Computation of Galerkin Double Surface Integrals in the 3-D Boundary Element Method
Ross Adelman, Nail A. Gumerov, and Ramani Duraiswami

Abstract—The Galerkin boundary element method (BEM), also known as the method of moments, is a powerful method for solving the Laplace equation in three dimensions. There are advantages to Galerkin formulations for integral equations, as they treat problems associated with kernel singularity, and lead to symmetric and better conditioned matrices. However, the Galerkin method requires the computation of double surface integral over pairs of triangles. There are many semianalytical methods to treat these integrals, which all have some issues and are discussed in this paper. Novel methods inspired by the treatment of these kernels in the fast multipole method are presented for computing all the integrals that arise in the Galerkin formulation to any accuracy. Integrals involving completely geometrically separated triangles are nonsingular, and are computed using a technique based on spherical harmonics and multipole expansions and translations, which require the integration of polynomial functions over the triangles. Integrals involving cases where the triangles have common vertices, edges, or are coincident are treated via scaling and symmetry arguments, combined with automatic recursive geometric decomposition of the integrals. The methods are validated, and example results are presented.

Index Terms—Boundary element methods (BEMs), boundary integral equations, Galerkin method, integral equations, Laplace equation, method of moments.

I. INTRODUCTION

The Galerkin boundary element method (BEM), also known as the method of moments, is a powerful method for solving the Laplace equation in three dimensions [1]–[3]. When the boundary is discretized using triangular elements, constructing the system matrix requires computing double surface integrals over pairs of these triangles. Because the kernels being integrated are singular, these integrals can be difficult to compute, especially when the two triangles are proximate, share a vertex, an edge, or are the same. Depending on the relative geometry of the two triangles, there are many different methods for computing them. For example, when the two triangles do not touch, the integral is completely regular and can be computed via numerical means, e.g., Gaussian quadrature. Semianalytical methods, where the inside integral is computed analytically and the outside integral is computed numerically, have also been proposed [4], [5].

However, in the cases when the two triangles share a vertex, an edge, or are the same, the integrals become much more complicated. There are analytical expressions for the case when the two triangles are the same [6], but not for when they share only a vertex or an edge. In these cases, the semianalytical methods do not always work. This is because, depending on the kernel being integrated, the inside integral can be hypersingular. While there are analytical expressions available for the inside integral, these expressions are singular along the corners and edges of the corresponding triangle. When the two triangles share a vertex or an edge, these singularities are included in the outside integral. The usual semianalytical methods will not work in these cases because they are not designed to properly handle the singularities.

The double integrals are weakly singular, so while the inside integrals may be hypersingular and the expressions for them may be singular in some places, they are completely integrable. Nevertheless, actually integrating them in practice can be hard. Therefore, more sophisticated semianalytical methods have been developed over the years. These include: singularity subtraction and “to the boundary” techniques [7]–[13]; singularity cancellation techniques [14], [15]; specialized quadrature methods that are designed for the singularities involved, such as those based on the double exponential formula [16], [17]; and other regularization methods, such as the Duffy transformation [18]–[20]. Many of these methods work very well, but because they all attempt to tackle the singularity issue directly, their analysis is very involved.

In this paper, we present a method for computing the integrals that completely avoids the computation of singular integrals. The approach relies on several scaling properties of the integrals and the kernels being integrated. When two triangles share a vertex, an edge, or are the same, the integral is decomposed into several smaller integrals, some of which are related back to the original integral via scaling and symmetry arguments. This is done in such a way that only regular integrals need to be computed explicitly. Any integrals involving singularities are computed implicitly during the procedure. The regular integrals can be computed using standard semianalytical methods, but in this paper, we also present an arbitrarily accurate approximate analytical method for doing so. This method uses spherical harmonics and multipole and local expansions
and translations. The only source of error in the method is from truncating these expansions. However, this error is precisely controlled by choosing the appropriate truncation number or recursively subdividing the problem. Finally, we verify the accuracy of these methods, and use them to solve some example problems.

II. BACKGROUND

The Galerkin BEM is a powerful method for solving the Laplace equation in three dimensions. Consider the following boundary value problem (BVP):

$$\nabla^2 \phi (\mathbf{x}) = 0, \quad \mathbf{x} \in \Omega. \tag{1}$$

Dirichlet boundary conditions are enforced on both sides of the boundary, and are given by

$$\alpha^+(\mathbf{x}) \phi^+(\mathbf{x}) + \alpha^-\phi^-(\mathbf{x}) = \gamma_1(\mathbf{x}) \quad (2)$$

$$\alpha^+(\mathbf{x}) \phi^+(\mathbf{x}) + \alpha^-\phi^-(\mathbf{x}) = \gamma_2(\mathbf{x}) \quad (3)$$

where $$\mathbf{x} \in \Gamma$$. In addition, appropriate boundary conditions must be set at infinity. The potential should decay to zero at large distances

$$\lim_{|\mathbf{x}| \to \infty} \phi(\mathbf{x}) = 0. \tag{4}$$

The boundaries can be closed or open. See Fig. 1 for some example boundaries. For closed boundaries, the interior side of the boundary is the “−” side and the exterior side is the “+” side. For open boundaries, since there is no inside or outside, designating each side of the boundary as “+” or “−” can be done arbitrarily. The values, $$\phi^+$$ and $$\phi^-$$, are the potential on the “+” and “−” sides of the surface, respectively.

To solve the BVP, we use an indirect boundary integral formulation called the layer potential formulation. Using Green’s theorem, the Laplace equation is transformed from a differential equation into an integral equation

$$\phi(\mathbf{x}) = L[\sigma](\mathbf{x}) + M[\mu](\mathbf{x}) \tag{5}$$

where

$$L[\sigma](\mathbf{x}) = \int_{\mathbf{x'} \in \Gamma} \sigma(\mathbf{x'}) G(\mathbf{x} - \mathbf{x'}) dS(\mathbf{x'}) \quad (6)$$

$$M[\mu](\mathbf{x}) = \int_{\mathbf{x'} \in \Gamma} \mu(\mathbf{x'}) (\mathbf{n'} \cdot \nabla_{\mathbf{x'}}) G(\mathbf{x} - \mathbf{x'}) dS(\mathbf{x'}) \quad (7)$$

are the single- and double-layer potentials [21], and $$G(\mathbf{r}) = 1/(4\pi |\mathbf{r}|)$$ is the Green’s function for the Laplace equation. The single-layer potential $$L[\sigma](\mathbf{x})$$ is the potential due to the monopole source density distribution $$\sigma(\mathbf{x'})$$ on the boundary. Likewise, the double-layer potential $$M[\mu](\mathbf{x})$$ is the potential due to the dipole source density distribution $$\mu(\mathbf{x'})$$ on the boundary. In the differential equation, we seek a solution to the potential governed by the Laplace equation. However, in the integral equation, we seek the source density distributions $$\sigma(\mathbf{x'})$$ and $$\mu(\mathbf{x'})$$ on the boundary that give rise to that potential. The advantage of the BEM is that the expression in (5) relating the source density distributions back to the potential automatically satisfies the original differential equation. Moreover, the Green’s function satisfies the boundary conditions at infinity, so as long as the source density distributions are bounded and finite, the single- and double-layer potentials will satisfy them as well. Thus, we need only concern ourselves with searching for the source density distributions that satisfy the remaining boundary conditions. To do this, we need to express the potential on either side of the boundary in terms of the source density distributions. Jump conditions provide such a relationship

$$\phi^+(\mathbf{x}) = L[\sigma](\mathbf{x}) + M[\mu](\mathbf{x}) \pm \frac{1}{2} \mu(\mathbf{x}). \tag{8}$$

Plugging (8) into the boundary conditions and rearranging

$$a_k L[\sigma] + a_k M[\mu] + b_k \mu = \gamma_k \quad (9)$$

where

$$a_k = \alpha^+_k + \alpha^-_k, \quad b_k = \frac{1}{2} (\alpha^+_k - \alpha^-_k) \quad (10)$$

and $$k = 1, 2$$. We have dropped the argument $$\mathbf{x}$$ to save space [i.e., $$\mu(\mathbf{x})$$ becomes $$\mu$$]. In order to make the problem computationally tractable, the source density distributions $$\sigma(\mathbf{x'})$$ and $$\mu(\mathbf{x'})$$ are each written as a linear combination of $$N$$ basis functions

$$\sigma(\mathbf{x'}) = \sum_{j=1}^{N} \sigma_j f_j(\mathbf{x'}), \quad \mu(\mathbf{x'}) = \sum_{j=1}^{N} \mu_j f_j(\mathbf{x'}). \tag{11}$$

For constant triangular elements, there is one basis function per triangle that is equal to one on that element and zero everywhere else. For linear triangular elements, there is one basis function per vertex that is equal to one at that vertex, zero at all the other vertices, and piecewise linear everywhere else. We need to compute the coefficients of these basis functions so that the boundary conditions are satisfied. In other words, we seek $$\sigma_1, \sigma_2, \ldots, \sigma_N, \mu_1, \mu_2, \ldots, \mu_N$$ such that

$$\sum_{j=1}^{N} \sigma_j A_k[f_j] + \sum_{j=1}^{N} \mu_j B_k[f_j] = \gamma_k \quad (12)$$

where

$$A_k[f_j] = a_k L[f_j], \quad B_k[f_j] = a_k M[f_j] + b_k f_j \quad (13)$$

and $$k = 1, 2$$. 

Fig. 1. Several different types of boundaries: $$\Gamma_1$$ is closed; $$\Gamma_2$$ is closed, but contains another boundary, $$\Gamma_3$$, also closed; $$\Gamma_4$$ and $$\Gamma_5$$ form a closed region, but this region is divided into two by $$\Gamma_6$$; and $$\Gamma_7$$ is open.
The two most commonly used methods for enforcing the boundary conditions are the collocation method and the Galerkin method. The collocation method works by enforcing the boundary conditions at \( N \) matching points

\[
\sum_{j=1}^{N} \sigma_j A_k[f_j](x_i) + \sum_{j=1}^{N} \mu_j B_k[f_j](x_i) = \gamma_k(x_i) \quad (14)
\]

where \( i = 1, 2, \ldots, N \) and \( k = 1, 2 \). Collocation methods have long been used \([22], [23]\). They are easy to understand, and the integral expressions necessary for implementing them have been derived by many authors. These include, e.g., piecewise constant and linear basis functions on triangular elements \([24]–[28]\), as well as higher order basis functions on curvilinear elements \([29]\). However, many of the boundary integrals are hypersingular, which make them hard (or sometimes even impossible) to compute, especially for points on the corners or edges of the boundary. Furthermore, the resulting system matrices are nonsymmetric.

The Galerkin method overcomes these problems by enforcing the boundary conditions in an integral sense. The boundary integral equation is multiplied by each of the same \( N \) basis functions and integrated over the boundary a second time

\[
\sum_{j=1}^{N} \int_{\Gamma} f_i A_k[f_j] dS(x) + \int_{\Gamma} f_i B_k[f_j] dS(x) = \int_{\Gamma} f_i \gamma_k dS(x) \quad (15)
\]

where \( i = 1, 2, \ldots, N \) and \( k = 1, 2 \). By doing so, all the hypersingular integrals become weakly singular. Moreover, the system matrices in the Galerkin method are typically symmetric, better conditioned, and have better convergence properties \([30], [31]\). However, the extra integral over the boundary complicates the computation of the entries in the system matrix.

### III. Double Surface Integrals

When the boundary is discretized using linear triangular elements, the double surface integrals are performed over pairs of these triangles. In each pair, one is called the “source” triangle, and the other the “receiver” triangle. The inside integral is over the source triangle \( S \) and the outside integral is over the receiver triangle \( R \) (see Fig. 2). Thus, when populating the system matrix, we need to compute integrals of the following form:

\[
I = \int_{x \in R} (\sigma_0 + p \cdot x) \int_{x' \in S} (\sigma_0' + p' \cdot x') \times F(x - x') dS(x') dS(x) \quad (16)
\]

where \( \sigma_0 + p \cdot x \) is the source density distribution over the source triangle, \( \sigma_0' + p' \cdot x' \) is the weight function over the receiver triangle, and \( F(r) \) is the kernel being integrated. To implement the Galerkin BEM described in Section II, we need

![Fig. 2. In practice, the relative geometry of two triangles is one of the following: (a) two triangles do not touch; (b) they share a vertex; (c) they share an edge; or (d) they are the same.](image-url)

to compute this integral for the following two kernels:

\[
F_1(x - x') = G(x - x') \quad (17)
\]

\[
F_2(x - x') = n' \cdot \nabla_{x'} G(x - x') \quad (18)
\]

where \( F_1(r) \) and \( F_2(r) \) correspond to the single- and double-layer potentials, respectively.

Computing the integral for these two kernels for all commonly encountered geometries is the focus of this paper. In practice, the relative geometry of the two triangles is one of the following: (a) the two triangles do not touch; (b) they share a vertex; (c) they share an edge; or (d) they are the same (see Fig. 2). These are called the zero-, one-, two-, and three-touch cases, respectively. In this naming scheme, the number represents how many vertices the two triangles share.

### IV. Zero-Touch Case

In the zero-touch case, because the two triangles do not touch, the double surface integral is regular and can be computed via standard numerical or semianalytical means. However, in this section, we present an analytical method for computing the integral in this case. This method was inspired by the fast multipole method and uses spherical harmonics and multipole and local expansions and translations. Similar methods were presented in \([32] \text{ and } [33]\) as part of fast multipole-accelerated solvers for problems in elastostatics. However, we build on the methods described in these references by: 1) adapting them to the kernels considered in Sections II and III; 2) computing the multipole expansion coefficients for a triangle exactly using Gaussian quadrature; and 3) controlling the error by adaptively truncating the multipole expansions and/or subdividing the problem when necessary.

#### A. Spherical Harmonics

The Green’s function for the Laplace equation can be expanded as

\[
G(x - x') = \frac{1}{4\pi |x - x'|}
\]

\[
= \sum_{n=0}^{\infty} \sum_{m=-n}^{n} \frac{1}{2n+1} \frac{r^n}{r+1} Y^n_m(\theta', \phi') Y^m_n(\theta, \phi) \quad (19)
\]

where \( r', \theta', \phi' \) and \( r, \theta, \phi \) are the spherical coordinates of \( x' \) and \( x \), respectively, \( r_\angle = \min(r', r) \), and \( r_\angle = \max(r', r) \) (see Fig. 3). The spherical harmonics are given by \([34]\).
expansion

| x potential at expansions. For example, suppose we want to compute the contained entirely inside an imaginary sphere of radius r centered around x.

Substituting (24) and rearranging

\[ \sum_{n=0}^{\infty} \sum_{m=-n}^{n} a_n R_n^m (x - x^*) = \int_{|x' - x^*| < r} \rho (x') R_n^m (x' - x^*) dV (x') \]

where \( P_n^m \) are the associated Legendre polynomials.

This expression can be used to build multipole and local expansions. For example, suppose we want to compute the potential at x due to a point source at x'. When r > r', we can build a multipole expansion

\[ \frac{1}{4 \pi |x - x'|} = \sum_{n=0}^{\infty} \sum_{m=-n}^{n} P_n^m (r', \theta', \phi') S_n^m (r, \theta, \phi) . \]

Here

\[ R_n^m (x) = R_n^m (r, \theta, \phi) = \left( \frac{1}{2n+1} \right)^{1/2} r^n Y_n^m (\theta, \phi) \]

\[ S_n^m (x) = S_n^m (r, \theta, \phi) = \left( \frac{1}{2n+1} \right)^{1/2} \frac{1}{r^{n+1}} Y_n^m (\theta, \phi) \]

are the local and multipole expansion basis functions, respectively. Instead of centering the expansion around the origin, we can center the expansion around x'. When |x - x'| > |x' - x*|

\[ \frac{1}{4 \pi |x - x'|} = \sum_{n=0}^{\infty} \sum_{m=-n}^{n} R_n^m (x' - x*) S_n^m (x - x*) . \]

Likewise, when |x - x*| < |x' - x*|, we can build a local expansion

\[ \frac{1}{4 \pi |x - x'|} = \sum_{n=0}^{\infty} \sum_{m=-n}^{n} S_n^m (x' - x*') R_n^m (x - x*) . \]

These expressions can be used to build multipole and local expansions for arbitrary source distributions. For example, let us build a multipole expansion for the source distribution \( \rho (x') \) contained entirely inside an imaginary sphere of radius r centered around x*. For a point x outside the sphere, the potential due to this source distribution is given by

\[ \Phi (x) = \int_{|x' - x^*| < r} \frac{\rho (x')}{|x - x'|} dV (x') \]

Substituting (24) and rearranging

\[ \Phi (x) = \sum_{n=0}^{\infty} \sum_{m=-n}^{n} a_n S_n^m (x - x^*) \]

We can build a local expansion using the same procedure. Consider a different source distribution contained entirely outside the imaginary sphere. For a point x inside the sphere, the potential due to this source distribution is given by

\[ \Phi (x) = \sum_{n=0}^{\infty} \sum_{m=-n}^{n} a_n^m R_n^m (x - x^*) \]

B. Analytical Method

We want to compute the following double surface integral over a source triangle S and a receiver triangle R:

\[ I = \int_{x \in \mathbb{R}} (\sigma_0 + p \cdot x) \int_{x' \in S} (\sigma_0' + p' \cdot x') \]

\[ \times G (x - x') dS (x') dS (x) . \]

This section gives an analytical method for computing this integral. A visual representation of steps in the method can be seen in Fig. 4.

First, expand the Green’s function as a multipole expansion

\[ I = \int_{x \in \mathbb{R}} (\sigma_0 + p \cdot x) \int_{x' \in S} (\sigma_0' + p' \cdot x') \]

\[ \times \sum_{n=0}^{\infty} \sum_{m=-n}^{n} R_n^m (x' - x^*) S_n^m (x - x^*) \]

\[ \times dS (x') dS (x) \]

where the expansion center x* is near the source triangle. Ideally, x* should be chosen so that the sphere centered around x* and completely containing the source triangle can be made as small as possible. There are actually only four possible choices of x*: the midpoints of the three edges of the source triangle and the center of the source triangle’s circumsphere. The one corresponding to the smallest sphere that completely contains the source triangle is chosen.

Second, rearrange (32) by moving the double sum and the \( S_n^{m'} \) outside the inside integral

\[ I = \int_{x \in \mathbb{R}} (\sigma_0 + p \cdot x) \sum_{n=0}^{\infty} \sum_{m=-n}^{n} \]

\[ \int_{x' \in S} (\sigma_0' + p' \cdot x') R_n^m (x' - x^*) S_n^{m'} (x' - x^*) \]

\[ \times dS (x') dS (x) . \]
The integral over the source triangle computes the expansion coefficients for the multipole expansion that represents the potential due to the linear source distribution over the source triangle

$$I = \int_{x \in \mathbb{R}} (\sigma_0 + p \cdot x) \sum_{n=0}^{\infty} \sum_{m=-n}^{n} a_{n,m}^{m'} S_{n,m'} (x - x^*) \, dS (x) \quad (34)$$

$$a_{n,m}^{m'} = \int_{x^* \in S} (\sigma_0^* + p' \cdot x^*) R_{n,m}^{m'} (x^* - x) \, dS (x^*). \quad (35)$$

Analytical expressions for computing the integral in (35) are available, e.g., in [35] and [36]. In [37], a recursive algorithm for computing the expansion coefficients was presented: only $a_{n,m}^{m'}$ for lower order and degree need to be computed explicitly; the others can be computed recursively from them. However, we make the following observation: $R_{n,m}^{m'}$ is polynomial, so the integrand in (35) is polynomial. Thus, the integral can be computed exactly via Gaussian quadrature. A similar approach was used in [38] and [39], although the integration domains in these references were lines and boxes, not triangles. We use the techniques given in [40] for performing Gaussian quadrature over the triangles.

Third, translate the multipole expansion centered around $x^*$ to a local expansion centered around $y^*$, where $y^*$ is near the receiver triangle

$$I = \int_{x \in \mathbb{R}} (\sigma_0 + p \cdot x) \sum_{n=0}^{\infty} \sum_{m=-n}^{n} b_{n,m} R_{n,m} (x - y^*) \, dS (x). \quad (36)$$

Ideally, $y^*$ should be chosen so that the sphere centered around $y^*$ and completely containing the receiver triangle can be made as small as possible. The procedure for choosing $y^*$ is the same as for choosing $x^*$. The local expansion coefficients $b_{n,m}$ are computed from the multipole expansion coefficients $a_{n,m}^{m'}$ via a multipole-to-local translation. There are several translation methods available, but to save space, they have been omitted from this paper. A good overview of them can be found online in [34]. For example, one popular method is the point and shoot method, whose computational cost scales as $p^3$.

Fourth, rearrange (36) by moving the double sum and the expansion coefficients outside the integral

$$I = \sum_{n=0}^{\infty} \sum_{m=-n}^{n} b_{n,m} c_{n,m} \quad (37)$$

$$c_{n,m} = \int_{x \in \mathbb{R}} (\sigma_0 + p \cdot x) R_{n,m} (x - y^*) \, dS (x). \quad (38)$$

Like before, since $R_{n,m}$ is polynomial, the integral in (38) can be computed exactly via Gaussian quadrature.

The above analysis assumes that the source distribution over the source triangle and the weight function over the receiver triangle are linear. Higher order basis functions, such as quadratic or cubic functions, could be used with very few changes. Indeed, the only change that must be made is to increase the order of the Gaussian quadrature to account for the increase in the degree of the integrand. The above analysis also assumes that the source and receiver surfaces are planar triangles. Arbitrary surfaces, e.g., curved elements, would be possible providing that there was an integration method for that surface. One possible method for doing this would be to transform the surfaces into triangles via a change in variables, and use the same Gaussian quadrature as before. Of course, the integrand of the transformed integral would now include the Jacobian associated with the change in variables. In this paper, however, only planar linear triangular elements are considered.

### C. Error Control

The expressions derived in Sections IV-A and IV-B involving multipole and local expansions must be truncated so that they can be implemented in code. For example, the last expression given in Section IV-B becomes

$$I \approx I_p = \sum_{n=0}^{P-1} \sum_{m=-n}^{n} b_{n,m} c_{n,m} \quad (39)$$

where only $p^2$ terms have been kept. Obviously, the expression is exact as $p \to \infty$, but there are truncation errors when $p < \infty$. These errors come from two sources: 1) the construction of the multipole expansion at the source triangle and 2) the translation of the multipole expansion to a local expansion at the receiver triangle. Luckily, these expansions converge geometrically, so these errors can be precisely controlled by picking an appropriate value of $p$.

Theoretical bounds for these errors as a function of $p$ have been derived by many authors over the years. A good overview is given in [41]. The relative error is bounded by

$$\varepsilon = \left| \frac{I_p - I}{I} \right| \leq A \eta^p \quad (40)$$

where the error constant $A$ depends on the problem being solved, and

$$\eta = \frac{\max (r_1, r_2)}{d - \min (r_1, r_2)} \quad (41)$$

where $r_1$ is the radius of the multipole expansion’s bounding sphere, $r_2$ is the radius of the local expansion’s bounding sphere, and $d = |y^* - x^*|$ (see Fig. 5). Using (40), we can easily pick a truncation number that gives us a desired accuracy

$$p = \left\lceil \log \left( \frac{\varepsilon}{A} \right) / \log (\eta) \right\rceil \quad (42)$$

where $\lceil x \rceil$ is the ceiling of $x$.

There are two issues. The number of terms in (39) grows as $p^2$, so the memory costs grow as $p^2$ as well. In addition, the computational costs grow as $p^3$ due to the multipole-to-local translation (assuming that the point and shoot method is used).
In the event that \( p \) becomes too large, we divide the larger triangle into four smaller triangles and recurse. We do the same in the event that the two bounding spheres overlap.

Second, while values of \( A \) have been derived for special cases, such as point sources, they have not been for the case of triangles. Instead of attempting to derive such a value analytically, we computed one experimentally. The experiment worked as follows. We generated 10,000 pairs of randomly placed triangles, where, in each pair, the two triangles did not touch. For each pair, we computed \( I \) using the analytical method for different values of \( \varepsilon \): \( 10^{-2}, 10^{-3}, 10^{-4}, \text{and } 10^{-5} \). Again, we used the value of \( I \) returned by the semianalytical method as the reference value. In Fig. 6, the actual errors (along the \( y \)-axis) for each pair of triangles (along the \( x \)-axis) are plotted for each choice of \( \varepsilon \) (the different colored curves/subplots). In all cases, the realized errors are below the desired error bounds.

The speed of the analytical method depends strongly on the truncation number. The number of expansion coefficients grows as \( p^2 \), and the computational cost of the analytical method grows as \( p^3 \). For pairs of triangles that are close to each other and require a large \( p \), the analytical method can be slower than semianalytical methods. However, for pairs of triangles that are far away from each other, \( p \) will be small, so the analytical method will be faster. In addition, the method can be used as part of a fast multipole-accelerated BEM. The method is used to compute the interactions between large groups of triangles instead of between pairs of individual triangles. In this case, the analytical method has great value and can provide tremendous speedups, as was seen in the references given at the beginning of Section IV.

V. ONE-, TWO-, AND THREE-TOUCH CASES

In the one-, two-, and three-touch cases, because the two triangles touch, the double surface integrals are not regular, so they cannot be computed using standard numerical or semianalytical means, or even the analytical method described in Section IV-B. In this section, we present a novel method for dealing with these integrals. The approach relies on several scaling properties of the integrals and the kernels being integrated. The method works in the following way: 1) the integral is broken up into several smaller integrals; 2) some of these integrals are related back to the original integral via scaling and symmetry arguments; and 3) the terms are rearranged to yield an expression for the original integral that only requires computing regular integrals explicitly (all other integrals are computed implicitly).

A. Preliminaries: Scaling Results

We want to compute the following double surface integral over a source triangle \( S \) and a receiver triangle \( R \):

\[
I = \int_{x \in R} \sigma_0 + p \cdot x \int_{x' \in S} (\sigma_0' + p' \cdot x') \\
\times F(x - x') dS(x') dS(x)
\]

where \( F(r) \) is any kernel that has the following scaling property:

\[
F(\alpha r) = s(\alpha) F(r).
\]

To begin, let us break the integral into four smaller integrals by expanding the product of the two linear functions

\[
I = \int_{x \in R} \int_{x' \in S} \times (\sigma_0 \sigma_0' + \sigma_0 (p' \cdot x') + (p \cdot x) \sigma_0' + (p \cdot x) (p' \cdot x')) \\
\times F(x - x') dS(x') dS(x)
\]

\[
I = \sigma_0 \sigma_0' I^1 + \sigma_0 I P' + \sigma_0' I P + I P P
\]

where

\[
I^1 = \int_{x \in R} \int_{x' \in S} F(x - x') dS(x') dS(x)
\]
\[ y = \alpha (x' + t) \]
\[ y = \alpha (x + t) \]

![Diagram showing the process for transforming the pair of triangles S and R into the pair of triangles S and R.](image)

Consider two different triangles \( S_\ast \) and \( R_\ast \), which, taken together, are scaled and translated versions of \( S \) and \( R \), also taken together (see Fig. 7). In other words, given a pair of points, \( x', x \in S \cup R \), there is a corresponding pair of points, \( y', y \in S \cup R \ast \), given by

\[ y' = \alpha (x' + t), \quad y = \alpha (x + t). \]

Suppose we want to compute the same integral as before, except over \( S_\ast \) and \( R_\ast \)

\[ I_\ast = \int_{y \in R_\ast} (\sigma_0 + p \cdot y) \int_{y' \in S_\ast} (\sigma_0' + p' \cdot y') \]
\[ \times F (y - y') dS (y') dS (y). \]

The integrand is exactly the same as before. The only thing we have changed is the integration domain from \( S \times R \) to \( S_\ast \times R_\ast \). Before, let us break the integral into four smaller integrals by expanding the product of the two linear functions

\[ I_\ast = \sigma_0 \sigma_0' I_1^1 + \sigma_0 p_0' I_1^p + \sigma_0' p_0 I_1^p + I_1^p \]

where \( I_1^1, I_1^p, I_1^p, \) and \( I_1^p \) are like their nonstarred versions from before, except for the different integration domain.

**Theorem 1:** The four integrals over \( S_\ast \) and \( R_\ast \), \( I_1^1, I_1^p, I_1^p, \) and \( I_1^p \), can be expressed in terms of the four integrals over \( S \) and \( R \), \( I_1^1, I_1^p, I_1^p, \) and \( I_1^p \)

\[ I_1^1 = s (\alpha) \alpha^4 I_1 \]
\[ I_1^p = s (\alpha) \alpha^5 \left( I_1^p + (p' \cdot t) I_1^1 \right) \]
\[ I_1^p = s (\alpha) \alpha^5 \left( I_1^p + (p \cdot t) I_1^1 \right) \]
\[ I_1^p = s (\alpha) \alpha^6 \times \left( I_1^p + (p \cdot t) I_1^p + (p' \cdot t) I_1^p + (p' \cdot t) (p' \cdot t) I_1^1 \right). \]

**Proof:** To prove (54)–(57) in Theorem 1: 1) make the change in variables in (51); 2) use the scaling property of \( F (r) \) in (44); and 3) break the resulting integral into one or more integrals that are equal to the original four integrals, \( I_1^1, I_1^p, I_1^p, \) and \( I_1^p \).

To prove (54)

\[ I_1^1 = \int_{y \in R_\ast} \int_{y' \in S_\ast} F (y - y') dS (y') dS (y) \]
\[ I_1^p = \int_{x \in R} \int_{x' \in S} (\alpha (x + t) - \alpha (x' + t)) \times dS (\alpha (x + t)) dS (\alpha (x' + t)) \]
\[ I_1^p = s (\alpha) \alpha^4 \int_{x \in R} \int_{x' \in S} F (x - x') dS (x') dS (x) \]
\[ I_1^p = s (\alpha) \alpha^4 I_1^1. \]

To prove (55)

\[ I_1^p = \int_{y \in R_\ast} \int_{y' \in S_\ast} F (y - y') dS (y') dS (y) \]
\[ I_1^p = s (\alpha) \alpha^5 \int_{x \in R} \int_{x' \in S} (p' \cdot (x' + t)) \times dS (x') dS (x) \]
\[ I_1^p = s (\alpha) \alpha^5 \left( I_1^p + (p' \cdot t) I_1^1 \right). \]

The analysis to prove (56) is the same as for (55).

To prove (57)

\[ I_1^p = \int_{y \in R_\ast} \int_{y' \in S_\ast} (p \cdot y) (p' \cdot y') \]
\[ \times F (y - y') dS (y') dS (y) \]
\[ I_1^p = s (\alpha) \alpha^6 \int_{x \in R} \int_{x' \in S} (p \cdot (x + t)) (p' \cdot (x' + t)) \times dS (x') dS (x) \]
\[ I_1^p = s (\alpha) \alpha^6 \times \left( I_1^p + (p \cdot t) I_1^p + (p' \cdot t) I_1^p + (p' \cdot t) (p' \cdot t) I_1^1 \right). \]

B. One-Touch Case

Consider the one-touch case in Fig. 8. Without loss of generality, assume the vertex that the two triangles share is located at the origin (i.e., \( p_1 = 0 \)). We divide each triangle into a triangle and a quadrilateral: the source triangle \( S \) is divided into \( 1' \) and \( 2' \), and the receiver triangle \( R \) is divided into 1 and 2. Note that \( 1' \) and 1 are similar to \( S \) and \( R \) (1/2 the size), respectively. We break the integral into three smaller integrals over these shapes.

Let us look at \( I^1 \)

\[ I^1 = I^1_{1'1} + I^1_{1'2} + I^1_{2'R}. \]
zero-touch integrals. Because we know how to compute them, let us combine them into a single integral

\[ I^1 = I^1_{1,1} + I^1_{\text{remainder}}. \]  

(69)

The integral \( I^1_{1,1} \), however, has the same problem as the original integral: \( 1' \) and \( 1 \) share a vertex. Fortunately, \( 1' \) and \( 1 \) are each scaled and translated versions of the original pair \( S \) and \( R \). We derived in Theorem 1 that \( I^1_1 = s(\alpha) \alpha^3 I^1 \). Since the pair of triangles \( 1' \) and \( 1 \) is \( 1/2 \), the size of the original pair \( S \) and \( R \) share, so

\[ I^1_{1,1} = s \left( \frac{1}{2} \right) \frac{1}{16} I^1. \]

(70)

Inserting this into (69) and rearranging

\[ I^1 = 1 - s \left( \frac{1}{2} \right) \frac{1}{16} I^1_{\text{remainder}}. \]

(71)

The integral \( I^1_{1,1} \) is computed implicitly in this expression.

Now, let us look at \( I^p \)

\[ I^p = I^p_{1,1} + I^p_{\text{remainder}} \]

(72)

where \( I^p_{\text{remainder}} = I^p_{1,2} + I^p_{2,1} \). We derived in Theorem 1 that

\[ I^p_s = s(\alpha) \alpha^5 \left( I^p + (p' \cdot t) I^1 \right). \]

(73)

Like before \( \alpha = 1/2 \), but we still need to determine \( t \). When \( \alpha = 1/2 \), \( t \) is the point that does not change during the scaling and translation. This is simply the vertex that the two triangles share, so \( t = p_1 \). Because we assumed that \( p_1 = 0 \), \( t = 0 \) as well, so

\[ I^p_{1,1} = s \left( \frac{1}{2} \right) \frac{1}{32} I^p. \]

(74)

Plugging this into (72) and rearranging

\[ I^p = 1 - s \left( \frac{1}{2} \right) \frac{1}{32} I^p_{\text{remainder}}. \]

(75)

These same procedures can be used to compute \( I^p \) and \( I^{p'} p \)

\[ I^p = 1 - s \left( \frac{1}{2} \right) \frac{1}{32} I^p_{\text{remainder}} \]

(76)

\[ I^{p'} p = 1 - s \left( \frac{1}{2} \right) \frac{1}{64} I^{p'} p_{\text{remainder}}. \]

(77)

C. Two-Touch Case

Consider the two-touch case in Fig. 9. Without loss of generality, assume the midpoint of the edge that the two triangles share is located at the origin (i.e., \( p_2 + p_3 = 0 \)). Like in the one-touch case, we solve this problem by breaking the integral into several smaller integrals.

Let us look at \( I^1 \)

\[ I^1 = I^1_{1,R} + I^1_{2,1} + I^1_{2,3} + I^1_{3,2} + I^1_{3,4} + I^1_{4,3} + I^1_{4,1} + I^1_{1,4} + I^1_{1,1} - I^1_{1,1}. \]

(78)

The integrals, \( I^1_{1,R}, I^1_{1,1}, \) and \( I^1_{1,1} \), are zero-touch integrals, and the integrals, \( I^1_{2,1}, I^1_{2,3}, I^1_{3,1}, I^1_{3,2}, I^1_{4,1}, I^1_{4,2}, I^1_{4,3}, \) and \( I^1_{4,4} \), are one-touch integrals. Since we know how to compute them, let us combine them into a single integral

\[ I^1 = I^1_{2,3} + I^1_{3,2} + I^1_{4,3} + I^1_{4,4} + I^1_{1,1}. \]

(79)

That leaves \( I^1_{2,3} \) and \( I^1_{3,2} \), which have the same problem as the original integral: they correspond to pairs of triangles that share an edge. Fortunately, the pair \( 2' \) and \( 3' \) and the pair \( 3' \) and \( 2 \) are each scaled and translated versions of the original pair \( S \) and \( R \). Using (54) from Theorem 1, we have

\[ I^1_{2,3} = s \left( \frac{1}{2} \right) \frac{1}{16} I^1, \quad I^1_{3,2} = s \left( \frac{1}{2} \right) \frac{1}{16} I^1. \]

(80)

Substituting these into (79) and rearranging

\[ I^1 = 1 - s \left( \frac{1}{2} \right) \frac{1}{16} I^1_{\text{remainder}}. \]

(81)

The integrals \( I^1_{2,3} \) and \( I^1_{3,2} \) are computed implicitly in this expression.

Now, let us look at \( I^{p'} \)

\[ I^{p'} = I^{p'}_{2,3} + I^{p'}_{3,2} + I^{p'}_{\text{remainder}} \]

(82)

where, again, the 10 integrals that we know how to compute have been combined into \( I^{p'}_{\text{remainder}} \). Using (55) from Theorem 1, we have

\[ I^{p'}_{2,3} + I^{p'}_{3,2} = s \left( \frac{1}{2} \right) \frac{1}{16} I^{p'} + s \left( \frac{1}{2} \right) \frac{1}{32} (p' \cdot (t_{2,3} + t_{3,2})) I^1. \]

(83)
We need to determine $t_{2/3}$ and $t_{y/2}$. As discussed earlier, when $\alpha = 1/2$, the translation vector $t$ corresponds to the point that does not change during the transformation. Thus, for the pair 2 and 3, that is $p_2$, and for the pair 2 and 3, that is $p_3$. The edge that $S$ and $R$ share is centered around the origin, so $t_{2/3} + t_{y/2} = 0$, which means that

$$I_{2/3}^p + I_{y/2}^p = s \left( \frac{1}{2} \right) \frac{1}{16} I_p^p. \tag{84}$$

Inserting this into (82) and rearranging

$$I_p^p = \left( 1 - s \left( \frac{1}{2} \right) \frac{1}{16} \right)^{-1} I_{\text{remainder}}^p. \tag{85}$$

The same analysis can be used for computing $I_p$

$$I_p = \left( 1 - s \left( \frac{1}{2} \right) \frac{1}{16} \right)^{-1} I_{\text{remainder}}. \tag{86}$$

Finally, let us look at $I_p^p$

$$I_p^p = I_{2/3}^p + I_{y/2}^p + I_{\text{remainder}}^p. \tag{87}$$

Using (57) from Theorem 1, we have

$$I_{2/3}^p + I_{y/2}^p = s \left( \frac{1}{2} \right) \frac{1}{32} I_p^p$$

$$+ s \left( \frac{1}{2} \right) \frac{1}{64} ((p \cdot t_{2/3}) (p' \cdot t_{2/3}) + (p \cdot t_{y/2}) (p' \cdot t_{y/2})) I_{1}. \tag{91}$$

The terms linear in $p'$ and $p$ disappear because $t_{2/3} + t_{y/2} = 0$, but the other terms remain. Plugging (88) into (87) and rearranging

$$I_p^p = \left( 1 - s \left( \frac{1}{2} \right) \frac{1}{32} \right)^{-1} I_{\text{remainder}}^p + a_p^p \tag{92}$$

where

$$a_p^p = s \left( \frac{1}{2} \right) \frac{1}{64} ((p \cdot t_{2/3}) (p' \cdot t_{2/3}) + (p \cdot t_{y/2}) (p' \cdot t_{y/2})) I_{1}. \tag{93}$$

**D. Three-Touch Case**

Consider the three-touch case in Fig. 10. In this case, the two triangles are the same. Without loss of generality, assume that the centroids of the two triangles are located at the origin (i.e., $p_1 + p_2 + p_3 = 0$). Like in the one- and two-touch cases, we solve this problem by breaking the integral into several smaller integrals.

Let us look at $I^1$

$$I^1 = \sum_{i=1}^{4} \sum_{j=1}^{4} I_{ij}^1. \tag{94}$$

The integrals, $I_{1/1}^1$, $I_{1/3}^1$, $I_{2/2}^1$, $I_{3/1}^1$, and $I_{3/2}^1$, are one-touch integrals, and the integrals, $I_{1/4}^1$, $I_{2/4}^1$, $I_{3/4}^1$, $I_{4/1}^1$, $I_{4/2}^1$, and $I_{4/3}^1$, are two-touch integrals. Like before, because we know how to compute them, let us combine these 12 integrals into a single integral

$$I^1 = I_{1/1}^1 + I_{1/2}^1 + I_{1/3}^1 + I_{1/4}^1 + I_{2/2}^1 + I_{3/3}^1 + I_{3/4}^1 + I_{4/4}^1. \tag{95}$$

That leaves the four integrals, $I_{1/1}^1$, $I_{1/2}^1$, $I_{3/3}^1$, and $I_{4/4}^1$, which have the same problem as the original: they correspond to pairs of triangles that are the same. Using (54) from Theorem 1, we have

$$I_{1/1}^1 + I_{1/2}^1 + I_{3/3}^1 = s \left( \frac{1}{2} \right) \frac{3}{16} I^1. \tag{96}$$

The pair of triangles 4' and 4 are not only scaled by a factor of 1/2 but also rotated by 180°. Because they are centered around the origin, we can achieve this rotation by setting $\alpha = 1/2$

$$I_{4/4}^1 = s \left( \frac{1}{2} \right) \frac{3}{16} I^1. \tag{97}$$

Substituting these into (92) and rearranging

$$I^1 = \left( 1 - s \left( \frac{1}{2} \right) \frac{3}{16} \right) - s \left( \frac{1}{2} \right) \frac{1}{16} \right)^{-1} I_{\text{remainder}}^1. \tag{98}$$

The same analysis from here and before can be used to compute expressions for $I_p^p$, $I_p$, and $I_{p}^p$

$$I_p^p = \left( 1 - s \left( \frac{1}{2} \right) \frac{3}{32} \right) + s \left( \frac{1}{2} \right) \frac{1}{32} \right)^{-1} I_{\text{remainder}}^p \tag{99}$$

where

$$a_{p}^p = s \left( \frac{1}{2} \right) \frac{1}{64} ((p \cdot t_{1/1}) (p' \cdot t_{1/1}) + (p \cdot t_{2/2}) (p' \cdot t_{2/2}) + (p \cdot t_{3/3}) (p' \cdot t_{3/3})) I^1. \tag{100}$$

and $t_{1/1} = p_1$, $t_{2/2} = p_2$, and $t_{3/3} = p_3$. 

Fig. 10. Three-touch case. A diagram showing how to compute the double surface integral for two triangles that are the same.
for that kernel. For the two kernels, $F_1 (r)$ (the Green’s function) and $F_2 (r)$ (the normal derivative of the Green’s function), needed by the Galerkin BEM and given in Section III, we have both of these things.

The scaling functions for the two kernels are $s_1 (\alpha) = |\alpha|^{-1}$ and $s_2 (\alpha) = \alpha |\alpha|^{-3}$.

In Section IV-B, we presented an analytical method for computing a zero-touch integral. However, the analytical method is better for pairs of triangles that are relatively far away from each other. In the present case, because the pairs of triangles are typically much closer together, we use a semianalytical method. In this method, the inside integral is computed analytically, and the outside integral is computed numerically via Gaussian quadrature. The use of the semianalytical method introduces error, but this error can be precisely controlled by choosing the number of quadrature points. Let $Q$ be this number. When computing one-, two-, and three-touch integrals, the only source of error comes from the zero-touch integrals, so $Q$ can be used to control the error in these cases as well. Each zero-touch integral requires a $Q$-point quadrature, so the computational complexity of the one-, two-, and three-touch integral methods scales as $O (ZQ)$, where $Z$ is the number of zero-touch integrals that need to be computed in each method.

G. Numerical Examples

We ran a series of computational experiments to validate the one-, two-, and three-touch integral computation methods described in Sections V-B, V-C, and V-D. There are two kernels that need to be integrated: 1) $F_1 (r)$, the Green’s function; and 2) $F_2 (r)$, the normal derivative of the Green’s function.

To validate that the methods work for $F_1 (r)$, we compared them to an exact method that was developed in [6] for computing the double surface integral in the three-touch case. We call this method the Eibert method. Because our three-touch method exercises our zero-, one-, and two-touch methods, this comparison provides a validation for all four methods.

The experiment worked as follows. We generated 10,000 randomly shaped triangles, and computed the three-touch integral for each using the Eibert method. Then, we computed the same integrals using our method with varying degrees of accuracy. As discussed in the previous section, the accuracy is governed by $Q$, which is the number of quadrature points used during the Gaussian quadrature in the zero-touch case. We varied the accuracy from low ($Q = 1$) to high ($Q = 1024$). After doing so, the triangles were divided up into four groups based on the largest interior angle $\theta$: 1) acute (nearly equilateral) triangles ($\theta \leq 75^\circ$); 2) right triangles ($75^\circ \leq \theta \leq 90^\circ$); 3) slightly obtuse triangles ($90^\circ \leq \theta \leq 120^\circ$); and 4) very obtuse triangles ($\theta \geq 120^\circ$). For each group, we plotted the maximum relative error in that group as a function of $Q$ (see Fig. 12). In general, all four types of triangles become more accurate as $Q$ increases. However, acute triangles performed the best, requiring only 144 quadrature points at the lowest level to achieve a maximum relative error of less than $10^{-10}$. For a good BEM mesh, we expect most triangles to be acute.

Next, let us validate that the methods work for $F_2 (r)$. To do so, we implemented the Galerkin BEM described in Section II
using linear elements, and solved four example BVP problems. The four problems were exactly the same, except for the boundaries used. The boundaries were: a tetrahedron, an octahedron, a cube, and an icosahedron (see Fig. 13). In each problem, the potential \( \phi \) was set to zero outside the boundary. Inside the boundary, the potential was set to \( \phi(x) = a + b \cdot x \), where, in these examples, \( a = 1 \) and \( b = (0.5, 0.8, -0.7) \). The exact solution to this problem can be derived using the jump conditions given in Section II. In fact, the single- and double-layer source distributions are piecewise linear along the boundaries, meaning that linear elements can provide an exact solution.

The experiment worked as follows. We computed the solution to each of the four problems using the indirect BEM. The integrals for when the kernel was \( F_1(r) \) were computed as accurately as possible (i.e., \( Q = 1024 \)). This left the integrals for when the kernel was \( F_2(r) \) as the primary source of error. We varied the accuracy at which these integrals were computed from low (\( Q = 1 \)) to high (\( Q = 1024 \)), and computed the maximum relative error of the solution for each choice of \( Q \). Fig. 14 shows the maximum relative error of the solution as a function of \( Q \) for each of the four different boundaries. As the accuracy of the integrals was increased, the total accuracy of the problem increased as well. For \( Q > 200 \), the maximum relative error drops to around \( 10^{-10} \) for all four geometries.

VI. CONCLUSION

We have presented a method for computing the double surface integrals encountered in the Galerkin BEM. When the boundary is discretized using triangular elements, these integrals are performed over pairs of these triangles. They can be extremely tough to compute, especially when the two triangles share a vertex, an edge, or are the same. This is because the kernels being integrated are often singular along the corners and edges of these triangles. We have solved this problem by using several scaling properties of the integrals and the kernels being integrated. The integral is broken up into several smaller ones, some of which are written in terms of the original. This is done in such a way that only completely regular integrals have to be computed explicitly. We have also presented an analytical method for computing the integrals when the two triangles do not touch. The method uses spherical harmonics and multipole and local expansions and translations. The only source of error in this method is how soon to truncate these expansions. However, the truncation number is adaptively selected to achieve a desired error bound. Finally, we have validated both of these methods, and have shown that they are accurate.

REFERENCES


