems on complex three-dimensional surfaces based on the integral formulation presented in [35], and using the locally corrected quadrature approach of [19]. In some respects, this work can be considered Part II of the earlier work [35]. The primary motivation of choosing this approach is two-fold: (1) the resulting integral equations tend to be as well-conditioned as the underlying physical problem (which is quite well-conditioned in this case, in an absolute sense), and (2) since the representation uses layer potentials related to the free space Green’s function in three dimensions, the numerical solver can be coupled to existing optimized FMM libraries [19, 20]. We demonstrate the well-conditioned nature of the integral equation through a battery of numerical experiments, each of which incorporates a high-order representation of the surface and high-order quadratures. We also show the importance of what we refer to as numerical second-kindness when discretizing the integral equation using locally corrected quadrature methods. These methods rely on the accurate computation of the principal-value part of the layer potentials in the representation. In particular, while there are several equivalent integral formulations of the same PDE on the surface, we show that the formulation which explicitly handles the identity term of the integral equation tends to be the most stable (from a numerical point of view). Lastly, with the acceleration provided by FMMs, our integral formulation can be used to compute a basis for harmonic vector fields in relatively high-genus geometries.

The rest of the paper is organized in follows. In Sect. 2, we describe the Laplace–Beltrami problem. In Sect. 3, we review the integral equation formulation presented in [35], and in Sect. 4, we review a high order FMM-accelerated iterative solver. In Sect. 5, we discuss an application of the solver to the computation of a basis for harmonic vector fields on non-contractible surfaces. In Sect. 6, we present several numerical examples to demonstrate the efficiency of our approach, including a discussion on the importance of numerical second-kindness. Finally, in Sect. 7, we conclude by discussing avenues for future research.
2 The Laplace–Beltrami Problem

The Laplace–Beltrami operator, also known as the surface Laplacian, is the generalization of the Laplace operator to general (smooth) manifolds. In this work, we will focus on smooth surfaces embedded in three dimensions. Suppose \( \Gamma \subset \mathbb{R}^3 \) is an oriented surface with at least two continuous derivatives, and let \( x(u, v) : D \subset \mathbb{R}^2 \to \mathbb{R}^3 \) denote a local parametrization of \( \Gamma \). The first fundamental form, or equivalently the metric tensor at \( x(u, v) \), is given by

\[
g = \begin{bmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{bmatrix} = \begin{bmatrix} \partial_u x \cdot \partial_u x & \partial_u x \cdot \partial_v x \\ \partial_v x \cdot \partial_u x & \partial_v x \cdot \partial_v x \end{bmatrix},
\]

(2.1)

where \( \partial_u x \) and \( \partial_v x \) are abbreviations for \( \partial x / \partial u \) and \( \partial x / \partial v \), respectively, and are tangent vectors to the surface at \( x \in \Gamma \). The unit normal at \( x \) can be written as

\[
n(x) = \frac{\partial_u x \times \partial_v x}{|\partial_u x \times \partial_v x|}.
\]

(2.2)

The unit vectors in the directions of \( \partial_u x, \partial_v x, \) and \( n(x) \) form a local coordinate system (not necessarily orthogonal) at each point \( x \in \Gamma \).

Next, suppose that \( F \) is a smooth tangential vector field defined along \( \Gamma \) and is expressed in terms of the local tangent vectors \( \partial_u x, \partial_v x \) as

\[
F(u, v) = F_1 \partial_u x + F_2 \partial_v x.
\]

(2.3)

The surface divergence of \( F \), denoted by \( \nabla_\Gamma \cdot F \), is then given by

\[
\nabla_\Gamma \cdot F = \frac{1}{\sqrt{|g|}} \left( \partial_u \left( \sqrt{|g|} F_1 \right) + \partial_v \left( \sqrt{|g|} F_2 \right) \right),
\]

(2.4)

where \( |g| \) is the determinant of the metric tensor \( g \). The surface gradient operator \( \nabla_\Gamma \), formally defined as the adjoint of the surface divergence operator on the space of \( L^2 \) functions with respect to the induced metric on \( \Gamma \), is given by

\[
\nabla_\Gamma \psi = \left( g^{11} \partial_u \psi + g^{12} \partial_v \psi \right) \partial_u x + \left( g^{21} \partial_u \psi + g^{22} \partial_v \psi \right) \partial_v x,
\]

(2.5)

where \( \psi = \psi(u, v) \) is a scalar function defined on \( \Gamma \) and \( g^{ij} \) denote the components of the inverse of the metric tensor, denoted by \( g^{-1} \).

The Laplace–Beltrami operator is then defined as \( \Delta_\Gamma = \nabla_\Gamma \cdot \nabla_\Gamma \), and is explicitly given by the expression

\[
\Delta_\Gamma \psi = \frac{1}{\sqrt{|g|}} \left( \partial_u \left[ \sqrt{|g|} \left( g^{11} \partial_u \psi + g^{12} \partial_v \psi \right) \right] + \partial_v \left[ \sqrt{|g|} \left( g^{21} \partial_u \psi + g^{22} \partial_v \psi \right) \right] \right).
\]

(2.6)

The standard Laplace–Beltrami problem is to solve the following PDE for \( \psi \) along \( \Gamma \),

\[
\Delta_\Gamma \psi = f,
\]

(2.7)

where \( f \) is a given function. Conditions on the regularity required in \( f \), and the subsequent regularity obtained in \( \psi \) have been studied in \([11, 18, 23, 38]\). We will assume that all functions and surfaces are sufficiently smooth so as to obtain high order accuracy in the subsequently numerical examples.

As is clear, the Laplace–Beltrami operator maps constant functions along \( \Gamma \) to zero. Thus, the Eq. (2.7) is rank-one deficient \([22, 40]\) (and formally self-adjoint). The following lemma \([35, 38]\) summarizes a classical result regarding the well-posedness of the above Laplace–Beltrami problem.
Lemma 1 Let $\Gamma$ be a smooth, closed, and orientable boundary. Along $\Gamma$, the Laplace–Beltrami operator is uniquely invertible as a map from $M_0 \to M_0$, the space of mean-zero functions defined on $\Gamma$,
\[ M_0 = \left\{ \psi : \int_{\Gamma} \psi \, da = 0 \right\}. \tag{2.8} \]
Indeed, if the function $f$ defined along $\Gamma$ has mean zero, then there exists a unique mean-zero solution, $\psi$, to the Laplace–Beltrami problem $\Delta_\Gamma \psi = f$.

3 Integral Equation Formulations

In this section, we review an equivalent well-conditioned second-kind integral equation for the solution of the Laplace–Beltrami problem. This formulation relies on preconditioning the Laplace–Beltrami operator on both the left and right using the Laplace single layer potential. The resulting Fredholm operator can be expressed in terms of standard layer potentials for the free space Laplace equation in three dimensions. Before describing this formulation, we first define the relevant layer potentials used in the formulation. The Green’s function $G$ for Laplace’s equation in three dimensions is given by
\[ G(x, y) = \frac{1}{4\pi |x - y|}. \tag{3.1} \]
The single and double layer potentials are then given by:
\[ S[\sigma](x) = \int_{\Gamma} G(x, y) \sigma(y) \, da, \]
\[ D[\mu](x) = \int_{\Gamma} (\nabla_y G(x, y) \cdot n(y)) \mu(y) \, da. \tag{3.2} \]
When the target point $x$ lies on $\Gamma$, the integral defining the double layer should be interpreted in a principal value sense. For smooth surfaces $\Gamma$, both $S$ and $D$ are compact operators as maps from $L^2(\Gamma) \to L^2(\Gamma)$. Related to the single and double layer potentials are the restrictions of their normal derivatives to $\Gamma$ given by
\[ S'[\sigma](x) = n(x) \cdot \nabla \int_{\Gamma} G(x, y)\sigma(y) \, da, \]
\[ D'[\mu](x) = \int_{\Gamma} n(x) \cdot \nabla_x \nabla_y G(x, y) \cdot n(y) \mu(y) \, da. \tag{3.3} \]
The integral defining $S'$ is to be interpreted in a principal value sense, and that for $D'$ is to be interpreted in a finite-part sense [27, 34]. The operator $S'$ is also a compact operator on $L^2(\Gamma)$, whereas $D'$ is a hypersingular operator.

Finally, let $S''$ denote the restriction of the second normal derivative of $S$ to $\Gamma$ given by:
\[ S''[\sigma](x) = \int_{\Gamma} (n(x) \cdot \nabla_x \nabla_x G(x, y) \cdot n(y)) \sigma(y) \, da, \tag{3.4} \]
where, as for the integral defining $D'$, the integral should be interpreted in a finite-part sense [27, 34, 35]. It can be shown that the difference operator $S'' + D'$ is a compact operator on $L^2(\Gamma)$ [34], as the dominant singularities in the integrand cancel each other out.

We now review two equivalent second-kind boundary integral formulations for the solution of the Laplace–Beltrami equation (2.7) as presented in [35]. One formulation is to represent...
the solution $\psi = S[\sigma]$, where $\sigma$ is an unknown density and precondition to Eq. (2.7) using $S$, i.e. solve

$$S\Delta_{\Gamma} S[\sigma] = S[f]. \quad (3.5)$$

Using Calderón identities, the equation above can be rewritten as

$$-\frac{\sigma}{4} + D^2[\sigma] - S(S'' + D' + 2HS')[\sigma] = S[f], \quad (3.6)$$

where $H$ is the mean curvature along $\Gamma$. The above equation still has a one dimensional null-space which is related to constant functions comprising the null-space of the Laplace Beltrami problem. To impose the integral equation on the space of mean-zero functions, we instead solve the integral equation

$$-\frac{\sigma}{4} + D^2[\sigma] - S(S'' + D' + 2HS')[\sigma] + SWS[\sigma] = S[f], \quad (3.7)$$

where $W$ is the operator given by

$$W[\sigma] = \frac{1}{|\Gamma|} \int_{\Gamma} \sigma \, da. \quad (3.8)$$

If $f$ is a mean-zero function in $L^2(\Gamma)$, then there exists a unique solution $\sigma$ in $L^2(\Gamma)$ which satisfies Eq. (3.7), along with $\int_{\Gamma} S[\sigma] \, da = 0$. Furthermore, $\psi = S[\sigma]$ is the unique mean-zero solution to the Laplace–Beltrami problem $\Delta_{\Gamma} \psi = f$.

An alternate integral formulation is to instead represent $\psi = S^2[\sigma]$, where $\sigma$ as before is an unknown density and solve

$$\Delta_{\Gamma} S^2[\sigma] = f, \quad (3.9)$$

which using Calderón identities, can be rewritten as

$$-\frac{\sigma}{4} + (S')^2[\sigma] - (S'' + D' + 2HS')S[\sigma] = f. \quad (3.10)$$

The integral equation above has a similar one-dimensional null space issue as Eq. (3.6), which can be addressed using the averaging operator $W$ by instead solving

$$-\frac{\sigma}{4} + (S')^2[\sigma] - (S'' + D' + 2HS')S[\sigma] + WS^2[\sigma] = f. \quad (3.11)$$

Again, if $f$ is a mean-zero function in $L^2(\Gamma)$, then there exists a unique solution $\sigma$ in $L^2(\Gamma)$ which satisfies Eq. (3.11), along with $\int_{\Gamma} S^2[\sigma] \, da = 0$. Furthermore, $\psi = S^2[\sigma]$ is the unique mean-zero solution the Laplace–Beltrami problem $\Delta_{\Gamma} \psi = f$. A proof of these results can be found in [35].

4 FMM-Accelerated Solvers

In this section, we describe the details of a numerical solver for integral equation formulations of the Laplace–Beltrami problem. For this, we first briefly recap the FMM-accelerated locally corrected quadrature approach of [19]. For simplicity, consider a second-kind integral equation of the form

$$\sigma(x) + \int_{\Gamma} K(x, x') \sigma(x') \, da(x') = f(x) \quad x \in \Gamma, \quad (4.1)$$
where \( K(x, x') \) is either \( G(x, x') \) or one of its directional derivatives. Suppose that the surface \( \Gamma \) is represented via a disjoint union of patches \( \Gamma_j \) so that \( \Gamma = \bigcup_{j=1}^{N_{\text{patches}}} \Gamma_j \), and that each patch \( \Gamma_j \) is parametrized by a non-degenerate chart \( \mathbf{X}^j : T_0 \to \Gamma_j \), where \( T_0 \) is the standard simplex given by

\[
T_0 = \{(u, v) : u \geq 0, v \geq 0, u + v \leq 1 \} \subset \mathbb{R}^2.
\]

An order \( p \) discretization of \( \Gamma \) is one where the components of \( \mathbf{X}^j, \partial_u \mathbf{X}^j, \) and \( \partial_v \mathbf{X}^j \) are represented by a polynomial expansion of total degree less than \( p \) on \( T_0 \). One way to obtain such a representation is to sample the chart and the derivative information at order \( p \) Vioreanu-Rokhlin nodes \([43]\) which are stable interpolation nodes for orthogonal polynomials \([6, 26]\). For each discretization point \( \mathbf{x}_i \), we can split the computational domain as

\[
\mathbb{R}^3 = \text{Far}(\mathbf{x}_i) \cup \text{Near}(\mathbf{x}_i),
\]

with \( \text{Near}(\mathbf{x}_i) \) containing \( O(1) \) patches, and hence \( O(1) \) discretization points.

Let \( \{s_i, w_{ij}^{\text{over}}\}_{i=1}^{N_{\text{over}}} \) denote the oversampled quadrature nodes and weights for integrating smooth functions on \( \Gamma \), then the Nyström discretization in Eq. (4.2) takes the form

\[
\sigma_i + \sum_{j \neq i} w_{ij} K(\mathbf{x}_i, \mathbf{x}_j) \alpha_j = f(\mathbf{x}_i),
\]

where \( \sigma_j \) and \( \tilde{\alpha}_j \) correspond to (approximate) interpolated values of \( \sigma \) evaluated at \( s_j \) and \( y_{ij} \), respectively, obtained via implicit local polynomial interpolation of the samples \( \sigma_i \). The pairs \( \{y_{ij}, w_{ij}\} \) are the quadrature nodes and weights used in the adaptive integration procedure to accurately compute the contribution of the layer potential due to the self panel \( \Gamma_i \) and the associated near panels at the target location \( \mathbf{x}_i \). After adding and subtracting the
contribution of the oversampled sources in \( \text{Near}(x_i) \), and then subsequently composing the interpolation matrices to compute \( \tilde{\sigma}_{ij} \) and \( \tilde{\sigma}_j \) in \( \text{Near}(x_i) \) and the corresponding kernel evaluations, the discretized integral equation in (4.4) can be rewritten as

\[
\sigma_i + \sum_{j=1}^{N} \tilde{w}_{ij} \sigma_j + \sum_{j=1}^{N_{\text{over}}} w_{\text{over}}^{j} K(x_i, s_j) \tilde{\sigma}_j = f(x_i),
\]

where \( \tilde{w}_{ij} \) represents an adjusted weight-kernel product obtained from the addition/subtraction procedure. (See [19] for a very detailed discussion of this procedure, which we have very concisely summarized above.)

The rate limiting step for numerically evaluating the layer potential via this procedure is the computation of the local quadrature corrections. However, when using an iterative algorithm such as GMRES [39] to solve the integral equation, the effective quadrature-kernel weights \( \tilde{w}_{ij} \) above are precomputed and stored. Thus the computational cost of generating the quadrature corrections is amortized over the number of GMRES iterations. The memory requirement of this storage scales like \( O(N) \) as there are \( O(1) \) points in \( \text{Near}(x_i) \). Since the far field part of the layer potential is computed using a target independent quadrature rule, i.e. the sum over the oversampled sources \( s \) in (4.5), the corresponding contribution can be computed using standard fast multipole methods [19, 20]. Using this fast layer potential evaluator, we can obtain the solution \( \sigma \) in \( O(N) \) time if (4.4) is well-conditioned.

The recent release of the FMM3D package allows for the computation of the vector version of FMM sums, i.e., FMM sums with the same kernel, same source and target locations, but with different strength vectors. Let \( s_j \) denote the source locations, \( c_{\ell,j} \) denote the charge strengths, and \( v_{\ell,j} \) denote the dipole vectors, at \( N \) sources, \( j = 1, 2, \ldots, N \), and \( \ell = 1, 2 \ldots n_d \) densities, then the vector version of the Laplace FMM computes the potentials \( u_{\ell}(x) \) given by

\[
u_{\ell}(x) = \sum_{j=1}^{N} \frac{c_{\ell,j}}{|x - s_j|} - v_{\ell,j} \cdot \nabla_x \frac{1}{|x - s_j|}, \quad \ell = 1, 2 \ldots n_d,
\]

along with the their gradients and Hessians (if requested) at a given set of target locations \( x = t_i, i = 1, 2 \ldots M \). The vector version of the FMM is (empirically) computationally more efficient than separately calling \( n_d \) separate FMMs since various density independent quantities can be reused in the FMM algorithm. This feature of the FMM can be used for the more efficient computation of multiple operators in the integral equations for the Laplace–Beltrami problem.

We now turn our attention to discussing the specifics of GMRES accelerated iterative solvers for the Laplace Beltrami problem. While piecewise smooth discretizations of surfaces offer convenience to describe complicated three dimensional surfaces, they pose an additional challenge when solving surface PDEs such as the Laplace–Beltrami problem. In particular, the direct discretization of the Laplace–Beltrami integral equation in the form

\[
\mathcal{S}(\Delta_{\Gamma} + \mathcal{W})\mathcal{S}[\sigma] = \mathcal{S}[f]
\]

results in a numerical null space proportional to the number of patches in the triangulation and the order of discretization nodes on each triangle. The null space is purely a numerical artifact and can be attributed to the particular choice of discretization for representing the surface and the density. This issue can be remedied by using a basis and discretization that enforces smoothness of solutions across adjacent triangles, however, this introduces
additional complexity for discretizing the surface and representing the solution. Alternatively, the issue can instead be remedied by solving the mathematically equivalent integral equation

$$S(\Delta \Gamma + \mathcal{W})S[\sigma] = \nabla \cdot \nabla \Gamma S[\sigma] + \mathcal{W} S[\sigma] = S[f].$$  (4.8)

In the above form, the operator on the left is discretized via the composition of \(S\) followed by \(\nabla \Gamma\), followed by an application of \(S\) on the resulting vector density, and lastly followed by \((\nabla \cdot \cdot\cdot)\). The compact term \(\mathcal{W} S\) is the discretized analogously and added to the previous computation. In the rest of the paper, the integral operators are assumed to be discretized in the order they are written, from right-to-left. We will denote the expanded operator above as

$$A_1 = \nabla \cdot \nabla \Gamma S + \mathcal{W} S.$$  (4.9)

For comparison, we also present results related to the numerical discretizations of the alternative integral representations for solving this problem:

$$S(\Delta \Gamma + \mathcal{W})S[\sigma] = -\frac{\sigma}{4} + D^2[\sigma] - S \left( S^{''} + D' + 2 HS' \right) S[\sigma] = S[f].$$  (4.10)

and

$$(\Delta \Gamma + \mathcal{W})S^2[\sigma] = -\frac{\sigma}{4} + (S')^2[\sigma] - (S^{''} + D' + 2 HS') S[\sigma] + \mathcal{W} S^2[\sigma] = f.$$  (4.11)

The operators in these equations will be denoted as \(A_2\) and \(A_3\), where

$$A_2 = -\frac{1}{4} I + D^2 - S \left( S^{''} + D' + 2 HS' \right),$$

$$A_3 = -\frac{1}{4} I + (S')^2 - (S^{''} + D' + 2 HS') S + \mathcal{W} S^2.$$  (4.12)

In the following, suppose that \(\tau_i\), for \(i = 1, 2, \ldots, N\), are estimates of the solution to the linear system on the current iteration of GMRES. We will consider solving discretized versions of the integral equations in (4.8), (4.10), and (4.11). Across all three integral equations, there are seven different layer potential operators that appear: \(S, D, S', \partial_1 S, \partial_2 S, \partial_3 S, \text{ and } S'' + D'\), where \(\partial_j S\) denotes a partial derivative of the single layer potential along the coordinate direction \(x_j\).

Referring to (4.5), let \((\tilde{\omega}_{ij}^K)\) for \(K \in \{S, D, \partial_1 S, \partial_2 S, \partial_3 S, S', (S'' + D')\}\) denote the effective quadrature corrections for the corresponding discretized layer potentials. In order to use a vector version of the FMM, it is essential that the far part of the layer potentials be computed using the same set of source locations. This is ensured by choosing oversampled quadrature nodes and weights \(\{s_i, w_i^{\text{overflow}}\}_{i=1}^N\) that are accurate for the far field part of all seven kernels.

For some of the kernels above such as \(S', D', \text{ and } D'',\) the normals need to be evaluated at the oversampled source locations. In order to evaluate \(n(s_j)\) for \(s_j \in \Gamma_\ell\), this is done by using the interpolated values of \(\partial_u X^\ell\) and \(\partial_v X^\ell\) as follows,

$$n(s_j) = \frac{\partial_u X^\ell(u_j, v_j) \times \partial_v X^\ell(u_j, v_j)}{|\partial_u X^\ell(u_j, v_j) \times \partial_v X^\ell(u_j, v_j)|},$$  (4.13)

where \((u_j, v_j) \in T_0\) are the local coordinates of \(s_j\) on \(\Gamma_\ell\). The reason for using interpolated values of \(\partial_u X^\ell\) and \(\partial_v X^\ell\) instead of directly interpolating the normals is the smoother nature of the tangential derivatives as compared to the surface normal owing to the normalization factor \(1/|\partial_u X \times \partial_v X|\) in the normal vector. In the \texttt{fmm3dbie} package, only first derivative
information of the charts are stored for representing the surface. The mean curvature $H(x_i)$ is obtained by computing second derivative of the charts $X^\ell$ using spectral differentiation of $\partial_u X^\ell$ and $\partial_v X^\ell$ for $x_i \in \Gamma_\ell$. Finally, for any function $f$ sampled on $\Gamma$, $f_j, j = 1, 2, \ldots N_{\text{over}}$, denotes the samples of the function at the oversampled discretization nodes.

### 4.1 FMM-Accelerated Application of $\mathcal{A}_1$

In this section, we discuss the FMM accelerated evaluation of

$$\mathcal{A}_1[\tau] = \nabla \cdot S \nabla_\tau S[\tau] + SY_S S[\tau].$$

(4.14)

Let $\phi^{(1)}$ denote the sum

$$\phi^{(1)}(x) = \sum_{j=1}^{N_{\text{over}}} \frac{w_j^{\text{over}}}{|x - s_j|}. \quad (4.15)$$

Using the FMM, we compute $\phi^{(1)}(x_i)$ and $\nabla \phi^{(1)}(x_i)$, $i = 1, 2 \ldots N$, where $c_{1,j} = w_j^{\text{over}} \bar{\tau}_j$, and $v_{1,j} = 0, j = 1, 2, \ldots N_{\text{over}}$. Let $\nabla_\ell S$ and $S$ denote the discretized versions of $\partial_\ell S$ and $S$ respectively, where $\partial_\ell S$ as before is the gradient of $S$ along the coordinate direction $\ell$, with $\ell = 1, 2, \text{ or } 3$. Then

$$S[\tau](x_i) = \phi^{(1)}(x_i) + \sum_{x_j \in \text{Near}(x_i)} \tilde{w}_{ij}^{\text{S}} \tau(x_j),$$

(4.16)

$$\partial_\ell S[\tau](x_i) = \partial_\ell \phi^{(1)}(x_i) + \sum_{x_j \in \text{Near}(x_i)} \tilde{w}_{ij}^{\partial_\ell \tau} \tau(x_j).$$

Let $\nabla S = [\partial_1 S; \partial_2 S; \partial_3 S]$ denote the discretized version of $\nabla S$, and let $\mu = [\mu^{(1)}; \mu^{(2)}; \mu^{(3)}]$ denote the discretized version of $\nabla_\tau S$, which can be expressed in terms of $\partial_\ell S$, and tangential derivatives $\partial_u X$, and $\partial_v X$ as follows

$$\mu(x_i) = (\nabla S(x_i) \cdot \partial_u X^\ell(u_i, v_i)) \left( g^{11} \partial_u X^\ell(u_i, v_i) + g^{12} \partial_v X^\ell(u_i, v_i) \right) + (\nabla S(x_i) \cdot \partial_v X^\ell(u_i, v_i)) \left( g^{21} \partial_u X^\ell(u_i, v_i) + g^{22} \partial_v X^\ell(u_i, v_i) \right), \quad (4.17)$$

where $x_i \in \Gamma_\ell$, $(u_i, v_i) \in T_0$ are the local coordinates of $x_i$ on $\Gamma_\ell$, and $g^{ij}, i, j = 1, 2$, are the components of the inverse of the metric tensor which are computed using $\partial_u X^\ell(u_i, v_i)$, and $\partial_v X^\ell(u_i, v_i)$.

Given the samples of $\mu$ at the oversampled nodes, let $\zeta = [\zeta^{(1)}; \zeta^{(2)}; \zeta^{(3)}]$ denote the sum

$$\zeta(x) = \sum_{j=1}^{N_{\text{over}}} \frac{w_j^{\text{over}} \tilde{\mu}_j}{|x - s_j|}. \quad (4.18)$$

Using a second call to the vector version of the FMM, we compute $\zeta(x_i)$, and its gradient $\nabla \zeta(x_i), i = 1, 2 \ldots N$, where $c_{\ell,j} = w_j^{\text{over}} \bar{\zeta}_j^{(\ell)}$, and $v_{\ell,j} = 0, \ell = 1, 2, 3, \text{ and } j =$
1, 2 \ldots N_{\text{over}}. The discretized version of $\nabla \cdot S[\mu]$ denoted by $\nabla \cdot S[\mu]$ is given by

$$
\nabla \cdot S[\mu](x_i) = 3 \sum_{\ell=1} \left( \partial_{\ell} \xi^{(\ell)}(x_i) \right) + \sum_{x_j \in \text{Near}(x_i)} w_{ij} \delta_{ij} S^{(\ell)}(x_j).
$$

(4.19)

In order to handle the inclusion of $SWS[\tau]$, we precompute and store the discretized version of $S[1]$ denoted by $\phi_0 = S[1]$. The computation of $\phi_0$ is identical to the computation of $S[\tau]$. This cost of evaluating $\phi_0$ is then amortized across the number of GMRES iterations. Let $\eta$ be the constant given by

$$
\eta = \frac{\sum_{j=1}^{N} S[\tau](x_j) w_j}{\sum_{j=1}^{N} w_j}.
$$

(4.20)

Combining the computations above, the discretized version of the integral operator in Eq. (4.8) acting on the density $\tau$ is given by

$$
\nabla \cdot S\nabla \cdot S[\tau] + SWS[\tau] = \nabla \cdot S[\mu] + \eta \phi_0.
$$

(4.21)

### 4.2 FMM-Accelerated Application of $A_2$

In this section, we discuss the FMM-accelerated evaluation of

$$
A_2[\tau] = -\frac{\tau}{4} + D^2[\tau] - S(S'' + D' + 2HS')[\tau] + SWS[\tau].
$$

(4.22)

Let $\phi^{(1)}$, $\phi^{(2)}$ denote the sums

$$
\phi^{(1)}(x) = \sum_{j=1}^{N_{\text{over}}} \frac{w_{\text{over}}^{j} \bar{\tau}_j}{|x - s_j|},
$$

$$
\phi^{(2)}(x) = -\sum_{j=1}^{N_{\text{over}}} w_{\text{over}}^{j} \bar{\tau}_j n(s_j) \cdot \nabla_x \frac{1}{|x - s_j|},
$$

(4.23)

where $n(s_j), j = 1, 2, \ldots N_{\text{over}},$ are the interpolated values of the normal vector at the oversampled nodes. Using the vector version of the FMM, we compute $\phi^{(\ell)}(x_i), \nabla \phi^{(\ell)}(x_i),$ and $\nabla \nabla \phi^{(\ell)}(x_i), \ell = 1, 2; i = 1, 2 \ldots N,$ where $c_{1,j} = w_{\text{over}}^{j} \bar{\tau}_j, c_{2,j} = 0, v_{1,j} = 0,$ and $v_{2,j} = w_{\text{over}}^{j} \bar{\tau}_j n(s_j), j = 1, 2, \ldots N_{\text{over}}.$ Let $S, D, S', S'' + D'$ denote the discretized versions of $S, D, S',$ and $S'' + D'$ respectively. Then,

$$
S[\tau](x_i) = \phi^{(1)}(x_i) + \sum_{x_j \in \text{Near}(x_i)} w_{ij} S^{(\ell)}(x_j),
$$

$$
D[\tau](x_i) = \phi^{(2)}(x_i) + \sum_{x_j \in \text{Near}(x_i)} w_{ij} D^{(\ell)}(x_j),
$$

$$
S'[\tau](x_i) = n(x_i) \cdot \nabla \phi^{(1)}(x_i) + \sum_{x_j \in \text{Near}(x_i)} w_{ij} S'(x_j),
$$

(4.24)
and
\[
(S'' + D')[\tau](x_i) = n(x_i) \cdot \nabla \nabla \phi^{(1)}(x_i) \cdot n(x_i) + n(x_i) \cdot \nabla \phi^{(2)}(x_i)
+ \sum_{j=1}^{N} \tilde{w}_{ij}^{(S'' + D')} \tau(x_j).
\] (4.25)

Given these four quantities, let \( \mu^{(1)} = D[\tau] \), and let \( \mu^{(2)} \) be given by
\[
\mu^{(2)}(x_i) = -(S'' + D')[\tau](x_i) - 2H(x_i)S'[\tau](x_i) + \frac{\sum_{j=1}^{N} w_j S[\tau](x_j)}{\sum_{j=1}^{N} w_j}.
\] (4.26)

The last term in the sum above is the discretized version of \( \mathcal{W}S[\tau] \).

Given the interpolated values of the densities \( \mu^{(\ell)} \) at the oversampled nodes, and let \( \zeta^{(\ell)} \), \( \ell = 1, 2 \), denote the sums
\[
\zeta^{(1)}(x) = -\sum_{j=1}^{N_{\text{over}}} w_{ij}^{\text{over}} \tilde{\mu}_{ij}^{(1)} n(s_j) \cdot \nabla x \frac{1}{|x - s_j|},
\]
\[
\zeta^{(2)}(x) = \sum_{j=1}^{N_{\text{over}}} w_{ij}^{\text{over}} \tilde{\mu}_{ij}^{(2)} \frac{1}{|x - s_j|}.
\] (4.27)

Using a second call to the vector version of the FMM, we compute \( \zeta^{(\ell)}(x_i) \), \( \ell = 1, 2 \), \( i = 1, 2 \ldots N \), where \( c_{1,j} = 0 \), \( c_{2,j} = w_{ij}^{\text{over}} \tilde{\mu}_{ij}^{(1)} \), \( v_{1,j} = w_{ij}^{\text{over}} \tilde{\mu}_{ij}^{(2)} n(s_j) \) and \( v_{2,j} = 0 \), \( j = 1, 2 \ldots N_{\text{over}} \). As before, let \( D[\mu^{(1)}] \), and \( S[\mu^{(2)}] \) denote the discretized versions of \( \mathcal{D}[\mu^{(1)}] \) and \( \mathcal{S}[\mu^{(2)}] \), then
\[
D[\mu^{(1)}](x_i) = \zeta^{(1)}(x_i) + \sum_{j=1}^{N} \tilde{w}_{ij}^{D} \mu^{(1)}(x_j),
\]
\[
S[\mu^{(2)}](x_i) = \zeta^{(2)}(x_i) + \sum_{j=1}^{N} \tilde{w}_{ij}^{S} \mu^{(2)}(x_j).
\] (4.28)

Finally, the discretized version of the integral operator in Eq. (4.10) acting on the density \( \tau \), is given by
\[
-\frac{\tau}{4} + D^2[\tau] - S(S'' + D' + 2HS')[\tau] + \mathcal{W}S[\tau] = -\frac{\tau}{4} + S[\mu^{(2)}] - D[\mu^{(1)}].
\] (4.29)

4.3 FMM-Accelerated Application of \( A_3 \)

In this section, we discuss the FMM-accelerated evaluation of
\[
A_3[\tau] = -\frac{\tau}{4} + (S'')^2[\tau] - (S'' + D' + 2HS')S[\tau] + \mathcal{W}S^2[\tau].
\] (4.30)


Table 1  Summary of FMMs used in discretizations of Eqs. (4.8), (4.10) and (4.11)

| Representation | # of densities | Interaction kernel | Output                        |
|---------------|----------------|--------------------|-------------------------------|
| \( A_1 \)     | 1st FMM 1      | Charge             | Potential and Gradient        |
|               | 2nd FMM 3      | Charge             | Potential and Gradient        |
| \( A_2 \)     | 1st FMM 2      | Charge, and Dipole | Potential, Gradient, and Hessian |
|               | 2nd FMM 2      | Charge, and Dipole | Potential                     |
| \( A_3 \)     | 1st FMM 1      | Charge             | Potential and Gradient        |
|               | 2nd FMM 3      | Charge, and Dipole | Potential, Gradient, and Hessian |

Let \( \phi^{(1)}(x) \) denote the sum

\[
\phi^{(1)}(x) = \sum_{j=1}^{N_{\text{over}}} w_j^{\text{over}} \tau_j \frac{x}{|x - s_j|}.
\]  

Using the FMM, we compute \( \phi^{(1)}(x_i), \nabla \phi^{(1)}(x_i), i = 1, 2 \ldots N \), where \( c_{1,j} = w_j^{\text{over}} \bar{v}_j \), and \( v_{1,j} = 0, j = 1, 2 \ldots N_{\text{over}} \). As before, let \( S \) and \( S' \) denote the discretized versions of \( S \), and \( S' \) respectively. Then

\[
S[\tau](x_i) = \phi^{(1)}(x_i) + \sum_{j=1}^{N} \tilde{w}_{ij}^S \tau(x_j),
\]  

\[
S'[\tau](x_i) = \nabla \phi^{(1)}(x_i) + \sum_{j=1}^{N} \tilde{w}_{ij}^{S'} \tau(x_j).
\]  

Given these two quantities, let \( \mu^{(1)} = S'[\tau] \), and \( \mu^{(2)} = S[\tau] \). Let \( \zeta^{(\ell)} \), \( \ell = 1, 2, 3 \) denote the sums

\[
\zeta^{(1)}(x) = \sum_{j=1}^{N_{\text{over}}} w_j^{\text{over}} \mu_j^{(1)} \frac{x}{|x - s_j|},
\]

\[
\zeta^{(2)}(x) = \sum_{j=1}^{N_{\text{over}}} w_j^{\text{over}} \mu_j^{(2)} \frac{x}{|x - s_j|},
\]

\[
\zeta^{(3)}(x) = -\sum_{j=1}^{N_{\text{over}}} w_j^{\text{over}} \nabla \mu_j^{(2)} \cdot \nabla \frac{1}{|x - s_j|}.
\]

Using a second call to the vector version of the FMM, we compute \( \zeta^{(\ell)}(x_i), \ell = 1, 2, 3, i = 1, 2 \ldots N \), where \( c_{1,j} = w_j^{\text{over}} \mu_j^{(1)} \), \( c_{2,j} = w_j^{\text{over}} \mu_j^{(2)} \), \( c_{3,j} = 0 \), \( v_{1,j} = v_{2,j} = 0 \), and \( v_{3,j} = w_j^{\text{over}} \mu_j^{(2)} n(s_j), j = 1, 2 \ldots N_{\text{over}} \). As before, let \( S'[\mu^{(1)}], S[\mu^{(2)}], S'[\mu^{(2)}] \) and \( (S'' + D')[\mu^{(2)}] \) denote the discretized versions of \( S[\mu^{(1)}], S[\mu^{(2)}], S'[\mu^{(2)}] \) and \( (S'' + D')[\mu^{(2)}] \).
\( D' \)[\( \mu^{(2)} \)] respectively. Then
\[
S'[\mu^{(1)}](x_i) = n(x_i) \cdot \nabla \zeta^{(1)}(x_i) + \sum_{j=1}^{N} \tilde{w}^{S'}_{ij} \mu^{(1)}(x_j), \\
S[\mu^{(2)}](x_i) = \zeta^{(2)}(x_i) + \sum_{j=1}^{N} \tilde{w}^{S}_{ij} \mu^{(2)}(x_j), \\
S'[\mu^{(2)}](x_i) = n(x_i) \cdot \nabla \zeta^{(2)}(x_i) + \sum_{j=1}^{N} \tilde{w}^{S'}_{ij} \mu^{(2)}(x_j),
\]
\( (S'' + D')[\mu^{(2)}](x_i) = n(x_i) \cdot \nabla \nabla \zeta^{(2)}(x_i) \cdot n(x_i) + n(x_i) \cdot \nabla \zeta^{(3)}(x_i) + \sum_{j=1}^{N} \tilde{w}^{(S'' + D')}_{ij} \mu^{(2)}(x_j). \)

Having computed \( S[\mu^{(2)}] = S^2[\tau] \), let \( \eta \) be the discretized version of \( WS^2[\tau] \) given by
\[
\eta = \frac{\sum_{j=1}^{N} S[\mu^{(2)}](x_j) w_j}{\sum_{j=1}^{N} w_j}. \tag{4.36}
\]

Finally, the discretized version of the integral operator in Eq. (4.11) acting on the density \( \tau \), is given by
\[
-\frac{\tau}{4} + (S')^2[\tau] - (S'' + D' + 2H S') S[\tau] + WS^2[\tau] \\
= -\frac{\tau}{4} + S'[\mu^{(1)}] - (S'' + D')[\mu^{(2)}] - 2H S'[\mu^{(2)}] + \eta. \tag{4.37}
\]

### 4.4 Summary of FMM-Accelerations

The rate limiting step in the solution of the Laplace Beltrami problem is the evaluation of the quadrature corrections \( \tilde{w}^{K}_{ij} \), and the FMM sums for all three integral equations. The computational performance of the fast multipole method depends on the distribution of sources and targets (which is identical for all three integral equations), whether the FMM sums are evaluating charge sums, or dipole sums, or charges and dipole sums, and whether just the potential is requested, or potentials and gradients, or potentials, gradients and Hessians. The numerical implementation of each of the three integral representations requires two calls to the vector FMM, where the sum of the total number of densities is four for all three representations. We summarize the specifics of the FMM sums for all three integral equations in the Table 1.
5 Harmonic Vector Fields

For a surface $\Gamma$ with unit normal $n$, a harmonic vector field $H$ is a tangential vector field which satisfies

$$\nabla_{\Gamma} \cdot H = 0, \quad \nabla_{\Gamma} \cdot (n \times H) = 0.$$  \hspace{1cm} (5.1)

On a genus $g$ surface, it is well known that the space of harmonic vector fields is $2g$ dimensional \cite{9, 11, 12}. Moreover, it follows from the definition that if $H$ is a harmonic vector field, then $n \times H$ is also a harmonic vector field which is linearly independent of $H$. Any tangential $C^1$ vector field $F$ along a smooth surface admits a Hodge decomposition \cite{11, 12}:

$$F = \nabla_{\Gamma} \alpha + n \times \nabla_{\Gamma} \beta + H,$$  \hspace{1cm} (5.2)

where $\alpha$ and $\beta$ are scalar functions defined on the surface $\Gamma$ and $H$ is a harmonic vector field. Given a smooth tangential vector field $F$, one can use the existence of this decomposition to compute linearly independent harmonic vector fields $H$ by solving the following Laplace–Beltrami equations and subsequently computing $H$:

$$\Delta_{\Gamma} \alpha = \nabla_{\Gamma} \cdot F,$$

$$\Delta_{\Gamma} \alpha = -\nabla_{\Gamma} \cdot (n \times F),$$

$$H = F - \nabla_{\Gamma} \alpha - n \times \nabla_{\Gamma} \beta.$$  \hspace{1cm} (5.3)

A convenient way of choosing such smooth vector fields $F$ is to first define a vector field $V \in \mathbb{R}^3$ where each component is a low-degree random polynomial, and then set $F = n \times n \times V$. We use scaled Legendre polynomials for this purpose. The maximum degree of Legendre polynomials should be at least $\lceil (2g)^{1/3} \rceil$ so that dimension of the space of polynomials is greater than the dimension of the space of the vector fields. For the examples in this paper, we choose the degree to be $\lceil (2g)^{1/3} \rceil + 3$. Moreover, suppose that $\Gamma$ is contained in $(-L, L)^3$, then each component is defined to be a scaled tensor product Legendre polynomial $P_\ell(x/2L)P_m(y/2L)P_n(z/2L)$, where $P_n(x)$ is the standard Legendre polynomials of degree $n$ on $[-1, 1]$. The extra factor of 2 in the choice of Legendre polynomials is to avoid requiring additional degrees of freedom to represent $F$ on the surface due to large gradients of the Legendre polynomials near the end points $-1$ and $1$.

With high probability, different random polynomial vector fields $F$ result in different projections on the different elements of the $2g$-dimensional space of vector fields. It is easy to show that if $H$ is a harmonic vector field then $n \times H$ is also a harmonic vector field which is linearly independent of $H$. Thus, in practice one can construct a basis for the harmonic vector field subspace by computing $H$ for $g$ different vector fields, and the remaining $g$ vector fields of the basis are defined as $n \times H$.

In the next section, we present a battery of numerical results demonstrating the accuracy and efficiency of our FMM-accelerated solver and its applications.

6 Numerical Examples

In this section, we provide several numerical examples demonstrating the accuracy and computational efficiency of our solver for the Laplace–Beltrami problem. The code was implemented in Fortran and compiled using the GNU Fortran 12.2 compiler. We use the point-based FMMs from the FMM3D package (https://github.com/flatironinstitute/FMM3D), and the locally-corrected quadratures from the fmm3dbie package (https://github.com/
Table 2  Summary of integral representations and the corresponding integral equations

| Operator | Representation | Integral equation |
|----------|----------------|------------------|
| $A_1$    | $u = S[\sigma]$ | $\nabla \cdot S_\Gamma[\sigma] + SWS[\sigma] = f$ |
| $A_2$    | $u = S[\sigma]$ | $-\frac{\sigma}{4} + D^2[\sigma] - S(S'' + D' + 2HS)[\sigma] + SWS[\sigma] = f$ |
| $A_3$    | $u = S^2[\sigma]$ | $-\frac{\sigma}{4} + (S')^2[\sigma] - (S'' + D' + 2HS)S[\sigma] + WS^2[\sigma] = f$ |

We compare the performance of the three different approaches to solve the Laplace–Beltrami problem. We summarize the different integral representations, and the corresponding integral equations being solved in Table 2.

In the following, let $p$ denote the discretization order, and $N$ denote the total number of discretization points on the surface. Let $\epsilon$ denote the tolerance for evaluating the quadrature corrections, and the tolerance for computing the FMM sums. Let $t_q$ denote the time taken for computing the quadrature corrections, $t_{nu}$ denote the time taken for applying the discretized integral operators $A_j$, and let $t_s$ denote the time taken to solve the linear system. Finally, let $n_{iter}$ denote the number of iterations required for the relative residual in GMRES to drop below the prescribed tolerance $\epsilon_{\text{GMRES}}$. We cap the maximum number of GMRES iterations at 100.

6.1 Speed and Accuracy Performance

To illustrate the accuracy of our solvers, consider the Laplace–Beltrami problem on a sphere with data given by a random 80th order spherical harmonic expansion,

$$ f(\theta, \phi) = \text{Re} \left( \sum_{n=1}^{80} \sum_{m=0}^{n} c_{n,m} Y_{n,m}(\theta, \phi) \right), \quad (6.1) $$

where $c_{n,m}$ are complex numbers with real and imaginary parts uniformly sampled in $(-0.5, 0.5)$. The analytic solution for this problem is given by

$$ u(\theta, \phi) = \text{Re} \left( \sum_{n=1}^{80} \sum_{m=0}^{n} - \frac{c_{n,m} Y_{n,m}(\theta, \phi)}{n(n+1)} \right). \quad (6.2) $$

The sphere is discretized by a stereographic projection of a triangulated cube. Let $u_{\text{comp}}$ denote the solution computed using any of the three integral equations. Let $\epsilon_s$ denote the relative error in the solution $u$ scaled by the by the boundary data given by

$$ \epsilon_s = \frac{|u_{\text{comp}} - u|_{L^2(\Gamma)}}{|f|_{L^2(\Gamma)}} \approx \sqrt{\sum_{j=1}^{N} \frac{|u_{\text{comp}}(\theta_j, \phi_j) - u(\theta_j, \phi_j)|^2 w_j}{\sum_{j=1}^{N} |f(\theta_j, \phi_j)|^2 w_j}}. \quad (6.3) $$

Here $(\theta_j, \phi_j)$ are the discretization nodes, and $w_j$ are the corresponding quadrature weights for integrating smooth functions.

We set $\epsilon = \epsilon_{\text{GMRES}}$, but use different values for $\epsilon_s$ depending on the value of $p$. In particular, we use the following combination of parameters: $(p, \epsilon) = (5, 5 \times 10^{-7})$, $(7, 5 \times 10^{-7})$, $(9, 5 \times 10^{-7})$, $(11, 5 \times 10^{-7})$. All CPU timings in these examples were obtained on a laptop with 8 cores of an Intel i9 2.4 GHz processor.
$10^{-9}$), and $(9.5 \times 10^{-11})$. Let $\rho_c$ denote the computed order of convergence given by $
abla \log_2(\varepsilon_s(N)/\varepsilon_s(4N))$.

Referring to Table 3, we observe that the solution converges at $O(h^{p+0.5})$. The fluctuating order of convergence for smaller values of $N$ can be attributed to the lack of resolution of the data $f$. Given the presence of order 0 kernels like $S'$ and $D$, one may have expected the discretized layer potentials to converge at $O(h^{p-1})$ [1, 19]. However, the added regularity in the solution obtained by applying $S$ or $S^2$, depending on the integral representation used, could account for the improved order of convergence.

With regards to CPU-time performance, we observe that $t_q$, $t_{mv}$, and $t_s$ scale linearly with $N$ (fluctuations in $t_{mv}$ for lower values of $N$ can be attributed to the cache effects). Among the three representations, the operator $A_3$ requires the smallest $t_q$ since it requires quadrature corrections for three kernels, as opposed to four for the other two representations. On the other hand, the $t_{mv}$ is smallest for $A_1$ since it requires only charge interactions for both the FMM calls, see Table 1.

Due to the well-conditioned nature of the problem, $n_{\text{iter}} = O(1)$ for $A_2$ and $A_3$, and thus the solve time $t_s$ is dominated by $t_{mv}$. Moreover, $n_{\text{iter}}$ is independent of the refinement of the mesh for $A_2$ and $A_3$. The ratio of $t_q$ and $t_{mv}$ lies between 10 and 40 for most configurations; this seems to indicate that when the iteration count exceeds 40, the quadrature time is no longer the dominant cost of solving the linear system. This might not be the case in many practical problems due to the well-conditioned nature of the integral equation.

We now turn our attention to the results for $A_1$. It is mathematically equivalent to $A_2$, but the GMRES iteration stalls and the iteration terminates because of reaching the maximum iteration count. This results in the largest solve time for $A_1$, despite having the smallest $t_{mv}$ per iteration. This behavior can be attributed to the lack of numerical second-kindness discussed in greater detail in Sect. 6.2 below.

To summarize, owing to its significantly smaller $t_q$, and small $n_{\text{iter}}$, $A_3$ has the best numerical performance amongst $A_1$, $A_2$, and $A_3$ for solving the Laplace–Beltrami problem.

**Remark 2** The operator $A_3$ also has an added analytical advantage over the other two representations: The map from $f \rightarrow u$ is a pseudo-differential operator of order $-2$. By representing the solution as $u = S^2[\sigma]$, the map from $\sigma \rightarrow u$ is also an pseudo-differential operator of order $-2$. Thus, $A_3$ tends to be more analytically faithful to the spectral properties of operators which contain compositions of solutions to the Laplace–Beltrami problem as compared to $A_1$ or $A_2$. This situation naturally arises when using the generalized-Debye formulation for solving Maxwell’s equations with perfect conductor or dielectric boundary conditions [11, 13], and also in modeling of type-I superconductors [14].
Table 3: Timing and accuracy results on the unit sphere for integral equations corresponding to $A_1$, $A_2$, and $A_3$, discretized with $p = 5$, 7, and 9 order patches

| Operator | $p$ | $N$   | $t_q$ | $t_{inv}$ | $t_s$   | $n_{iter}$ | $\varepsilon_s$ | $\rho_c$ |
|----------|-----|-------|-------|-----------|---------|------------|-----------------|----------|
| $A_1$    | 5   | 11,520| 9.5   | 0.5       | 61.1    | 100        | $5.5 \times 10^{-3}$ |          |
|          | 5   | 46,080| 37.6  | 1.8       | 248.2   | 100        | $3.3 \times 10^{-4}$ | 4.1      |
|          | 5   | 184,320| 171.8 | 7.8       | 1096.6  | 100        | $6.5 \times 10^{-6}$ | 5.7      |
|          | 7   | 21,504| 35.8  | 1.4       | 198.9   | 100        | $2.2 \times 10^{-3}$ |          |
|          | 7   | 86,016| 157.2 | 7.0       | 947.6   | 100        | $6.2 \times 10^{-5}$ | 5.1      |
|          | 7   | 344,064| 630.2 | 31.1      | 3931.3  | 100        | $2.2 \times 10^{-7}$ | 8.1      |
|          | 9   | 34,560| 206.0 | 6.0       | 976.9   | 100        | $1.1 \times 10^{-3}$ |          |
|          | 9   | 138,240| 837.1 | 33.0      | 4001.8  | 100        | $7.4 \times 10^{-6}$ | 7.21     |
|          | 9   | 552,960| 2988.2| 75.5      | 12,517.0| 100        | $6.8 \times 10^{-9}$ | 10.0     |
| $A_2$    | 5   | 11,520| 6.8   | 1.4       | 16.8    | 5          | $1.4 \times 10^{-3}$ |          |
|          | 5   | 46,080| 31.2  | 4.5       | 62.2    | 5          | $1.4 \times 10^{-4}$ | 3.3      |
|          | 5   | 184,320| 127.4 | 14.0      | 254.3   | 5          | $1.3 \times 10^{-7}$ | 10.0     |
|          | 7   | 21,504| 28.2  | 3.1       | 54.6    | 6          | $3.7 \times 10^{-4}$ |          |
|          | 7   | 86,016| 116.6 | 15.7      | 228.5   | 6          | $5.9 \times 10^{-7}$ | 9.29     |
|          | 7   | 344,064| 499.1 | 47.8      | 850.0   | 6          | $3.0 \times 10^{-9}$ | 7.61     |
|          | 9   | 34,560| 156.5 | 8.9       | 245.8   | 7          | $3.9 \times 10^{-5}$ |          |
|          | 9   | 138,240| 680.2 | 39.6      | 1046.7  | 7          | $5.4 \times 10^{-8}$ | 9.49     |
|          | 9   | 552,960| 2404.4| 109.0     | 3429.2  | 7          | $7.4 \times 10^{-11}$ | 9.51     |
| $A_3$    | 5   | 11,520| 6.0   | 0.9       | 11.8    | 5          | $1.4 \times 10^{-3}$ |          |
|          | 5   | 46,080| 24.4  | 3.4       | 50.4    | 5          | $1.4 \times 10^{-4}$ | 3.3      |
|          | 5   | 184,320| 114.6 | 12.4      | 187.3   | 4          | $4.6 \times 10^{-7}$ | 8.2      |
|          | 7   | 21,504| 23.4  | 2.8       | 41.8    | 6          | $3.7 \times 10^{-4}$ |          |
|          | 7   | 86,016| 96.0  | 13.6      | 199.4   | 6          | $5.9 \times 10^{-7}$ | 9.29     |
|          | 7   | 344,064| 377.4 | 57.1      | 722.3   | 5          | $4.0 \times 10^{-9}$ | 7.2      |
|          | 9   | 34,560| 123.8 | 8.5       | 196.1   | 7          | $3.9 \times 10^{-5}$ |          |
|          | 9   | 138,240| 501.3 | 30.7      | 822.8   | 7          | $5.4 \times 10^{-8}$ | 9.49     |
|          | 9   | 552,960| 1755.4| 104.3     | 2749.3  | 7          | $7.4 \times 10^{-11}$ | 9.51     |

$N$ is the total number of discretization points, $t_q$ is the time taken for computing quadrature corrections in seconds, $t_{inv}$ is the time taken to apply the FMM-accelerated discretized integral operators in seconds, $t_s$ is the time taken to solve the linear system using GMRES in seconds, $n_{iter}$ is the number of GMRES iterations required for the solution to converge to the prescribed GMRES tolerance, $\varepsilon_s$ is the accuracy of the computed solution, relative to the norm of the data, and $\rho_c$ is the estimated order of convergence.
6.2 Numerical Second-Kindness

The poorer performance of $A_1$ in an iterative solver as compared to $A_2$ can be attributed to the lack of numerical second-kindness in its implementation. Despite $A_1$ being mathematically equivalent to $A_2$, their numerical discretizations can have wildly different behaviors. We define a discretized numerical operator to be *numerically second-kind* if its spectral behavior, under refinement, converges to the spectral behavior of the associated continuous second-kind operator. This is usually accomplished via an explicit handling of the identity term contained in the continuous operator (which corresponds to an identity matrix in the discrete setting); this is the situation for operators $A_2$ and $A_3$. A standard Nyström or Galerkin discretization of the factors of $A_1$, and then composing them, does not mirror the spectral behavior of the associated continuous second-kind operator. Depending on the particular quadrature scheme used, most integral operators (including many hypersingular ones) have a discrete spectrum which clusters closely to the origin (due to an inexactness of nearly all quadrature schemes), resulting in numerical conditioning issues. An identical behavior was discussed in Section 3.2 of [25] in the context of applying layer potential operators using Quadrature by Expansion (QBX). Explicitly adding the identity term, as in the case of $A_2$, results in the spectrum of the discretized integral operator clustering at $-1/4$. This greatly enhances the convergence of iterative methods such as GMRES [39].

To avoid any additional errors due to using the FMM, we illustrate this behavior on a smaller problem which would allow for using dense linear algebra routines. Consider the Laplace–Beltrami operator on the sphere with data given by a random fourth-order spherical harmonic expansion,

$$f(\theta, \phi) = \text{Re} \left( \sum_{n=1}^{4} \sum_{m=0}^{n} c_{n,m} Y_{n,m}(\theta, \phi) \right),$$

where $c_{n,m}$ as before are complex numbers with real and imaginary parts uniformly sampled in $(-0.5, 0.5)$. We restrict our attention to $p = 6$ and $N = 4,032$ and 16,128. The linear systems are solved using GMRES with $\varepsilon_{\text{GMRES}} = 10^{-14}$ and $\varepsilon = 5 \times 10^{-8}$. The matrices corresponding to the integral equations are precomputed, and at every iteration the matrix vector product is computed using BLAS routines. Referring to Fig. 1, we see that the GMRES residuals for $A_2$ and $A_3$ drop exponentially with iteration number, while the residuals for $A_1$ decrease at a far slower rate and eventually stall. Furthermore, referring to Table 4, we see that the singular values of the operators validate the lack of numerical second-kindness of $A_1$ as well as the stability of condition number under refinement for the numerically second-kind $A_2$ and $A_3$.

6.3 Harmonic Fields

In this section, we present the numerical results for the computation of the harmonic vector fields on two geometries: a twisted torus geometry whose boundary $\Gamma$ is parametrized by $X : [0, 2\pi]^2 \rightarrow \Gamma$ with

$$X(u, v) = \sum_{i=-1}^{2} \sum_{j=-1}^{2} \delta_{i,j} \begin{pmatrix} \cos v \cos((1-i)u + jv) \\ \sin v \cos((1-i)u + jv) \\ \sin((1-i)u + jv) \end{pmatrix},$$

where $c_{n,m}$ as before are complex numbers with real and imaginary parts uniformly sampled in $(-0.5, 0.5)$. We restrict our attention to $p = 6$ and $N = 4,032$ and 16,128. The linear systems are solved using GMRES with $\varepsilon_{\text{GMRES}} = 10^{-14}$ and $\varepsilon = 5 \times 10^{-8}$. The matrices corresponding to the integral equations are precomputed, and at every iteration the matrix vector product is computed using BLAS routines. Referring to Fig. 1, we see that the GMRES residuals for $A_2$ and $A_3$ drop exponentially with iteration number, while the residuals for $A_1$ decrease at a far slower rate and eventually stall. Furthermore, referring to Table 4, we see that the singular values of the operators validate the lack of numerical second-kindness of $A_1$ as well as the stability of condition number under refinement for the numerically second-kind $A_2$ and $A_3$. 
Fig. 1 GMRES residual as a function of iteration number for $\mathcal{A}_1$, $\mathcal{A}_2$, and $\mathcal{A}_3$ for a sphere discretized with sixth-order patches and $N = 4032$ and $N = 16128$

Table 4 Maximum and minimum singular values, as well as condition numbers, for the three formulations $\mathcal{A}_1$, $\mathcal{A}_2$, and $\mathcal{A}_3$ discussed in the paper

| Operator | $N$     | $\sigma_{\text{max}}$ | $\sigma_{\text{min}}$ | $\kappa = \sigma_{\text{max}}/\sigma_{\text{min}}$ |
|----------|---------|------------------------|------------------------|--------------------------------------------------|
| $\mathcal{A}_1$ | 4032    | $1.00$                 | $6.75 \times 10^{-7}$ | $1.49 \times 10^6$                              |
|           | 16,128  | $1.00$                 | $1.56 \times 10^{-8}$ | $6.42 \times 10^7$                              |
| $\mathcal{A}_2$ | 4,032   | $1.00$                 | $0.22$                 | $4.50$                                           |
|           | 16,128  | $1.00$                 | $0.22$                 | $4.50$                                           |
| $\mathcal{A}_3$ | 4032    | $1.00$                 | $0.22$                 | $4.50$                                           |
|           | 16,128  | $1.00$                 | $0.22$                 | $4.50$                                           |

where the non-zero coefficients are

\begin{align*}
\delta_{-1,-1} &= 0.17, \\
\delta_{-1,0} &= 0.11, \\
\delta_{0,0} &= 1, \\
\delta_{1,0} &= -0.25, \\
\delta_{0,1} &= 0.01, \\
\delta_{2,1} &= -0.45, \\
\end{align*}

and a genus-10 surface obtained by first constructing the surface as a union of rectangular faces. A high-order mesh for a smooth version of the geometry is obtained using the surface smoother algorithm of [42], see Fig. 3 for a graphical depiction. For both examples, the surface is discretized with order $p = 9$ patches. The quadrature and GMRES tolerances were set to $\varepsilon = 5 \times 10^{-8}$, and $\varepsilon_{\text{GMRES}} = 5 \times 10^{-10}$, respectively. In a slight abuse of notation, let $t_s$ now denote the time taken for computing the Hodge decomposition of a given vector field, which corresponds to solving two Laplace–Beltrami problems. Since the computation of the quadrature corrections tend to be the dominant cost, we reuse the computed quadrature corrections across the two solves. The solution to the Laplace–Beltrami problems were computed using the representation $\mathcal{A}_3$.

For the twisted torus example, let $V = \ell \times (x - x_0)/|x - x_0|^3$, with $x_0 = (0.2, 0.2, 0.2)$ and $\ell = (0, 1, 1)$, and then define $F = -n \times n \times V$. It is straightforward to see that a current source defined in the exterior projects onto the harmonic vector fields for the torus. In Fig. 2, we plot the tangential vector field $F$, its curl-free component, its divergence-free component and the harmonic vector field $H$. The second linearly independent harmonic vector field

\begin{align*}
\delta_{-1,-1} &= 0.17, \\
\delta_{-1,0} &= 0.11, \\
\delta_{0,0} &= 1, \\
\delta_{1,0} &= -0.25, \\
\delta_{0,1} &= 0.01, \\
\delta_{2,1} &= -0.45, \\
\end{align*}

and a genus-10 surface obtained by first constructing the surface as a union of rectangular faces. A high-order mesh for a smooth version of the geometry is obtained using the surface smoother algorithm of [42], see Fig. 3 for a graphical depiction. For both examples, the surface is discretized with order $p = 9$ patches. The quadrature and GMRES tolerances were set to $\varepsilon = 5 \times 10^{-8}$, and $\varepsilon_{\text{GMRES}} = 5 \times 10^{-10}$, respectively. In a slight abuse of notation, let $t_s$ now denote the time taken for computing the Hodge decomposition of a given vector field, which corresponds to solving two Laplace–Beltrami problems. Since the computation of the quadrature corrections tend to be the dominant cost, we reuse the computed quadrature corrections across the two solves. The solution to the Laplace–Beltrami problems were computed using the representation $\mathcal{A}_3$.

For the twisted torus example, let $V = \ell \times (x - x_0)/|x - x_0|^3$, with $x_0 = (0.2, 0.2, 0.2)$ and $\ell = (0, 1, 1)$, and then define $F = -n \times n \times V$. It is straightforward to see that a current source defined in the exterior projects onto the harmonic vector fields for the torus. In Fig. 2, we plot the tangential vector field $F$, its curl-free component, its divergence-free component and the harmonic vector field $H$. The second linearly independent harmonic vector field

\begin{align*}
\delta_{-1,-1} &= 0.17, \\
\delta_{-1,0} &= 0.11, \\
\delta_{0,0} &= 1, \\
\delta_{1,0} &= -0.25, \\
\delta_{0,1} &= 0.01, \\
\delta_{2,1} &= -0.45, \\
\end{align*}
Fig. 2 Hodge decomposition on a twisted torus with initial field $V$ given by a point current source at $x_0$. The color bar represents the magnitude of the vector field and the arrows represent the direction of the field (Color figure online)

Table 5 Results summary for computing harmonic vector fields on the twisted torus geometry

| $p$ | $N$      | $n_{\text{iter}}$ | $t_s$  | $||\nabla_{\Gamma} \cdot H||_{L^2(\Gamma)}$ | $||\nabla_{\Gamma} \cdot (n \times H)||_{L^2(\Gamma)}$ |
|-----|----------|-------------------|--------|------------------------------------------|--------------------------------------------------|
| 9   | 27,000   | (27,28)           | 317.1  | $1.4 \times 10^{-2}$                     | $1.0 \times 10^{-2}$                              |
| 9   | 108,000  | (27,28)           | 1182.0 | $2.8 \times 10^{-4}$                     | $1.5 \times 10^{-4}$                              |

can be obtained by computing $n \times H$. In Table 5, we tabulate the number of iterations for computing the curl-free and divergence-free parts of decomposition denoted by $n_{\text{iter}}$, and the solve time $t_s$. The error in the computation of the harmonic vector fields is estimated by computing the $L^2$ norm of $\nabla_{\Gamma} \cdot H$ and $\nabla_{\Gamma} \cdot (n \times H)$ using spectral differentiation. For this example $||H|| \approx 0.62498$, so the above error estimate serves as a proxy for the relative error as well.

We also compute the 20 linearly independent harmonic vector fields on a genus 10 surface. For this geometry, we choose a random surface vector field $V$ whose components are random tensor product Legendre polynomials as discussed in Sect. 5. In Fig. 3, we plot the surface along with these 20 linearly independent harmonic vector fields computed using our solver. As before, we verify the computations by calculating the $L^2$ norms of the surface divergence of $H$ and $n \times H$, where $H$ is the computed harmonic field. In Table 6, we tabulate the number of iterations for computing the curl-free and divergence-free parts of decomposition for one of the solves on the genus 10 surface, and also the computation time for obtaining the hodge decomposition.
Fig. 3  The 20 harmonic fields computed on a genus 10 geometry discretized with $N = 388800$ points. The color bar represents the magnitude of the vector field and the arrows represent the direction of the field (Color figure online)

Table 6  Results summary for computing harmonic vector fields on the genus 10 geometry

| $p$ | $N$    | $n_{\text{iter}}$ | $t_s$ | $||\nabla\Gamma \cdot \mathbf{H}||_{L^2(\Gamma)}$ | $||\nabla\Gamma \cdot \mathbf{n} \times \mathbf{H}||_{L^2(\Gamma)}$ |
|-----|--------|-------------------|------|----------------------------------|----------------------------------|
| 9   | 388,800| (19, 19)          | 2910.5| $3.2 \times 10^{-2}$             | $5.6 \times 10^{-3}$             |
| 9   | 1,555,200| (19, 18)        | 10,298.0| $5.8 \times 10^{-4}$             | $6.5 \times 10^{-5}$             |
The summarized results for both examples illustrate the linear scaling of the algorithm, and also the constancy of number of GMRES iterations under mesh refinement.

7 Conclusions and Future Work

In this paper, we have presented a high-order FMM-accelerated iterative solver for the numerical solution of the Laplace–Beltrami equation on complex smooth surfaces in three dimensions. The Laplace–Beltrami problem $\Delta_{\Gamma} \psi = f$ is converted to a second-kind integral equation by representing the solution as $\psi = S[\sigma]$ or $\psi = S^2[\sigma]$, and using appropriate Caldéron identities. The resulting integral equation, which requires the application of various Laplace layer potentials, is then discretized using a high-order method with locally corrected quadratures for computing the weakly-singular layer potentials; their application is accelerated using fast multipole methods.

While the integral equation can be written in several equivalent analytic forms, we demonstrate the necessity of maintaining numerical second-kindness by explicitly isolating the identity term of the Fredholm equation in order to avoid stagnation of iterative solvers such as GMRES. We also illustrate that the using the representation $\psi = S^2[\sigma]$ has the better numerical performance over using $\psi = S[\sigma]$ and preconditioning with $S$.

Finally, we also presented numerical examples demonstrating the computation of harmonic vector fields on surfaces using the Laplace–Beltrami solver. These vector fields are both of pure mathematical interest and are also required for solving problems in type I superconductivity and electromagnetism on topologically non-trivial geometries.

There are still several open questions that remain to be addressed. These include coupling high-order discretizations of the Laplace–Beltrami integral equations to fast direct solvers; subsequently using these solvers to develop iterative solvers for the solution of Maxwell’s equations using the generalized Debye formulation; and extending these ideas for the solution of surface diffusion problems that arise in pattern formation and cell biology. These are all ongoing areas of research.

Funding Dhwanit Agarwal’s research was supported in part by UT-Austin CoLab and UTEN Partnership, and the Portuguese Science and Technology Foundation under funding #201801976/UTA18-001217. Michael O’Neil’s research was supported in part by the Office of Naval Research under Award Numbers #N00014-17-1-2451 and #N00014-18-1-2307, and the Simons Foundation/SFARI (560651, AB).

Data availability The code used in the numerical examples in this manuscript is available in two publicly accessible git repositories: https://github.com/flatironinstitute/FMM3D. https://github.com/fastalgorithms/fmm3dbie.

Declarations

Conflict of interest The authors do not have any conflict of interest.

References

1. Atkinson, K.E., Chien, D.: Piecewise polynomial collocation for boundary integral equations. SIAM J. Sci. Comput. 16, 651–681 (1995)
2. Bänsch, E., Morin, P., Nochetto, R.H.: A finite element method for surface diffusion: the parametric case. J. Comput. Phys. 203(1), 321–343 (2005)
3. Beirão da Veiga, L., Brezzi, F., Marini, L.D., Russo, A.: The hitchhiker’s guide to the virtual element method. Math. Models Methods Appl. Sci. 24(08), 1541–1573 (2014)
4. Bertalmio, M., Cheng, L.-T., Osher, S., Sapiro, G.: Variational problems and partial differential equations on implicit surfaces. J. Comput. Phys. 174(2), 759–780 (2001)
5. Bonito, A., Demlow, A., Nochetto, R.H.: Finite element methods for the Laplace–Beltrami operator. In: Bonito, A., Nochetto, R.H. (eds.) Handbook of Numerical Analysis, vol. 21, pp. 1–103. Elsevier, Amsterdam (2020)
6. Bremer, J.: On the Nyström discretization of integral equations on planar curves with corners. Appl. Comput. Harmon. Anal. 32, 45–64 (2012)
7. Burman, E., Hansbo, P., Larson, M.G., Massing, A.: A cut discontinuous Galerkin method for the Laplace–Beltrami operator.IMA J. Numer. Anal. Anal. 37(1), 138–169 (2017)
8. Chen, Y., Macdonald, C.B.: The closest point method and multigrid solvers for elliptic equations on surfaces. SIAM J. Sci. Comput. 37(1), A134–A155 (2015)
9. Dai, Q.I., Chew, W.C., Jiang, L.J., Wu, Y.: Differential-forms-motivated discretizations of electromagnetic differential and integral equations. IEEE Antennas Wirel. Propag. Lett. 13, 1223–1226 (2014)
10. Demlow, A., Dziuk, G.: An adaptive finite element method for the Laplace–Beltrami operator on implicitly defined surfaces. SIAM J. Numer. Anal. 45(1), 421–442 (2007)
11. Epstein, C., Greengard, L.: Debye sources and the numerical solution of the time harmonic Maxwell equations. Commun. Pure Appl. Math. 63(4), 413–493 (2010)
12. Epstein, C.L., Gimbutas, Z., Greengard, L., Klöckner, A., O’Neil, M.: A consistency condition for the vector potential in multiply-connected domains. IEEE Trans. Magn. 49(3), 1072–1076 (2013)
13. Epstein, C.L., Greengard, L., Kropinski, M.C.A., Nigam, N.: Fast multipole methods for the evaluation of layer potentials with locally-corrected quadratures. J. Comput. Phys. X 10, 100092 (2021)
14. Greer, J.B., Bertozzi, A.L., Sapiro, G.: Fourth order partial differential equations on general geometries. J. Comput. Phys. 216(1), 216–246 (2006)
15. Kropinski, M.C.A., Nigam, N.: Fast integral equation methods for the Laplace–Beltrami equation on the sphere. Adv. Comput. Math. 40(2), 577–596 (2014)
16. Kropinski, M.C.A., Nigam, N., Quaife, B.: Integral equation methods for the Yukawa–Beltrami equation on the sphere. Adv. Comput. Math. 42(2), 469–488 (2016)
17. Macdonald, C.B., Ruuth, S.J.: Level set equations on surfaces via the closest point method. J. Sci. Comput. 35(2), 219–240 (2008)
18. Macdonald, C.B., Ruuth, S.J.: The implicit closest point method for the numerical simulation of partial differential equations on surfaces. SIAM J. Sci. Comput. 31(6), 4330–4350 (2010)
19. Malhotra, D., Cerfon, A., Imbert-Gérard, L.-M., O’Neil, M.: Taylor states in stellarators: a fast high-order boundary integral solver. J. Comput. Phys. 397, 108791 (2019)
34. Nedelec, J.: Acoustic and Electromagnetic Equations. Springer, New York (2001)
35. O’Neil, M.: Second-kind integral equations for the Laplace–Beltrami problem on surfaces in three dimensions. Adv. Comput. Math. 44(5), 1385–1409 (2018)
36. Plaza, R.G., Sanchez-Garduno, F., Padilla, P., Barrio, R.A., Maini, P.K.: The effect of growth and curvature on pattern formation. J. Dyn. Differ. Equ. 16(4), 1093–1121 (2004)
37. Rahimian, A., Lashuk, I., Veerapaneni, S., Chandramowlishwaran, A., Malhotra, D., Moon, L., Sampath, R., Shringarpure, A., Vetter, J., Vuduc, R., et al.: Petascale direct numerical simulation of blood flow on 200k cores and heterogeneous architectures. In: SC’10: Proceedings of the 2010 ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis, pp. 1–11. IEEE (2010)
38. Rosenberg, S.: The Laplacian on a Riemannian Manifold. Cambridge University Press, Cambridge (1997)
39. Saad, Y., Schultz, M.H.: GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. SIAM J. Sci. Stat. Comput. 7, 856–869 (1986)
40. Sifuentes, J., Gimbutas, Z., Greengard, L.: Randomized methods for rank-deficient linear systems. Electron. Trans. Numer. Anal. 44, 177–188 (2015)
41. Veerapaneni, S.K., Rahimian, A., Biros, G., Zorin, D.: A fast algorithm for simulating vesicle flows in three dimensions. J. Comput. Phys. 230, 5610–5634 (2011)
42. Vico, F., Greengard, L., O’Neil, M., Rachh, M.: A fast boundary integral method for high-order multiscale mesh generation. SIAM J. Sci. Comput. 42(2), A1380–A1401 (2020)
43. Vioreanu, B., Rokhlin, V.: Spectra of multiplication operators as a numerical tool. SIAM J. Sci. Comput. 36, A267–A288 (2014)
44. Wang, M., Leung, S., Zhao, H.: Modified virtual grid difference for discretizing the Laplace–Beltrami operator on point clouds. SIAM J. Sci. Comput. 40(1), A1–A21 (2018)

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.