# Optimization of Fast Algorithms for Global Quadrature by Expansion Using Target-Specific Expansions

Matt Wala*[1] and Andreas Klöckner†[1]

[1]Department of Computer Science, University of Illinois at Urbana-Champaign

November 14, 2019

**Abstract**

We develop an algorithm for the asymptotically fast evaluation of layer potentials close to and on the source geometry, combining Geometric Global Accelerated QBX ('GIGAQBX') and target-specific expansions. GIGAQBX is a fast high-order scheme for evaluation of layer potentials based on Quadrature by Expansion ('QBX') using local expansions formed via the Fast Multipole Method (FMM). Target-specific expansions serve to lower the cost of the formation and evaluation of QBX local expansions, reducing the associated computational effort from $O((p+1)^2)$ to $O(p+1)$ in three dimensions, without any accuracy loss compared with conventional expansions, but with the loss of source/target separation in the expansion coefficients. GIGAQBX is a 'global' QBX scheme, meaning that the potential is mediated entirely through expansions for points close to or on the boundary. In our scheme, this single global expansion is decomposed into two parts that are evaluated separately: one part incorporating near-field contributions using target-specific expansions, and one part using conventional spherical harmonic expansions of far-field contributions, noting that convergence guarantees only exist for the sum of the two sub-expansions. By contrast, target-specific expansions were originally introduced as an acceleration mechanism for 'local' QBX schemes, in which the far-field does not contribute to the QBX expansion. Compared with the unmodified GIGAQBX algorithm, we show through a reproducible, time-calibrated cost model that the combined scheme yields a considerable cost reduction for the near-field evaluation part of the computation. We support the effectiveness of our scheme through numerical results demonstrating performance improvements for Laplace and Helmholtz kernels.

## 1 Introduction

The numerical realization of integral equation methods for the solution of boundary value problems of elliptic partial differential equations presents a number of technical challenges. Chief among them is the *accurate* and *rapid* evaluation of *layer potentials*, such as the single-layer potential

$$(\mathcal{S}\mu)(x) \coloneqq \int_\Gamma \mathcal{K}(x,y)\mu(y)\,dS(y), \tag{1}$$

an integral defined over a surface $\Gamma$, where $\mathcal{K}$ is a free-space Green's function for the underlying PDE, and $\mu : \Gamma \to \mathbb{C}$ is a surface density function. What makes this task challenging is a combination of requirements for a quadrature scheme, including the ability to handle singularities and near-singularities, complex geometries $\Gamma$, on-surface and near-surface evaluation, and support for simultaneous evaluation at a large number of target points with low algorithmic complexity.

The potential that results from discretizing (1) with a smooth high-order quadrature rule is a *point potential* of the form

$$\Pi(x_i) = \sum_{j=1}^{N} w_j \mathcal{K}(x_i, y_j) \quad (i = 1, \ldots, M), \tag{2}$$

*wala1@illinois.edu

†andreask@illinois.edu

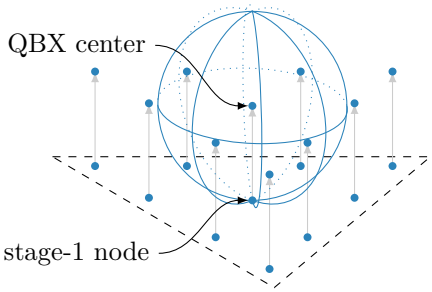


Figure 1: Depiction of QBX center placement for a discretization over a triangular element. The QBX centers for on-surface evaluation are spawned in the normal direction at the discretization nodes termed 'stage-1' nodes in [32], which serve as targets for on-surface evaluation. See also Section 2.5.

where $\{x_i\}_{i=1}^M \subseteq \mathbb{R}^3$ is a set of targets, $\{y_i\}_{i=1}^N \subseteq \Gamma$ a set of sources, and $\{w_i\}_{i=1}^N \subseteq \mathbb{R}$ is a set of source weights. While $\Pi$ is a generally accurate approximation of $\mathcal{S}\mu$ for target points away from the source geometry $\Gamma$, accuracy substantially decreases in the region close to $\Gamma$ or on $\Gamma$ itself. A recent approach to resolving this problem is based on Quadrature by Expansion (QBX, [21]). By using the broadly applicable assumption that the underlying kernel is a (locally) analytic function of the target point $x$, and leveraging the strengths of smooth high-order quadrature rules for one- and two-dimensional functions, QBX achieves high-order accuracy without sacrificing generality. The key idea behind QBX is that, to extend the applicability of smooth high-order quadrature to all points $x \in \mathbb{R}^3$, the potential (2) can be expanded in a local expansion about a center $c \in \mathbb{R}^3 \setminus \Gamma$, and this expansion may recover the value of the potential by analytic continuation in regions where smooth quadrature is not adequate. A potential of the form (2) can be evaluated in $O(N+M)$ time by way of the Fast Multipole Method (FMM, [5]). Furthermore, as a by-product of evaluation, the FMM forms local expansions of the point potential covering the entire computational domain. Since QBX expansions are precisely local expansions of (2) at appropriately chosen centers, the FMM therefore provides a suitable path towards an 'acceleration' strategy for reducing the algorithmic complexity of QBX, by translating a suitably-chosen local expansion to the QBX center. Care must be taken to ensure accurate combination of this far-field potential with a suitable QBX-mediated near-field contribution.

The recent contribution [32] combines a version of QBX, termed 'global' QBX, with the FMM for the evaluation of layer potentials in three dimensions. This scheme, termed 'Geometric Global Accelerated QBX', or 'GIGAQBX', in [33, 32], carefully controls the error introduced by the FMM acceleration by enforcing strict geometric separation criteria between intermediate multipole or local expansions and QBX local expansions. As a consequence of these separation criteria, the size of the near-field of a QBX center increases if compared with a scenario where the QBX center is treated as a particle in a 'point'-based FMM. Empirically, the dominant cost of the scheme typically appears to come from computing the spherical harmonic coefficients of QBX expansions from source points in the near-field. A second important cost is the conversion of near-field multipole expansions to QBX expansions. The primary purpose of this paper is to reduce these two costs.

Siegel and Tornberg [29] recently proposed using *target-specific* expansions to reduce the cost of forming QBX expansions. Target-specific expansions are based on leveraging information about both the source and the target to reduce the number of terms in the expansion. Specifically, for expansions in spherical harmonics, they are based on rewriting the expression for a local expansion using the addition theorem for Legendre polynomials, an analytical tool previously applied to QBX in the context of quadrature estimates [18]. The number of terms in the expansion is reduced in the polynomial order from $(p+1)^2$ to $p+1$, with a corresponding reduction in computational effort, where $p$ is the expansion order. Furthermore, the use of these expansions incurs no additional error because they are based on a mathematical identity. The expansions are termed 'target-specific' because they do not separate the influence of the source and target in the way that spherical harmonic expansions do, which means that a different set of expansion 'coefficients' is needed for each source-target pair. Because of the need to recompute the coefficients for each target, target-specific expansions cannot generally be used as a replacement for spherical harmonic expansions in an FMM. (An exception to this is Anderson's FMM [2], in which the expansion of potentials is based on the Poisson integral formula, which is evaluated as a target-specific expansion.)

Siegel and Tornberg introduce the numerical use of target-specific expansions in the context of a scheme termed 'local' QBX. The primary difference between 'global' QBX (such as the scheme in this paper) and 'local' QBX is that, in global QBX, the entire potential is conveyed to a target by a QBX expansion (hence 'global' QBX), whereas in local QBX only the potential due to sources in a neighborhood of the target comes through the QBX expansion, which is more akin to conventional FMM-accelerated quadrature schemes employing local correction. Compared with global QBX, local QBX features relatively straightforward integration with the FMM due to relying only on 'point' evaluations for the accelerated part. A second advantage of local QBX is that the placement of expansion centers is subject to less stringent geometric requirements, leading to higher efficiency in areas where the geometry is close to touching or highly irregular. Despite these advantages, for achieving the same level of accuracy as global QBX, current versions of local QBX appear to require higher quadrature order and oversampling in the local neighborhood of a target. This appears to be due to a form of truncation error not present in global QBX—specifically, error introduced when matching the transition between the QBX-mediated local neighborhood and the 'point' field from far-away sources.

The main observation in this paper is that target-specific expansions, originally developed for local QBX and used a post-processing step in combination with a 'point' FMM, can also be used within the context of global QBX, and, more specifically, within the GIGAQBX FMM. An approximation to the QBX expansion $\mathsf{L}_c^{\mathrm{qbx}}(t)$ in the GIGAQBX FMM at a target $t \in \mathbb{R}^3$ associated with an expansion center $c \in \mathbb{R}^3$ is formed from three parts (see Section 3.3 for notation)

$$\mathsf{L}_c^{\mathrm{qbx}}(t) = \mathsf{L}_c^{\mathrm{qbx,far}}(t) + \mathsf{L}_c^{\mathrm{qbx},W}(t) + \mathsf{L}_c^{\mathrm{qbx,near}}(t), \tag{3}$$

where the portion $\mathsf{L}_c^{\mathrm{qbx,near}}$ is mediated by direct formation of the expansion due to the sources in the near-field of the box 'owning' the center $c$, and the quantities $\mathsf{L}_c^{\mathrm{qbx,far}}$ and $\mathsf{L}_c^{\mathrm{qbx},W}$ are obtained through the various expansion translations present in the Fast Multipole Method. Unlike local QBX, analytical truncation error estimates for global QBX only apply to the combined expansion $\mathsf{L}_c^{\mathrm{qbx}}$ and not the sub-expansions. Thus, while no truncation bounds are known for the individual terms in (3), truncation bounds do hold on the overall sum, and the partitioning of the potential into sub-expansions has no effect on truncation error. In the context of the present work (and in keeping with the cost argument above) we use target-specific expansions to evaluate the component $\mathsf{L}_c^{\mathrm{qbx,near}}$.

The observation that a component of the FMM field can be mediated with target-specific expansions applies conceptually to not only the GIGAQBX variant of global QBX but to others as well such as the ones in [26, 27], or three-dimensional versions thereof. It is however likely that target-specific expansions will have greater cost impact on GIGAQBX due to the larger proportion of direct interactions.

An especially important use case, and one for which target-specific expansions can excel in cost compared with ('target-independent') spherical harmonic expansions, is *on-surface evaluation*. In our current treatment of on-surface evaluation, each on-surface target uses a different QBX center (Figure 1), meaning that no QBX expansion is typically evaluated at more than one target. As such, there is no advantage to be had from the source-target separation of variables in spherical harmonic expansions, and when there is only one target per center, forming and evaluating spherical harmonic expansions is more expensive than target-specific expansions.

A second observation we make in this paper is that, while replacing near-field evaluations with target-specific expansions results in cost improvements, adjusting parameters to the algorithm expectedly leads to further opportunities for cost reduction. By design, the FMM is tasked with making choices between whether to evaluate an interaction directly or mediate it via expansions. Assuming one is presented with a set of evaluation strategies which meet required accuracy tolerances, a standard method for minimizing computational cost is to use thresholds based on particle counts to decide which strategy to use. Because target-specific expansions make direct evaluations less expensive, adjusting the thresholds in such a way as to shift a larger fraction of the work onto direct interactions reduces the overall cost of the FMM. To aid in this *rebalancing* of the FMM, in this paper we develop a cost model that models the number of floating point operations in the GIGAQBX FMM. With appropriate fitting of empirical per-stage calibration factors, this model is able to approximate the total computational time used by the algorithm with high accuracy. This in turn provides a reproducible cost measure, used here for wall-time independent balancing and the reporting of cost and scalability results.

Much analytical modeling work for optimizing for FMM cost is restricted to the case of uniform distributions (e.g. [7, 15, 30, 35]), which makes it inapplicable to layer potential evaluation, since the particle distributions arising from surface discretizations for layer potentials are not uniform in the volume. Our work differs from the approaches for uniform distributions by making use of more information from the geometry in order to give a precise prediction of cost. A similar approach to FMM cost modeling based on approximating the number of floating point operations through direct inspection of the geometry is used in [2, 16], though the details of the modeled algorithms differ substantially from GIGAQBX. Other work on cost models for nonuniform particle distributions includes the contributions [25, 1, 23]. In [25], a model is developed to optimize for FMM parameters when the particle distribution is a fractal set. In [1], an empirical model is developed for predicting the cost of a particular task-based FMM implementation on arbitrary particle distributions. The contribution [23] discusses general conditions on the distribution under which linear scaling may be expected. In addition to optimization for cost, models have been applied to aid in solving the problem of distributing work among heterogeneous systems [8] or predicting execution characteristics taking into account both computation and memory accesses [6].

In summary, we present the following contributions in this paper.

- We describe how to use target-specific expansions (TSQBX) inside the global QBX (GIGAQBX) FMM, a technique previously only used in 'local' QBX, to reduce the cost of the near-neighborhood interactions.
- We present a cost model for the GIGAQBX algorithm on a shared memory system, which predicts the running time of the GIGAQBX FMM with very high accuracy.
- We demonstrate a 1.7–3.3× reduction in modeled cost using TSQBX on test cases with complex unstructured geometries for the Laplace and Helmholtz equations.

The organization of this paper is as follows. Section 2 describes background material pertaining to QBX and the GIGAQBX FMM. In Section 3, we present for completeness a full statement of the GIGAQBX algorithm using target-specific expansions. In Section 4, we present a study pertaining to the cost impact of target-specific expansions within the GIGAQBX FMM, making concluding remarks in Section 5. Appendix A presents a derivation of target-specific expansions for various kernels. Appendix B describes how to obtain the software used in this paper.

# 2 Background

## 2.1 Layer Potentials

For the sake of exposition, we consider the solution of the exterior Neumann problem in three dimensions, for a smooth, bounded, simply or multiply-connected domain $\Omega$ with boundary $\Gamma$. For continuous Neumann data $g$, the boundary value problem is to find a function $u : \mathbb{R}^3 \setminus \Omega \to \mathbb{R}$ such that

$$\begin{aligned} \triangle u(x) &= 0 \qquad x \in \mathbb{R}^3 \setminus \Omega, \\ \lim_{h \to 0^+} \hat{\nu}(x) \cdot \nabla u(x + h\hat{\nu}(x)) &= g(x) \quad x \in \Gamma, \\ \lim_{|x| \to \infty} u(x) &= 0. \end{aligned} \tag{4}$$

The notation $\hat{\nu}(x)$ refers to the outward-facing unit normal vector at $x$. The method under consideration here lends itself to the solution of a considerably broader family of boundary value problems.

We represent the solution of the problem by means of layer potentials. In the remainder of this paper, we will use $|\cdot|$ to denote the Euclidean ($\ell^2$) norm unless otherwise specified. Recalling the Green's function for the Laplace equation,

$$\mathcal{G}(x, y) = (4\pi)^{-1} |x - y|^{-1}, \tag{5}$$

we consider in this section the single-layer potential (1) $\mathcal{S}\mu$ with kernel $\mathcal{K} = \mathcal{G}$ and we introduce the following layer potential $\mathcal{S}'\sigma$ with density function $\sigma : \Gamma \to \mathbb{R}$:

$$(\mathcal{S}'\sigma)(x) \coloneqq \int_\Gamma \frac{\partial \mathcal{G}(x, y)}{\partial \hat{\nu}(x)} \sigma(y) \, dS(y).$$

With the aid of these operators, we represent the solution $u$ as

$$u \coloneqq \mathcal{S}\mu$$

using an unknown density $\mu : \Gamma \to \mathbb{R}$. By differentiation under the integral sign, any function of the form $\mathcal{S}\mu$ satisfies the Laplace PDE in the exterior of $\Omega$, and by considering the asymptotic behavior of the Green's function $\mathcal{G}$, it also automatically satisfies the far-field decay conditions.

We enforce the Neumann boundary conditions on this representation as follows. The classical *jump relations* [22, Thm. 6.19] imply that the normal derivative of $\mathcal{S}\mu$ is discontinuous across the boundary, in the sense that

$$\lim_{h \to 0^{\pm}} \hat{\nu}(x) \cdot \nabla u(x + h\hat{\nu}(x)) = \mathcal{S}'\mu(x) \mp \frac{1}{2}\mu(x) \quad (x \in \Gamma).$$

Thus, to satisfy the boundary condition, $\mu$ needs to solve the following second kind integral equation:

$$g = \left(\mathcal{S}' - \frac{1}{2}\right)\mu. \tag{6}$$

With the help of the Fredholm theory for second kind integral equations one obtains that the solution to this equation exists, is unique, and is continuously dependent on $g$ [22, Thm. 6.28, 6.30]. Under a suitable discretization, this equation provides the basis for numerical methods for solution of the exterior Neumann problem.

## 2.2 High-Order Quadrature for Smooth Functions

The primary concern of this paper is the numerical evaluation of layer potentials such as the single layer potential (1) anywhere in $\mathbb{R}^3$, including near or on the surface $\Gamma$. A natural but ultimately deficient approach to this problem uses smooth composite quadrature. In this approach, the surface is assumed to be tessellated into $K$ disjoint surface elements

$$\Gamma = \bigcup_{k=1}^{K} \Gamma_k.$$

Each element $\Gamma_k$ is parametrized by a smooth mapping function $\Psi_k : E \to \mathbb{R}^3$ from a reference element $E$ in the plane. Then, with the use of an $M$-point quadrature rule defined over the reference element with weights $\{w_i\}_{i=1}^{M}$ and nodes $\{y_i\}_{i=1}^{M}$, the single layer potential admits the approximation

$$\begin{aligned}(\mathcal{S}\mu)(x) &= \sum_{k=1}^{K} \iint_E \mu(\Psi_k(y)) \cdot \mathcal{G}(x, \Psi_k(y)) \cdot \left|(\partial_{e_1}\Psi_k \times \partial_{e_2}\Psi_k)(y)\right| \, dS(y) \\ &\approx \sum_{k=1}^{K} \sum_{i=1}^{M} w_i \cdot \mu(\Psi_k(y_i)) \cdot \mathcal{G}(x, \Psi_k(y_i)) \cdot \left|(\partial_{e_1}\Psi_k \times \partial_{e_2}\Psi_k)(y_i)\right|.\end{aligned}$$

For points $x$ far from the surface $\Gamma$, the integrand is a smooth function favorable to numerical treatment with conventional quadrature rules for smooth functions on plane regions. However, for $x$ near to the surface, the quadrature error is at worst unbounded due to near-singularity of the Green's function under the integrand, which necessitates a massive increase in quadrature order to resolve. This effectively prevents the practical use of traditional high-order composite quadrature for these integrands as an evaluation strategy in a neighborhood of the surface, where the size of the neighborhood depends on the size of the element, the quadrature order, and the desired accuracy [21].

The key insight in QBX as a quadrature scheme is that $\mathcal{S}\mu$ is an analytic function on $\mathbb{R}^3 \setminus \Gamma$. This fact can be exploited to recover quadrature accuracy for $x$ near the surface via analytic continuation, at a lower cost than smooth quadrature.

## 2.3 Expansion of Potentials in Spherical Harmonics

We recall the addition theorem for the Laplace potential in three dimensions. Let $a, b \in \mathbb{R}^3$ with $0 < |a| < |b|$, and let $\gamma_{ab}$ be the angle (about the origin) between $a$ and $b$. By expanding $\mathcal{G}$ in a binomial series [17], the potential can be represented as

$$\mathcal{G}(a,b) = \frac{1}{4\pi} \sum_{n=0}^{\infty} \frac{|a|^n}{|b|^{n+1}} P_n(\cos\gamma_{ab}). \tag{7}$$

The function $P_n$ is the Legendre polynomial of degree $n$. The term $P_n(\cos\gamma_{ab})$ may be further expanded in a series of spherical harmonics. Let $a$ and $b$ be written in polar and azimuthal spherical coordinates $(\theta_a, \phi_a)$ and $(\theta_b, \phi_b)$ respectively, i.e. with $\theta = \cos^{-1}(z/r), \phi = \operatorname{atan2}(y,x)$. The identity known as the spherical harmonic addition theorem, or the addition theorem for Legendre polynomials, says that [29]

$$P_n(\cos\gamma_{ab}) = \frac{4\pi}{2n+1} \sum_{m=-n}^{n} Y_n^m(\theta_a,\phi_a) Y_n^{-m}(\theta_b,\phi_b). \tag{8}$$

The spherical harmonics $Y_n^m$, $m, n \in \mathbb{N}_0$, $|m| \le n$, are defined (following [29]) as

$$Y_n^m(\theta,\phi) \coloneqq \sqrt{\frac{2n+1}{4\pi}\frac{(n-|m|)!}{(n+|m|)!}} \cdot P_n^{|m|}(\cos\theta) e^{im\phi}, \tag{9}$$

where $P_n^m$ is the associated Legendre function of order $m$ and degree $n$. Substituting (8) into (7), we obtain that for $0 \le |a| < |b|$,

$$\mathcal{G}(a,b) = \sum_{n=0}^{\infty} \frac{1}{2n+1} \frac{|a|^n}{|b|^{n+1}} \sum_{m=-n}^{n} Y_n^m(\theta_a,\phi_a) Y_n^{-m}(\theta_b,\phi_b). \tag{10}$$

This series allows us to expand the Green's function as follows. For a given choice of source point $b$ and expansion center $c$, one can define local expansion coefficients as a doubly-indexed sequence

$$L_n^m \coloneqq \frac{1}{2n+1} \frac{Y_n^{-m}(\theta_{b-c},\phi_{b-c})}{|b-c|^{n+1}} \tag{11}$$

for integer $|m| \le n$ and $n \in \mathbb{N}_0$. Then the Green's function evaluated at a target $a$, with $|a-c| < |b-c|$, can be written as

$$\mathcal{G}(a,b) = \sum_{n=0}^{\infty} \sum_{m=-n}^{n} L_n^m |a-c|^n Y_n^m(\theta_{a-c},\phi_{a-c}). \tag{12}$$

A $p$-th order expansion is one in which the series (12) is truncated to the first $p+1$ terms.

## 2.4 Quadrature by Expansion

Using the local expansion of the potential as an analytical tool, we are ready to discuss the fundamentals of QBX. Among the current work on QBX, two major variants of the scheme have been considered by various authors: 'global' and 'local.' The variant that is the subject of this paper is a global scheme that borrows some ideas from recent work on local QBX. Hence, we review both of these variants in this section. QBX may be understood as a discretization involving two inter-related stages: formation of a truncated local expansion and smooth quadrature.

**First Stage: Formation of a Truncated Local Expansion.** With reference to the source surface $\Gamma$, for each target point $t$ close to or on the surface, this stage chooses a point $c$ to act as an expansion center. The distance $|t-c|$ from $c$ to $\Gamma$ is called the *expansion radius*. Using the selected center, this stage forms a local expansion about $c$ to mediate the potential $\mathcal{S}\mu$.

In *global* QBX, the potential $\mathcal{S}\mu$ due to the entire source geometry is expanded about $c$. A sufficient condition for the convergence of the expansion is that $|t-c| \leq \operatorname{dist}(c,\Gamma)$. By applying (11), one defines QBX coefficients via the integrals

$$(L_{\text{global}})_n^m(c) \coloneqq \frac{1}{2n+1}\int_\Gamma \frac{Y_n^{-m}(\theta_{y-c},\phi_{y-c})}{|y-c|^{n+1}}\mu(y)\,dS(y). \tag{13}$$

Then, fixing an expansion order $p \in \mathbb{N}_0$, the coefficients $(L_{\text{global}})_n^m(c)$, for $|m| \leq n$ for all $n \leq p$, are used to approximate the single layer potential in a series centered at $c$

$$(\mathcal{S}_{\text{global}}^{(p)})\mu(t) \coloneqq \sum_{n=0}^{p}\sum_{m=-n}^{n}(L_{\text{global}})_n^m(c)|t-c|^n Y_n^m(\theta_{t-c},\phi_{t-c}). \tag{14}$$

In contrast, local QBX mediates only a part of $\mathcal{S}\mu$ due to source geometry in a neighborhood of the target via a local expansion. In *local QBX*, formation of the expansion starts with the splitting

$$(\mathcal{S}\mu)(t) = \left(\mathcal{S}\ \mu|_{\Gamma_{\text{local},t}}\right)(t) + \left(\mathcal{S}\ \mu|_{\Gamma\setminus\Gamma_{\text{local},t}}\right)(t),$$

where the region $\Gamma_{\text{local},t} \subseteq \Gamma$ is chosen in such a way as to include the nearly singular/singular portion of the integrand. We therefore have

$$\left(\mathcal{S}\ \mu|_{\Gamma_{\text{local},t}}\right)(t) = \int_{\Gamma_{\text{local},t}} \mathcal{G}(t,s)\mu(s)\,dS(s),$$
$$\left(\mathcal{S}\ \mu|_{\Gamma\setminus\Gamma_{\text{local},t}}\right)(t) = \int_{\Gamma\setminus\Gamma_{\text{local},t}} \mathcal{G}(t,s)\mu(s)\,dS(s).$$

One then should ensure that $|t-c| \leq \operatorname{dist}(c,\Gamma_{\text{local},t})$. The QBX coefficients are defined via integrals

$$(L_{\text{local},t})_n^m(c) \coloneqq \frac{1}{2n+1}\int_{\Gamma_{\text{local},t}} \frac{Y_n^{-m}(\theta_{y-c},\phi_{y-c})}{|y-c|^{n+1}}\mu(y)\,dS(y), \tag{15}$$

and the approximation to the potential is given by

$$(\mathcal{S}_{\text{local},t}^{(p)})\mu(t) \coloneqq \sum_{n=0}^{p}\sum_{m=-n}^{n}(L_{\text{local},t})_n^m(c)|t-c|^n Y_n^m(\theta_{t-c},\phi_{t-c}) + \left(\mathcal{S}\ \mu|_{\Gamma\setminus\Gamma_{\text{local},t}}\right)(t). \tag{16}$$

The difference between the expansions (14) or (16) and the value of the layer potential is termed the truncation error. The following result gives an accuracy estimate for the expansion (14) used by global QBX, for the case that $t$ is an on-surface target. Bounds are available for the off-surface case through the same analysis.

**Lemma 1** (QBX truncation error, based on [10, Thm 3.1])**.** *Suppose that $\Gamma$ is smooth, non-self-intersecting and let $r > 0$. Suppose that $\{x : |x-c| \leq r\} \cap \Gamma = \{t\}$. Then for each $p > 0$ and $\delta > 0$, a constant $M_{p,\delta}$ exists such that*

$$\left|(\mathcal{S}\mu)(t) - (\mathcal{S}_{\text{global}}^{(p)})\mu(t)\right| \leq M_{p,\delta} r^{p+1}\|\mu\|_{W^{3+p+\delta,2}(\Gamma)}. \tag{17}$$

Error estimates for local QBX evaluation (16) can be found in [29]. The primary difference between the truncation error in global QBX and local QBX is that for local QBX, there is a dependence on the ratio $r/R$, where $r$ is the expansion radius and $R$ is the distance between the expansion center $c$ and the nearest boundary point of the surface region $\Gamma_{\text{local},t}$. This error can be interpreted as the error associated with the non-smooth transition between the expansion-mediated contribution from the local neighborhood $\Gamma_{\text{local},t}$ and the non-expansion-mediated 'far-field.'

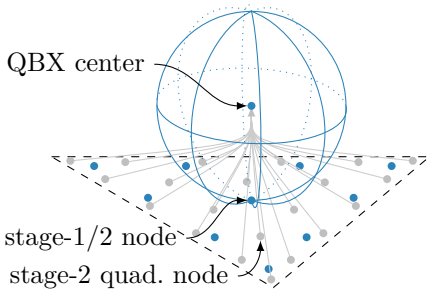


Figure 2: Depiction of a stage-1 discretization, stage-2 discretization, and a stage-2 quadrature discretization on a triangular element. In this example the stage-1 and stage-2 discretizations coincide; in general they may be different. The stage-2 quadrature discretization provides the quadrature nodes for the local expansion of the potential formed at the QBX centers. See also Figure 1.

**Second Stage: Smooth High-Order Quadrature.** In the second stage, the QBX coefficients $(L_{\text{global}})_n^m(c)$ or $(L_{\text{local},t})_n^m(c)$, and, in the local case, the 'far-field' integral are approximated with a quadrature rule, commonly with smooth high-order 'panel'-based quadrature discussed in Section 2.2. When the source points are close to the expansion center, accuracy of the quadrature approximation depends strongly on the expansion radius. Specifically, to maintain high-order quadrature accuracy, the expansion center must be placed sufficiently far from the surface element spawning the center as well as nearby elements, where the critical distance depends on the 'size' of the element. As an example providing quantitative detail, the following result, due to af Klinteberg and Tornberg [18], gives the asymptotic error for the case of a smooth tensor product rule over a flat $2h \times 2h$ panel.

**Lemma 2** (QBX quadrature error, flat $2h \times 2h$ panel [18, eqn. (157)])**.** *Let $\Gamma = [-h, h] \times [-h, h] \times \{0\}$, and let $t = (x, y, 0) \in \Gamma$ be a target point and $c = (x, y, r)$ be the corresponding expansion center. Let $\mu : \Gamma \to \mathbb{R}$ be a density function defined on $\Gamma$. Assume that $\mu$ is smooth (see [18, Sec. 3.3] for more discussion concerning smoothness of $\mu$). Suppose that the coefficients $(L_{\text{global}})_n^m(c)$ to the series (14) up to order $p$ are computed using a $q$-point Gauss-Legendre tensor product rule. Then a constant $C > 0$ exists, independent of $h$, $\mu$, $x$, $y$, $r$, $p$, and $q$, such that the quadrature-based approximation $(\tilde{\mathcal{S}}_{\text{global}}^{(p)})\mu(t)$ to the QBX expansion $(\mathcal{S}_{\text{global}}^{(p)})\mu(t)$ satisfies the following error bound asymptotically as $q \to \infty$:*

$$\left|(\mathcal{S}_{\text{global}}^{(p)})\mu(t) - (\tilde{\mathcal{S}}_{\text{global}}^{(p)})\mu(t)\right| \lesssim C\frac{h}{q}\sum_{l=1}^{p}\frac{1}{\sqrt{l}}\left(\frac{4qre}{hl}\right)^{l} e^{-4qr/h}\|\mu\|_{\infty}.$$

When combined, Lemmas 1 and 2 suggest that, with careful control over the expansion radius, quadrature order, and element size, QBX may be used to obtain high-order quadrature accuracy for on-surface targets $t$ (and, completely analogously, for off-surface targets as well). The two stages of QBX described in this section are interrelated, in that the choice of quadrature discretization influences the choice of expansion radius. For local QBX, an additional degree of freedom is the size of the local neighborhood $\Gamma_{\text{local},t}$, which has non-trivial implications on truncation error and quadrature error.

The remainder of this paper focuses on the global scheme. The next section discusses how to achieve the type of error control described in the previous paragraph for global QBX, on user-supplied meshes of smooth geometries.

## 2.5 Surface Discretization

In the remainder of this paper, we will assume that the surface discretization is an unstructured triangular surface mesh given as the union of images of a triangular reference element under a polynomial mapping. Our specific choice of basis for reference nodes/degrees of freedom follows [31], and for quadrature nodes is taken from [34]. A density on the discretization is represented by its values at the mapped counterparts of quadrature or interpolation nodes given on a reference element.

As part of the GIGAQBX algorithm in three dimensions, the work in [32] describes an algorithm for preparing an arbitrary smooth geometry to serve as a surface source/target discretization for applying QBX. We briefly review its main steps for the benefit of the reader. More details, including considerations for efficient implementation, may be found there.

The mesh processing algorithm for GIGAQBX involves four versions of the surface discretization, connected via interpolation operators that bring density values from one version to a subsequent refined version. First, the unmodified mesh is received from an external mesh generator. Second, the mesh is refined by iterative bisection of the elements $\triangle$ to avoid conflicts of the QBX expansion balls with other source geometry, to obtain the version called the 'stage-1 discretization'. Expansion centers are placed in the normal directions at the stage-1 interpolation nodes (which become on-surface targets) at a distance proportional to the element size (Figure 1), and remain fixed in the subsequent discretizations. Third, the mesh is refined with iterative bisection to assure sufficient quadrature resolution for all interactions (called the 'stage-2 discretization'). Lastly, the quadrature nodes in the mesh are oversampled (by increasing the quadrature order) to ensure accurate evaluation of the QBX coefficient integrals (13), producing the 'stage-2 quadrature discretization' (Figure 2), providing the source points for the 'point potential' approximation to the layer potential. The latter discretization is suitable as input to an algorithm for the evaluation of point potentials such as the FMM.

## 2.6 FMM Acceleration

The focus of the contributions [33, 32] is the accelerated evaluation of the quadrature for layer potentials. This is done using an appropriately modified version of the Fast Multipole Method (FMM). Algebraically, QBX may be regarded as the evaluation of the *local expansion* of the point potential (2) due to the quadrature nodes. Because such local expansions may be viewed as the output of the FMM, a natural approach to modifying the FMM for QBX is to use the FMM to form local expansions of the potential at the QBX expansion centers. The first practical implementation of this approach, as described in [26], is not backed by error estimates and does not achieve the same level of accuracy for a given FMM expansion order as the 'point' FMM, though an empirically determined increase in FMM order can recover accuracy, at some expense. As explored in detail in [33], the geometrical root cause of this loss of accuracy is that the FMM separation constraints for accurate evaluation of 'point' potentials are not strong enough to prevent inaccurate contributions from entering the QBX expansion.

The main modification to the QBX FMM in [33, 32] permits targets, such as QBX expansion balls, to be 'sized'. Similar to point targets, sized targets have a near-field that disallows certain nearby sources from using expansion mediation, permitting for analytical accuracy bounds to be established. This is accounted for by only allowing sized targets to protrude beyond their containing boxes by at most a given factor relative to the box size, called the *target confinement factor*. If a sized target cannot fit in a child box, it remains in the parent box. These changes require a careful reworking of the definitions of a number of aspects of the classical Fast Multipole algorithm for accuracy and scalability of the resulting method. The resulting method was termed *GIGAQBX*—for 'Geometric Global Accelerated QBX'—in [33, 32].

The redefinition of the near-field of a sized target entails a larger number of 'direct' FMM interactions at the QBX centers. Assuming the size of the QBX near-field remains bounded, this does not threaten the theoretical scaling of the algorithm. However, a practical implication of this is that, in three dimensions, direct interactions may take a large portion of time due to the high cost of expansion formation. The main contribution of this paper consists of an approach for reducing this cost, discussed next.

## 2.7 Target-Specific Expansions

Siegel and Tornberg [29] observe that the series expansion of the Green's function (7) provides a way to accelerate the formation of QBX expansions in certain circumstances. Consider the computational problem of evaluating the local expansion due to $n_s$ sources at $n_t$ targets. Assume for simplicity that we use a single expansion center mediating the expansion of the entire potential. We describe two approaches to this problem.

In the first approach, using the formula (10), for each source point $s$ we compute $(p+1)^2$ intermediate local coefficients, which takes $O((p+1)^2)$ time per source with well-known recurrences. After combining local coefficients additively to obtain $(p+1)^2$ final coefficients, we evaluate the local expansion at each target point,

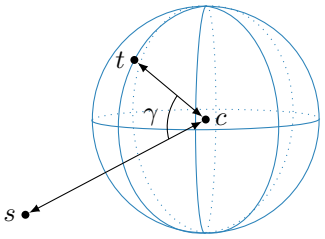


Figure 3: Components of a target-specific QBX expansion.

which costs $O((p+1)^2)$ at each target, again using well-known recurrences. It follows that this approach requires $O((n_s+n_t)(p+1)^2)$ work.

In the second approach, we use formula (7). Recall that this formula implies that the $p$-th order local expansion of the potential due to $s$ about the center $c \in \mathbb{R}^3$, with coefficients $L_n^m$, satisfies

$$\sum_{n=0}^{p} \sum_{m=-n}^{n} L_n^m |t-c|^n Y_n^m(\theta_{t-c}, \phi_{t-c}) = \frac{1}{4\pi} \sum_{n=0}^{p} \frac{|t-c|^n}{|s-c|^{n+1}} P_n(\cos\gamma), \tag{18}$$

where $\gamma$ is the angle between $s-c$ and $t-c$. A depiction of the geometrical situation is given in Figure 3. The quantity (18) requires $p+1$ summation terms and, using recurrences for the Legendre polynomials $P_n$, can be evaluated in $O(p+1)$ time. It must be evaluated once per source/target pair. It follows that the total cost of the second approach is $O(n_s n_t(p+1))$.

The first approach is the one used internally within the FMM. It has the advantage of scaling linearly in the number of sources and targets. The disadvantage is the high cost of $O((p+1)^2)$ operations per particle.

The second approach, while not scaling linearly in the number of particles, is computationally advantageous over the first approach if the number of sources or targets is small. Such a situation arises in on-surface evaluation for QBX when there is one target per center (Figure 1), a common evaluation pattern.

The second approach is termed *target-specific* because, unlike the first approach, the 'local coefficient' $P_n(\cos\gamma)/|s-c|^{n+1}$ depends on the target through the angle $\gamma$. Because of this, it is generally unsuitable for use within a point FMM, which is premised on using expansions that separate the influence of the source and the targets. However, target-specific expansions are suitable for use within the GIGAQBX FMM whenever a QBX local expansion is formed directly from sources. Furthermore, analogous formulas for target-specific QBX are available for different kernels derived from the Laplace and Helmholtz kernels, making this approach general. We give some of these in Appendix A.

In the next section, we discuss the incorporation of target-specific QBX into the GIGAQBX algorithm.

# 3 Algorithm

The algorithm in this section is based on the GIGAQBX algorithm in three dimensions [32]. For the benefit of the reader familiar with the version in [32], we briefly point out the main modifications. The primary change in the algorithm presented in this paper is the use of target-specific expansions (Section 2.7) to mediate the contributions of the 'directly evaluated' portion of the field—i.e., List 1, List 3 close, and List 4 close. This replaces the formation of spherical harmonic expansions at the QBX centers due to this portion of the field. However, QBX local expansions in spherical harmonics, formed at the QBX centers, remain part of the algorithm as computational entities since they mediate the potential due to the 'far-field' as well as the field due to boxes in List 3 far.

There are two other less significant differences from the statement of the algorithm in [32]. First, we give a more precise statement of the construction of the octree in this paper than that was given in [32], when it comes to distinguishing what type of particles may be 'owned' by boxes. Second, we introduce the notion of a 'List 3 far candidate box,' and we modify the criteria for placement of boxes in List 3 far by introducing a source count threshold.

## 3.1 Notation

In this section, we introduce the same notation as [32] in support of the precise statement of the modified algorithm. Let $b$ be a box in an octree with center $c$. We use the notation $\overline{B_\infty}(r, c)$ to denote the set $\{x \in \mathbb{R}^3 : |x - c|_\infty \leq r\}$ and $\overline{B_2}(r, c)$ to denote the set $\{x \in \mathbb{R}^3 : |x - c| \leq r\}$.

We use $|b|$ to refer to the $\ell^\infty$ radius of the box, i.e. half the box width.

The *target confinement region (TCR)* of $b$, or $\mathsf{TCR}(b)$, is the set $\overline{B_2}(\sqrt{3}|b|(1 + t_f), c)$, where $t_f \geq 0$ is the *target confinement factor*.

The *$k$-near neighborhood* of $b$ is the region $\overline{B_\infty}(|b|(1 + 2k), c)$. The *$k$-colleagues* of $b$ are same-level boxes contained inside the $k$-near neighborhood of $b$. In particular, $T_b$ denotes the set of 2-colleagues of a box $b$.

Two same-level boxes that are not $k$-colleagues are termed *$k$-well-separated*.

$\mathsf{Parent}(b)$ denotes the parent of $b$.

The sets $\mathsf{Ancestors}(b)$ and $\mathsf{Descendants}(b)$ denote the sets of ancestors and descendants of $b$. These are also defined for a set of boxes, as the union of the boxes' sets of ancestors or descendants, respectively.

A box owning a point or QBX center target is called a *target box*. A box owning a source quadrature node is called a *source box*. Ancestors of target boxes are called *target-ancestor boxes*.

Two boxes are *adjacent* if the intersection of their boundaries is non-empty, i.e. they share a common face, edge, or corner.

We define a relation $\prec$ over the set of boxes and target confinement regions within the tree, with $a \prec b$ to be read as '$a$ is adequately separated from $b$, relative to the size of $a$'. We write $a \prec \mathsf{TCR}(b)$ for boxes $a$ and $b$ if the $\ell^2$ distance from the center of $a$ to the boundary of $\mathsf{TCR}(b)$ is at least $3|a|$. We write $\mathsf{TCR}(a) \prec b$ for boxes $a$ and $b$ if the $\ell^2$ distance from the center of $a$ to the boundary of $b$ is at least $3|a|(1 + t_f)$. (As $\ell^2$ distance is bounded from below by $\ell^\infty$ distance, a computationally convenient approximation for checking whether $\mathsf{TCR}(a) \prec b$ is to check if the $\ell^\infty$ distance from the center of $a$ to the boundary of $b$ is at least $3|a|(1 + t_f)$. This check is sufficient (but not necessary) for $\mathsf{TCR}(a) \prec b$. We use this approximation in our implementation.) We write $a \not\prec b$ to denote the negation of $a \prec b$.

## 3.2 Interaction Lists

The computational domain of the algorithm is a box with equal-length sides (such as the closed cube $[-1, 1]^3$). This 'root box' is recursively partitioned into equal-sized octants, together forming an octree. The root box contains all source quadrature nodes, QBX centers, and target points, which we refer to generically as *particles*, and also the entirety of each QBX ball. Target points are classified as either *conventional targets* and *QBX targets*, the latter requiring potential evaluation through a QBX expansion. The class of *thresholded particles* includes every particle type except QBX targets. These are the particles to which the box particle count threshold applies, as described below. Boxes, regardless of their having children, may 'own' a subset of thresholded particles. QBX targets are not owned by boxes but are instead 'associated' with a QBX ball, which in turn is 'owned' by a box.

To construct the octree, boxes of the tree owning more than $n_{\max} > 0$ (a user-set parameter) thresholded particles are iteratively subdivided, transferring the ownership of particles into the child boxes, until the number of thresholded particles per leaf (childless) box is below $n_{\max}$, or if all potentially split boxes are empty due to constraints on QBX center placement. A QBX center whose expansion ball cannot be contained in the TCR of the child box is not transferred to the child and remains owned by the parent.

Information on the parts of the potential travels between boxes through *translation operators*, from sets of boxes indicated by *interaction lists*, which are lists of boxes attached to target or target-ancestor boxes in the tree. These lists are based around a *near-field* $T_b$ consisting of the 2-colleagues of a given box; i.e. the same-level nearest-neighbors and second nearest-neighbors. For a more detailed description of the interaction lists in GIGAQBX, see [33, 32].

For a given box $b$, *List 1* consists of interactions with adjacent boxes.

**Definition 1** (List 1, $U_b$ [32, Def. 2])**.** *For a target box $b$, $U_b$ consists of all leaf boxes from among* $\mathsf{Descendants}(b) \cup \{b\}$ *and the set of boxes adjacent to $b$.*

*List 2* consists of interactions with non-adjacent same-level boxes that are descendants of the near-field of the parent.

**Definition 2** (List 2, $V_b$ [32, Def. 3]). *For a target or target-ancestor box $b$, $V_b$ consists of the children of the 2-colleagues of $b$'s parent that are 2-well-separated from $b$.*

*List 3* consists of interactions between non-adjacent boxes where the source box is in the near-field of the target.

**Definition 3** (List 3, $W_b$ [32, Def. 4]). *For a target box $b$, a box $d \in \mathsf{Descendants}(T_b)$ is in $W_b$ if $d$ is not adjacent to $b$ and, for all $w \in \mathsf{Ancestors}(d) \cap \mathsf{Descendants}(T_b)$, $w$ is adjacent to $b$.*

*List 4* consists of interactions between non-adjacent boxes where the target box is in the near-field of the source.

**Definition 4** (List 4, $X_b$, [32, Def. 5]). *For a target or target-ancestor box $b$, a source box $d$ is in List 4 of $b$ if $d$ is a 2-colleague of some ancestor of $b$ and $d$ is adjacent to $\mathsf{Parent}(b)$ but not $b$ itself. Additionally, a source box $d$ is in $X_b$ if $d$ is a 2-colleague of $b$ and $d$ is not adjacent to $b$.*

Lists 1–4 (cf. [5]) are generalizations of the interaction lists present in most variants of the FMM, modified for a '2-away' near-field and the presence of targets in non-leaves. To these definitions, we adjoin a set of 'close' and 'far' lists. The purpose of these lists is to ensure that interactions directed at QBX centers maintain sufficient separation so that intermediate translations involving QBX centers have controlled accuracy. The field due to a 'close' list is evaluated directly (without the use of intermediate expansions), while the field due to a 'far' list is sufficiently separated to allow for the use of intermediate expansions.

*List 3 close* and *List 3 far* consist of boxes from List 3 and their descendants. A box is placed into one of these lists depending on whether it is adequately separated from the TCR of the target box, and whether it exceeds a certain 'source count' threshold.

In order to define the close and far lists associated with $W_b$, we introduce the notion of a 'List 3 far candidate' box.

**Definition 5** (List 3 far candidate box). *For a target box $b$, a box $d$ is a* List 3 far candidate of $b$ *if (a) $d \in W_b \cup \mathsf{Descendants}(W_b)$, (b) $d \prec \mathsf{TCR}(b)$, and, (c) for all $w \in \mathsf{Ancestors}(d) \cap (\mathsf{Descendants}(W_b) \cup W_b)$, $w \not\prec \mathsf{TCR}(b)$.*

In other words, a List 3 far candidate box is a box in the near-field which is the largest box which is adequately separated from the TCR of the target among itself and its chain of ancestors. It follows that each ancestor chain of boxes contains at most a single List 3 far candidate. The 'multipole threshold', $n_{\text{mpole}} \geq 0$, a user-set threshold related to the cumulative number of sources in the descendants of a List 3 far candidate, contributes to the placement of candidates into List 3 far or List 3 close.

**Definition 6** (List 3 far, $W_b^{\text{far}}$, modified from [32, Def. 7]). *For a target box $b$, a List 3 far candidate $d$ of $b$ is said to be in $W_b^{\text{far}}$ if the cumulative number of sources owned by $d$ and its descendants is at least $n_{\text{mpole}}$.*

**Definition 7** (List 3 close, $W_b^{\text{close}}$, modified from [32, Def. 6]). *For a target box $b$, a leaf box $d$ is said to be in $W_b^{\text{close}}$ if $d \in \mathsf{Descendants}(W_b) \cup W_b$ and one of the following is true: (a) $d \not\prec \mathsf{TCR}(b)$, (b) a box $w \in \{d\} \cup \mathsf{Ancestors}(d)$ exists such that $w$ is a List 3 far candidate box of $b$ and the cumulative number of sources owned by $w$ and its descendants is less than $n_{\text{mpole}}$.*

*List 4 close* and *List 4 far* consist of boxes that are in the List 4 of a target/target ancestor box or its chain of ancestors. A box is placed into List 4 close if the TCR of the target/target ancestor box is not adequately separated from it; otherwise it is placed in List 4 far.

**Definition 8** (List 4 close, $X_b^{\text{close}}$, [32, Def. 8]). *Let $b$ be a target or target-ancestor box. A box $d$ is in $X_b^{\text{close}}$ if for some $w \in \mathsf{Ancestors}(b) \cup \{b\}$ we have $d \in X_w$ and furthermore $\mathsf{TCR}(b) \not\prec d$.*

**Definition 9** (List 4 far, $X_b^{\text{far}}$, [32, Def. 9]). *Let $b$ be a target or target-ancestor box. A box $d \in X_b$ is in List 4 far if $\mathsf{TCR}(b) \prec d$. Furthermore, if $b$ has a parent, a box $d \in X_{\mathsf{Parent}(b)}^{\text{close}}$ is in List 4 far if $\mathsf{TCR}(b) \prec d$.*

### 3.3 Formal Statement

For completeness, we give a full statement of the GIGAQBX algorithm using target-specific expansions in Algorithm 1. We use the following notation for 'point' potentials (those formed without QBX mediation): (a) $\mathsf{P}_b^{\text{near}}(t)$ denotes the potential at a target point $t$ due to all sources in $U_b \cup W_b^{\text{close}} \cup X_b^{\text{close}}$, and (b) $\mathsf{P}_b^{W}(t)$ denotes the potential at a target $t$ due to all sources in $W_b^{\text{far}}$. For QBX-mediated potentials, we use the following notation, where $c$ is a QBX center owned by a box $b$: (a) $\mathsf{L}_c^{\text{qbx,near}}(t)$ denotes the (QBX) local expansion at a target $t$ due to all sources in $U_b \cup W_b^{\text{close}} \cup X_b^{\text{close}}$, (b) $\mathsf{L}_c^{\text{qbx},W}(t)$ denotes the (QBX) local expansion at a target $t$ due to all sources in $W_b^{\text{far}}$, and (c) $\mathsf{L}_c^{\text{qbx,far}}(t)$ denotes the (QBX) local expansion at a target $t$ due to sources not in $U_b \cup W_b \cup X_b^{\text{close}}$.

In Algorithm 1, the stages changed to use target-specific expansions are indicated with a star (★).

---

**Algorithm 1:** GIGAQBX FMM with Target-Specific Expansions

---

**Require:** The maximum number of thresholded FMM targets/sources $n_{\text{max}}$ per box for octree refinement, a multipole threshold $n_{\text{mpole}}$, and a target confinement factor $t_f$ are chosen.
**Require:** The input geometry and targets are preprocessed according to [32, Sec. 3].
**Require:** Based on the precision $\varepsilon$ to be achieved, a QBX order $p_{\text{qbx}}$, an FMM order $p_{\text{fmm}}$, and an oversampled quadrature node count $p_{\text{quad}}$ are chosen.
**Ensure:** An accurate approximation to the potential at all target points is computed.
{★ indicates that a stage uses target-specific expansions.}

*Stage 1: Build tree*
Create an octree on the computational domain containing all sources, targets, and QBX centers, as well as the entirety of each expansion ball.
**repeat**
  Subdivide each box owning more than $n_{\text{max}}$ thresholded particles into eight children, pruning any empty child boxes. If a QBX center cannot be owned by the child box with target confinement factor $t_f$ due to its radius, it remains in the parent box.
**until** no box needs subdivision

*Stage 2: Form multipoles*
**for all** boxes $b$ **do**
  Form a $p_{\text{fmm}}$-th order multipole expansion $\mathsf{M}_b$ centered at $b$ due to sources owned by $b$.
**end for**
**for all** boxes $b$ in postorder **do**
  For each child of $b$, shift the center of the multipole expansion at the child to the center of $b$. Add the resulting expansions to $\mathsf{M}_b$.
**end for**

★*Stage 3: Evaluate direct interactions*
**for all** boxes $b$ **do**
  For each non-QBX target $t$ owned by $b$, add to $\mathsf{P}_b^{\text{near}}(t)$ the contribution due to the interactions from sources owned by boxes in $U_b$ to $t$.
**end for**
**for all** boxes $b$ **do**
  For each QBX target $t$ associated to a QBX center $c$ owned by $b$, use target-specific expansions to add to $\mathsf{L}_c^{\text{qbx,near}}(t)$ the contribution due to the interaction from all sources in $U_b$.
**end for**

*Stage 4: Translate multipoles to local expansions*
**for all** boxes $b$ **do**
  For each box $d \in V_b$, translate the multipole expansion $\mathsf{M}_d$ to a local expansion centered at $b$. Add the resulting expansions to obtain $\mathsf{L}_b^{\text{far}}$.
**end for**

★*Stage 5(a): Evaluate direct interactions due to* $W_b^{\text{close}}$
Repeat Stage 3 with $W_b^{\text{close}}$ instead of $U_b$.

*Stage 5(b): Evaluate multipoles due to $W_b^{\text{far}}$*
**for all** boxes $b$ **do**
For each conventional target $t$ owned by $b$, evaluate the multipole expansion $\mathsf{M}_d$ of each box $d \in W_b^{\text{far}}$ to obtain $\mathsf{P}_b^W(t)$.
**end for**
**for all** boxes $b$ **do**
For each QBX center $c$ owned by $b$, compute the expansion $\mathsf{L}_c^{\text{qbx},W}$, due to the multipole expansion $\mathsf{M}_d$ of each box $d \in W_b^{\text{far}}$.
**end for**

★*Stage 6(a): Evaluate direct interactions due to $X_b^{\text{close}}$*
Repeat Stage 3 with $X_b^{\text{close}}$ instead of $U_b$.

*Stage 6(b): Form locals due to $X_b^{\text{far}}$*
**for all** boxes $b$ **do**
Convert the field of every source particle owned by boxes in $X_b^{\text{far}}$ to a local expansion about $b$. Add to $\mathsf{L}_b^{\text{far}}$.
**end for**

*Stage 7: Propagate local expansions downward*
**for all** boxes $b$ in preorder **do**
For each child $d$ of $b$, shift the center of the local expansions $\mathsf{L}_b^{\text{far}}$ to the child. Add the resulting expansion to $\mathsf{L}_d^{\text{far}}$.
**end for**

*Stage 8: Form local expansions at QBX centers*
**for all** boxes $b$ **do**
For each QBX center $c$ owned by $b$, translate $\mathsf{L}_b^{\text{far}}$ to $c$, obtaining $\mathsf{L}_c^{\text{qbx,far}}$.
**end for**

*Stage 9: Evaluate final potential at targets*
**for all** boxes $b$ **do**
For each non-QBX target $t$ owned by $b$, evaluate $\mathsf{L}_b^{\text{far}}(t)$. Add $\mathsf{P}_b^{\text{near}}(t)$, $\mathsf{P}_b^W(t)$, and $\mathsf{L}_b^{\text{far}}(t)$ to obtain the potential at $t$.
**end for**
**for all** boxes $b$ **do**
For each QBX target $t$ associated to a QBX center $c$ owned by $b$, add $\mathsf{L}_c^{\text{qbx,near}}(t)$, $\mathsf{L}_c^{\text{qbx},W}(t)$, and $\mathsf{L}_c^{\text{qbx,far}}(t)$ to obtain the potential at $t$.
**end for**

---

## 3.4 Accuracy and Scaling

The accuracy bound for the error introduced in acceleration in Algorithm 1 is unchanged compared with the original algorithm [32, Thm. 6], due to the fact that target-specific expansions are mathematically identical to their target-independent counterparts. In essence, assuming the validity of a number of hypotheses with strong numerical evidence [32, Hyp. 1–3], the asymptotic acceleration error is $O\left((3/4)^{p_{\text{fmm}}+1}\right)$ (the same as the 'one-away' point FMM in three dimensions [24]) when $t_f \leq 0.84$.

One can analyze the scaling of Algorithm 1 in a way that closely parallels that of [32]. The main difference in the analysis is connected with the change in the asymptotic complexity due to the use of target-specific expansions. In Table 1, we provide a summary of the asymptotic complexity of the stages of the algorithm. The complexity is measured in terms of asymptotic floating point operations (see Section 4.1).

We make a number of simplifying assumptions in the complexity model. The first is that $p_{\text{qbx}} \leq p_{\text{fmm}}$, which is true in all practical situations as the error introduced by FMM acceleration decreases more slowly with increasing order compared with the error due to truncation of the QBX expansion (cf. Lemma 1). Secondly, we assume the use of spherical harmonic expansions with 'point-and-shoot' translations (see Section 4.1).

Table 1: Complexity of each stage of Algorithm 1.

| Stage | Asymptotic Operation Count | Note |
|---|---|---|
| Stage 1 | $O(NL)$ | cf. [32, Tab. 1] |
| Stage 2 | $O(N_S n_{\text{fmm}} + N_B n_{\text{fmm}}^{3/2})$ | — " — |
| Stage 3 | $O((27(N_C + N_S)n_{\max} + N_C M_C)n_{\text{qbx}}^{1/2})$ | Using TSQBX; cf. [32, Lem. 11] |
| Stage 4 | $O(875 N_B n_{\text{fmm}}^{3/2})$ | cf. [32, Lem. 12] |
| Stage 5 | $O(N_C M_C n_{\text{qbx}}^{1/2} + 124 L N_S n_{\max} n_{\text{qbx}}^{1/2})$ | Using TSQBX; cf. [32, Lem. 13] |
| Stage 6(a) | $O(250 N_C n_{\max} n_{\text{qbx}}^{1/2})$ | Using TSQBX; cf. [32, Lem. 15] |
| Stage 6(b) | $O(375 N_B n_{\max} n_{\text{fmm}})$ | cf. [32, Lem. 15] |
| Stage 7 | $O(8 N_B n_{\text{fmm}}^{3/2})$ | cf. [32, Tab. 1] |
| Stage 8 | $O(N_C n_{\text{fmm}}^{3/2})$ | — " — |
| Stage 9 | $O(N_T n_{\text{qbx}})$ | — " — |

Finally, we assume that the algorithm is performing on-surface evaluation, in which the potential at every target is mediated through a QBX center and there is at most one target per QBX center. The presence of off-surface targets causes no significant changes to the complexity analysis. We omit it for the sake of simplicity.

The following parameters are used as inputs to the complexity model. $N$, $N_T$, $N_C$, $N_S$ refer respectively to the number of particles, targets, sources, and centers. $N_B$ refers to the number of boxes and $L$ refers to the number of levels in the tree. The quantity $M_C$, as found in [32, Lem. 8], is a measure of the average number of sources in a neighborhood of a QBX center, where the size of the neighborhood is proportional to the ball size, defined specifically as:

$$M_C \coloneqq \frac{1}{N_C} \sum_{c \in C} \left| S \cap \overline{B_\infty} \left( \frac{4\sqrt{3} r_c}{t_f}, c \right) \right|,$$

where $C$ is the set of QBX centers, $S$ is the set of sources, and $r_c$ is the expansion radius associated with the center $c$. $n_{\text{fmm}}$ and $n_{\text{qbx}}$ refer to the number of coefficients in an FMM and QBX expansion in spherical harmonics.

### 3.5 Cost Considerations

While Table 1 presents an understanding of the cost dependence of the GIGAQBX FMM's stages on the algorithmic parameters, the cost analysis is asymptotic rather than a predictor of the time duration from start to finish of the FMM, also known as the 'wall time.' In this section, we present a more qualitative understanding of the effect of algorithmic parameters on cost. The observations in this section motivate an empirical study of the cost of the GIGAQBX FMM with target-specific expansions in the next section.

When considering parameters that affect the cost of Algorithm 1, a complication arises in that some algorithmic parameters simultaneously have a large effect on both accuracy and cost, so that modifying these parameters could result in output that is vastly different in accuracy. A number of these parameters primarily affect the size of the QBX near-field (which is covered by List 1, List 3 close, and List 3 far): $t_f$, $p_{\text{quad}}$, and, to some extent, the input geometry itself. Additionally, the parameters $p_{\text{qbx}}$ and $p_{\text{fmm}}$ do not affect the size of QBX near-field but nevertheless have a major effect on accuracy. To ensure comparable levels of accuracy in the output of the algorithm before and after optimization, we take the point of view that parameters primarily affecting the accuracy of the layer potential evaluation are fixed.

This leaves two parameters to be considered that primarily affect the cost of the various stages of the algorithm, while leaving accuracy nearly unchanged. The first of these parameters is $n_{\max}$, the maximum number of thresholded particles per box. The main consideration in choosing a value of $n_{\max}$ is that as $n_{\max}$ increases, the number of boxes decreases, while simultaneously each box holds more particles. The cost of those stages where the amount of total computational effort increases with the number of boxes, such as

Stage 4 (List 2), benefits from a reduction in the number of boxes. On the other hand, stages involving direct interactions require more work as $n_{\max}$ increases.

The second of these parameters is $n_{\text{mpole}}$. It was mentioned in [33] as a degree of freedom for optimization but not examined in detail. It is based on the observation that we can avoid translating a multipole expansion into a QBX local expansion by replacing it with direct interactions with the source particles whose field makes up the multipole expansion. This is a less expensive evaluation strategy for multipole expansions whose cumulative source count is below a 'smallness' threshold.

# 4 Experimental Results

## 4.1 Cost Model

In this section, we present a cost model for Algorithm 1 which estimates the amount of computational time used by the algorithm in a way that is reproducible while remaining predictive of actual machine computation time. To aid the construction of a realistic and reproducible cost metric, the model makes use of direct examination of the FMM tree and data structures—a strategy we have found to yield data useful for a variety of purposes. The model produces an estimate of the total number of floating point operations required for the algorithm on a particular input geometry. Through the introduction of additional weight constants, we use these counts to approximate the total amount of computational time used by the algorithm, a quantity we refer to as the *modeled process time*. (The phrase 'process time' in UNIX-type operating systems is used to describe the total time spent executing process code, excluding the time executing operating system code, summed across all cores of a multi-core processor if relevant.)

The first step in obtaining an asymptotic estimate of the number of floating point operations is to count the number of interactions of each category—e.g., local expansion formation, multipole-to-local translation, multipole evaluation, etc.—performed by the FMM by analyzing the interaction lists and tree. The model multiplies each of these counts by a category-dependent symbolic expression, parametrized by the number of coefficients in the FMM and QBX expansions, to obtain an asymptotic number of floating point operations (e.g., for forming a multipole/local expansion with $(p+1)^2$ coefficients, the asymptotic amount of work is modeled as $(p+1)^2$). Lastly, the asymptotic number of floating point operations is multiplied by a category-dependent 'calibration constant', an empirically determined parameter representing a ratio of running time in seconds to modeled floating point operations.

The modeled process time per interaction category is shown in Table 2. To count the number of floating point operations, the cost model assumes the use of spherical harmonic expansions (see Section 2.3). For each category of interaction, the modeled number of floating point operations is designed to be asymptotically correct to leading order. We assume that translations between two expansions occur using a 'point-and-shoot' strategy — this reduces the cost of the relevant translations from $O((p+1)^4)$ to $O((p+1)^3)$ (see for instance [14]) for homogeneous source and target order $p$. The procedure for a 'point-and-shoot' translation between source order $p$ and target order $q$ is as follows:

1. At a cost of $O((p+1)^3)$, rotate the source expansion so that the translation direction is $z$-axis aligned.
2. At a cost of $O((p+1)(q+1)^2)$, translate the source expansion along the $z$-axis to the target expansion.
3. At a cost of $O((q+1)^3)$, rotate the target expansion back.

Our model includes a term for each of these three stages. In the case of a homogeneous source and target order, this entails using a leading factor of 3. Lastly, we model the cost of evaluation of a target-specific expansion as $O(p_{\text{qbx}}+1)$ floating-point operations, corresponding to the evaluation of formula (18) using recurrences for the Legendre polynomials.

The model is fitted to the results obtained from timing our implementation of Algorithm 1 on geometries of fixed QBX and FMM order. Timing data is obtained by timing each stage of our implementation on a 20-core 2.30 GHz Intel Xeon E5-2650 v3 machine. We use a least-squares fit to obtain calibration coefficients from the timing data. Our implementation, which uses double-precision floating point arithmetic throughout, is based on FMMLIB [13] compiled by GCC 7.2.0 with `-Ofast` and `-march=native` flags, and making use of shared memory parallelism via OpenMP. The calibration coefficients obtained for the order pair

$(p_{\text{qbx}}, p_{\text{fmm}}) = (5, 15)$ are displayed in Table 3. These were obtained by fitting to the process times for the 'urchin' geometries $\gamma_3, \gamma_5$ (see Section 4.2.1).

A natural interpretation of these coefficients is that they represent the time of a single 'asymptotic flop' in their respective interaction category. It is therefore not unreasonable to expect each flop to have an execution time roughly corresponding to the inverse of the clock frequency of the processor. Since the processor we used for our experiments has a clock frequency of 2.30 GHz, we expect and observe asymptotic flop times of a magnitude around $10^{-9}$. Further, any major discrepancies in their comparative magnitude may indicate a difference in implementation quality. While most of the calibration coefficients are of roughly the expected magnitude, the coefficient associated with QBX-local-to-target evaluation is much larger due to inefficiencies our implementation. (Despite the high overhead of this evaluation, it does not play a significant role in the overall cost of the scheme.)

Empirically, we have observed the model to give accurate cost estimates, within a few percent of the true execution time, for the same QBX and FMM order pair it is fitted to. As an example, Table 4 gives actual versus predicted process times for a sequence of 'urchin' test geometries introduction in Section 4.2.1, using the calibration constants from Table 3. This accuracy does not necessarily carry over when differing values of $p_{\text{qbx}}$ and $p_{\text{fmm}}$ are used from those that the model was fitted to, likely to the overhead of unmodeled lower-order costs in certain interactions in our implementation. While this issue may be addressed either extending the model to include more terms or by further optimization work to reduce the lower-order costs in the implementation, in this paper we handle this issue by re-fitting the model to each example that we use, ensuring its fidelity as a predictor of modeled process time.

## 4.2 Scaling and Balancing Study

This section presents a numerical study of the impact of our optimizations to the GIGAQBX algorithm. It is possible to use target-specific QBX as a direct replacement for the version of the algorithm in [32], leaving all other parameters in the algorithm unchanged, but the simple adaptation of a few algorithmic parameters has the potential to drastically affect performance characteristics by changing the proportion of work performed by different translation operators in the FMM. In the experiments of this section, we apply this procedure to Algorithm 1 in order improve algorithmic cost, by reducing performance bottlenecks, in a process known as 'balancing'.

For all experiments, we report on (a) the total modeled process time as well as (b) the modeled process times of the algorithmic stages that involve the interaction lists (see Section 3.2). In the remainder of this section, we freely use the name of an interaction list to refer to the stage it is associated with.

### 4.2.1 Test Geometry

As a first test of the scaling behavior and cost of our algorithm, we use a family of test geometries $\gamma_k$ that we call 'urchins', parametrized by $k \in \mathbb{N}$. These surfaces are given in spherical coordinates as $(r_k, \theta, \phi)$, using the definition of spherical harmonics in (9), letting $r_k$ vary with $\theta$ and $\phi$ as follows:

$$\begin{aligned} r_k(\theta, \phi) &= 0.2 + \frac{\operatorname{Re} Y_k^{\lfloor k/2 \rfloor}(\theta, \phi) - m_k}{M_k - m_k}, \\ M_k &= \max_{\theta \in [0,\pi], \phi \in [0,2\pi]} \operatorname{Re} Y_k^{\lfloor k/2 \rfloor}(\theta, \phi), \\ m_k &= \min_{\theta \in [0,\pi], \phi \in [0,2\pi]} \operatorname{Re} Y_k^{\lfloor k/2 \rfloor}(\theta, \phi). \end{aligned}$$

Figure 4 gives a visual impression of the geometry $\gamma_8$. The mesh of the geometry consists of triangular elements whose parameter mapping function is an 8th degree polynomial. The construction starts with the image of an icosahedral mesh under the mapping function $r_k$. To ensure that the piecewise polynomial elements accurately represent the geometry, an iterative refinement procedure is applied to the mapped elements, and at each iteration any refined elements are nodally reevaluated. The details of this procedure can be found in [32, Sec. 6].

In our numerical experiments, we use the family of geometries $\gamma_2, \gamma_4, \ldots, \gamma_{10}$, which range in size from 4,116,770–178,529,030 particles, where a 'particle' is a source, target, or QBX center. We use $(p_{\text{qbx}}, p_{\text{fmm}}) =$

Table 2: Cost model used in this paper for evaluation of the scaling of Algorithm 1, where $n_{\text{qbx}} = (1 + p_{\text{qbx}})^2$ and $n_{\text{fmm}} = (1 + p_{\text{fmm}})^2$.

| Interaction | Modeled Process Time (s) |
|---|---|
| Source → Local | $c_{\text{p2l}} \cdot n_{\text{fmm}}$ |
| Source → Multipole | $c_{\text{p2m}} \cdot n_{\text{fmm}}$ |
| Source → QBX Local | $c_{\text{p2qbxl}} \cdot n_{\text{qbx}}$ |
| Target-Specific QBX | $c_{\text{ts}} \cdot n_{\text{qbx}}^{1/2}$ |
| Local → Local | $3 \cdot c_{\text{l2l}} \cdot n_{\text{fmm}}^{3/2}$ |
| Local → QBX Local | $c_{\text{l2qbxl}} \cdot \left( n_{\text{fmm}}^{3/2} + n_{\text{fmm}}^{1/2} \cdot n_{\text{qbx}} + n_{\text{qbx}}^{3/2} \right)$ |
| Multipole → Local | $3 \cdot c_{\text{m2l}} \cdot n_{\text{fmm}}^{3/2}$ |
| Multipole → Multipole | $3 \cdot c_{\text{m2m}} \cdot n_{\text{fmm}}^{3/2}$ |
| Multipole → QBX Local | $c_{\text{m2qbxl}} \cdot \left( n_{\text{fmm}}^{3/2} + n_{\text{fmm}}^{1/2} \cdot n_{\text{qbx}} + n_{\text{qbx}}^{3/2} \right)$ |
| QBX Local → Target | $c_{\text{qbxl2p}} \cdot n_{\text{qbx}}$ |

Table 3: Calibration constants for the model in Table 2, obtained for the order pair $(p_{\text{qbx}}, p_{\text{fmm}}) = (5, 15)$ on a particular machine using an implementation based on FMMLIB.

| Constant | Value | Constant | Value |
|---|---|---|---|
| $c_{\text{p2l}}$ | $1.10 \cdot 10^{-8}$ | $c_{\text{l2qbxl}}$ | $4.72 \cdot 10^{-9}$ |
| $c_{\text{p2m}}$ | $1.24 \cdot 10^{-8}$ | $c_{\text{m2l}}$ | $3.24 \cdot 10^{-9}$ |
| $c_{\text{p2qbxl}}$ | $1.42 \cdot 10^{-8}$ | $c_{\text{m2m}}$ | $5.35 \cdot 10^{-9}$ |
| $c_{\text{ts}}$ | $9.45 \cdot 10^{-9}$ | $c_{\text{m2qbxl}}$ | $3.37 \cdot 10^{-9}$ |
| $c_{\text{l2l}}$ | $5.94 \cdot 10^{-9}$ | $c_{\text{qbxl2p}}$ | $6.74 \cdot 10^{-7}$ |

Table 4: Actual versus predicted process times using the model calibration constants in Table 3 for the 'urchin' geometries in Section 4.2.1, with fits obtained on the geometries $\gamma_3$, $\gamma_5$.

| | Process Time (s) | | | | |
|---|---|---|---|---|---|
| Kind | $\gamma_2$ | $\gamma_4$ | $\gamma_6$ | $\gamma_8$ | $\gamma_{10}$ |
| Actual | 1457.07 | 5741.86 | 20705.79 | 46238.00 | 98943.60 |
| Model | 1447.81 | 5731.83 | 20715.11 | 46261.86 | 98618.80 |

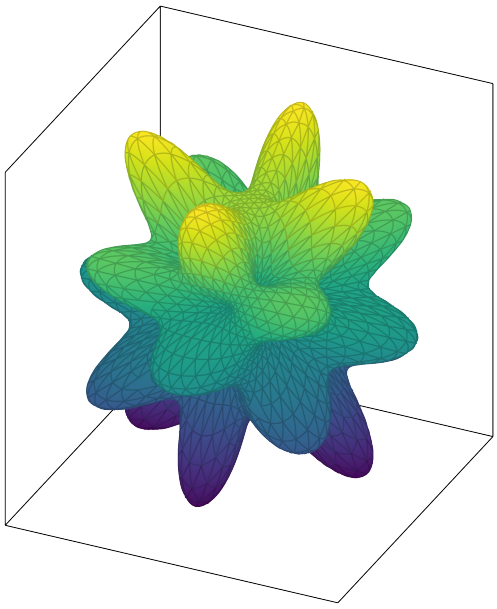

Figure 4: A mesh of the 'urchin' test geometry $\gamma_8$.

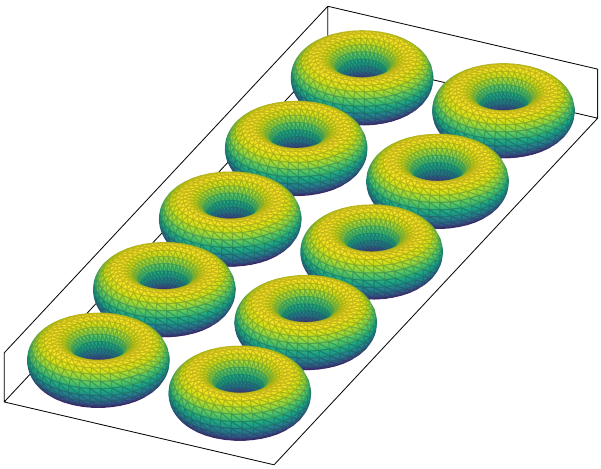

Figure 5: A mesh of the 'torus grid' test geometry $\tau_{10}$.

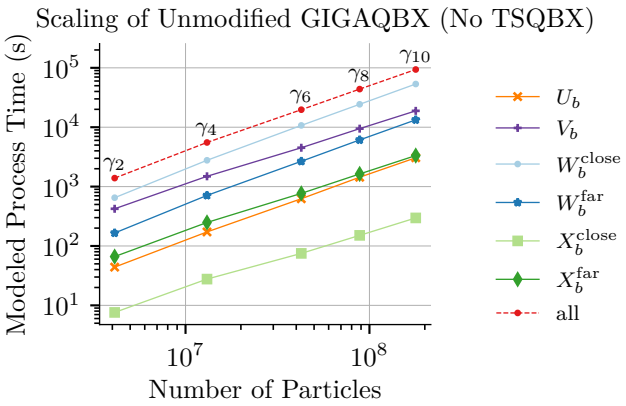


Figure 6: Scaling of the unmodified GIGAQBX algorithm (with spherical harmonic expansions instead of target-specific expansions) for evaluation of the Laplace single-layer potential on the sequence of 'urchin' geometries $\gamma_2, \gamma_4, \ldots, \gamma_{10}$.

$(5, 15)$. As reported in [32], this corresponds to about five digits of accuracy for evaluating Green's identity [22, Thm. 6.5], which we use as a proxy test for accuracy in layer potential evaluation. The upsampled quadrature rule has 295 nodes per element, and we use $t_f = 0.9$.

### 4.2.2 Cost Evaluation

In this section, we consider the cost of evaluation of the on-surface value of the Laplace single-layer potential on the 'urchin' geometry family. We establish a 'baseline' cost by modeling version of the GIGAQBX algorithm as described in [32] using the same framework from Section 4.1. The results are shown in Figure 6 (cf. [32, Fig. 12], which reports modeled floating point operations for the same geometries). The total reported time (under 'all') includes the time contribution from all stages of the algorithm. These results are obtained using the model in Section 4.1. To obtain this set of results, we chose a set of balancing parameters that minimized modeled process time as described in Section 4.1, which are $n_{\max} = 96$ and $n_{\text{mpole}} = 40$.

To assess the impact of each of the optimizations in this paper, we present results of three stages of cumulative optimizations. These are summarized in Figure 7.

**Using TSQBX Without Rebalancing.** The first optimization we consider is making use of target-specific QBX expansions, leaving all other algorithmic parameters constant. According to Section 4.1, assuming an average of one target per center, this should give a speedup of

$$c_{\text{p2qbxl}} \left(1 + p_{\text{qbx}}\right)^2 / \left(c_{\text{ts}}(1 + p_{\text{qbx}})\right)$$

in the evaluation of List 3 close, List 4 close, and List 1, while leaving the cost of the other stages of the algorithm unchanged. This turns out to be the case. The data in Table 3 predict a cost reduction by a factor of about 9 according to this formula. In Figure 7, a cost reduction of this magnitude is evident in $U_b$, $W_b^{\text{close}}$, and $X_b^{\text{close}}$ under the label 'ts'. On average, the overall time is 50% of the baseline.

**Rebalancing $n_{\max}$.** Next, we consider the effect of varying $n_{\max}$. Using $\gamma_6$ as a reference geometry, we conduct a study measuring total modeled process time against $n_{\max}$. The results are presented in Figure 8. Based on these results we choose $n_{\max} = 992$, which appears to be empirically near-optimal. The effect on the running time of the various stages is given under '$n_{\max}$' in Figure 7. On average, the running time is 38% of the baseline.

As we can see, the increase in $n_{\max}$ has a complex effect on the time for the different stages of the algorithm. Perhaps most easily explained, the time associated with List 2 decreases. This is consistent with a decrease in the overall number of boxes in the tree that may be expected with an increase in $n_{\max}$.

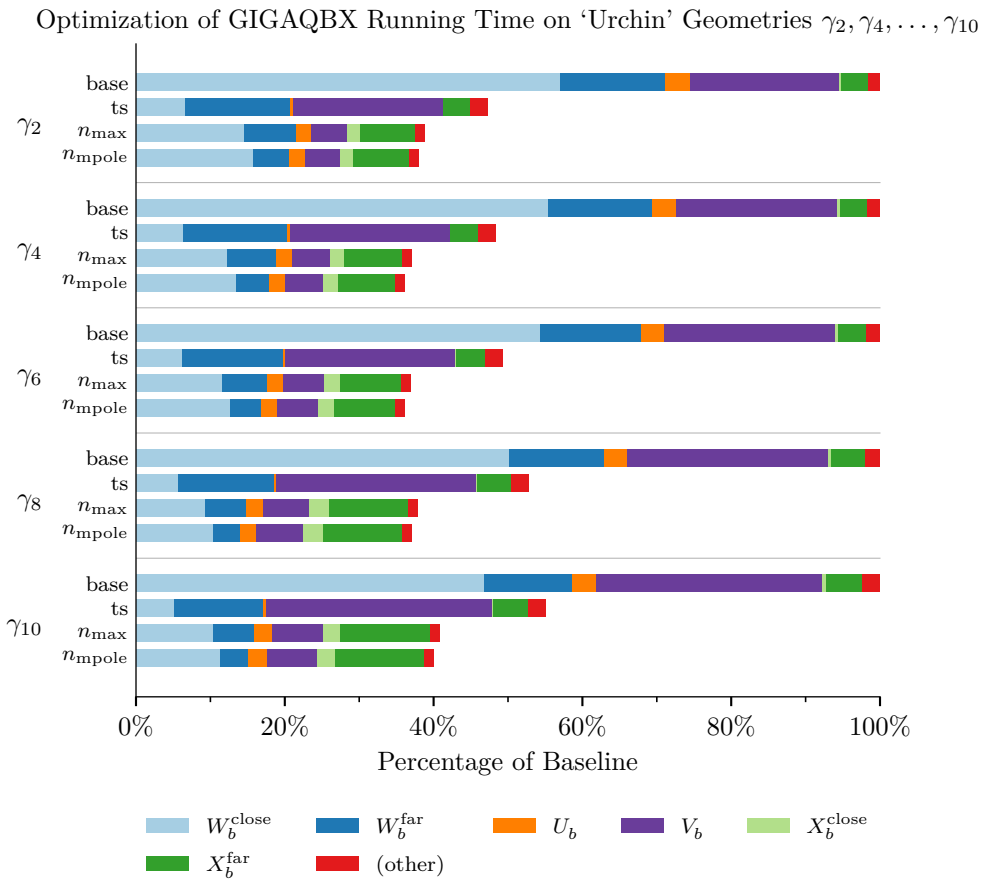


Figure 7: Cumulative impact of a sequence of optimizations, applied to the evaluation of the Laplace single-layer potential for the 'urchin' family of geometries, on modeled process time as well as time for individual stages. 'base' denotes the baseline time; 'ts' denotes the result of using target-specific expansions; '$n_{\max}$' denotes the result of rebalancing $n_{\max}$; and '$n_{\mathrm{mpole}}$' denotes the result of rebalancing $n_{\mathrm{mpole}}$.

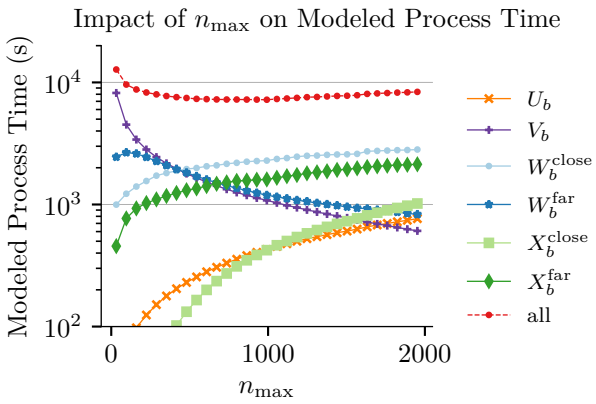


Figure 8: Modeled process time of the GIGAQBX algorithm and its various stages, versus $n_{\max}$, for the geometry $\gamma_6$, using target-specific QBX.

We also observe an increase in the amount of work for List 4 far. The amount of work for List 4 far is proportional to both $n_{\max}$ and the number of boxes (cf. [32, Lem. 15], Table 1). While the number of boxes decreases as $n_{\max}$ increases, the increase in $n_{\max}$ appears to have the dominant effect on the cost of this stage.

The proportion of time associated with target-specific expansions appears to increase roughly proportionally with $n_{\max}$. This is plausible as the cost estimates from previous work (cf. [32, Lem. 11, 13, 15], Table 1) show that at least a portion of the time in these stages is proportional to $n_{\max}$.

It is more difficult to explain why the time associated List 3 far decreases. However, as the decrease in the time associated with List 3 far closely mirrors that of List 2 (see Figure 8), it is likely to be related to the decrease in the number of boxes.

**Rebalancing $n_{\text{mpole}}$.** The last optimization we consider is the balancing of the constant $n_{\text{mpole}}$. Using an experiment similar to that of the one done to rebalance $n_{\max}$, we obtain a threshold value of $n_{\text{mpole}} = 280$.

The cost model can also be used to obtain this threshold. From the cost model, we expect approximately minimal cost when

$$n_{\text{mpole}} \approx \frac{c_{\text{m2qbxl}}}{c_{\text{ts}}} \frac{(1+p_{\text{fmm}})^3 + (1+p_{\text{qbx}})^2(1+p_{\text{fmm}}) + (1+p_{\text{qbx}})^3}{(1+p_{\text{qbx}})}.$$

Based on the data from Table 3, this value is about 291, so that the theoretical value and the empirically obtained value from the cost model are nearly in agreement.

The impact of increasing $n_{\text{mpole}}$ to reduce the amount of work done in $W_b^{\text{far}}$, shifting the corresponding work to $W_b^{\text{close}}$, for a reduction in the total modeled process time time to 37% of the original, on average. This can be seen in Figure 7 under the label '$n_{\text{mpole}}$'.

**Remarks.** Overall, the cumulative impact of these optimizations is to reduce the overall running time to an average of 37% of the original. As the percentage reduction in cost is nearly uniform across geometries, the scaling characteristics in geometry size are essentially unchanged from the baseline version in Figure 6, in that we observe an approximately linear scaling of the algorithm cost with the particle count.

From the relative costs in Figure 7, it is evident that this version of the algorithm is more evenly 'balanced' than the baseline, in that the proportion of time spent in individual stages is distributed more uniformly. Specifically, the cost contribution of List 3 close is now significantly closer to the other interaction lists. The dominant costs appear to be List 3 close and List 4 far.

### 4.2.3 An Example with Higher Accuracy and Simpler Geometry

To demonstrate the generality of the optimizations mentioned in this paper, we repeat the cost evaluation in Section 4.2.2 on a different, simpler geometry while choosing parameters to attain higher accuracy. The

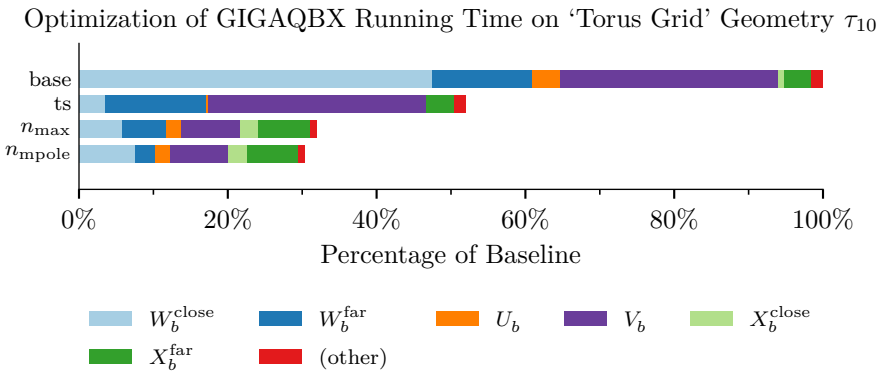


Figure 9: Cumulative impact of a sequence of optimizations, applied to the evaluation of the Laplace single-layer potential for the 'torus grid' geometry $\tau_{10}$, on modeled process time as well as time for individual stages.

geometry $\tau_1$ is a torus parametrized as the image of the set $(u, v) \in [0, 2\pi) \times [0, 2\pi)$ under the mapping

$$\begin{aligned} x &= \cos u\,(1 + 2\cos v) \\ y &= \cos u\,(1 + 2\cos v) \\ z &= 2\sin v. \end{aligned}$$

The 'torus grid' geometries $\tau_{2k}$, $k \in \mathbb{N}$, are obtained by spacing $2k$ copies of $\tau_1$ on a $2 \times k$ grid, with a uniform spacing of 0.6. See Figure 5 for a visual impression of $\tau_{10}$. To obtain the initial mesh of the torus, we tile the parameter domain into $40 \times 20$ contiguous rectangles, and then subdivide each rectangle into two triangles. We represent each triangle as the image of an triangular reference element under a polynomial mapping of degree 8, with 295 quadrature nodes per element, and use $t_f = 0.9$.

We focus on the evaluation of the Laplace single-layer potential on the geometry $\tau_{10}$. We use $(p_{\text{qbx}}, p_{\text{fmm}}) = (9, 20)$. From the initial mesh for $\tau_{10}$, refinement produces a mesh with 16,000 stage-1 elements and 64,000 stage-2 elements, for a total number of about 19 million source particles. Since the effects of closeness to touching would reflect in the count of stage-1 elements, the number of stage-1 elements $(40 \cdot 20 \cdot 2 \cdot 10)$ indicates that refinement due to closeness to touching of the different components is not necessary. A test of Green's identity with the QBX and FMM order parameters yields about eight digits of accuracy.

We optimize for evaluation on $\tau_{10}$. The modeled process time for the baseline GIGAQBX FMM, without using target-specific expansions, is minimized with a choice of $n_{\text{max}} = 96$ and $n_{\text{mpole}} = 40$. The baseline modeled process time is 18,968 seconds. We apply the sequence of optimizations mentioned in Section 4.2.2. The results are shown in Figure 9. The use of target-specific expansions reduces the modeled process time to 52% of the baseline, which is shown under the label 'ts'. Using an empirically determined value of $n_{\text{max}} = 928$ reduces the time to 32% of the baseline, which is shown under the label '$n_{\text{max}}$'. Finally, using an empirically determined value of $n_{\text{mpole}} = 420$ reduces the time to 30% of the baseline, which is shown under the label '$n_{\text{mpole}}$'.

In many ways, the cost characteristics of this example are similar to that of the examples in the previous section, although, due to the effects of higher order FMM and QBX expansions, the cost reduction is more significant. Like for the urchin, the dominant cost for the baseline example is in the near-field evaluations. After rebalancing, the dominant costs are List 3 close and List 4 far.

## 4.3 A Large-Scale BVP for the Helmholtz Equation

As a final demonstration of the broad applicability of the optimizations in this paper, we present a numerical example involving a large-scale boundary value problem with complex geometry. Thus far, we have only discussed the use of target-specific expansions for the Laplace equation, but everything we have stated in this paper has an analogue involving the Helmholtz kernel (see Appendix A). To demonstrate this, we solve the

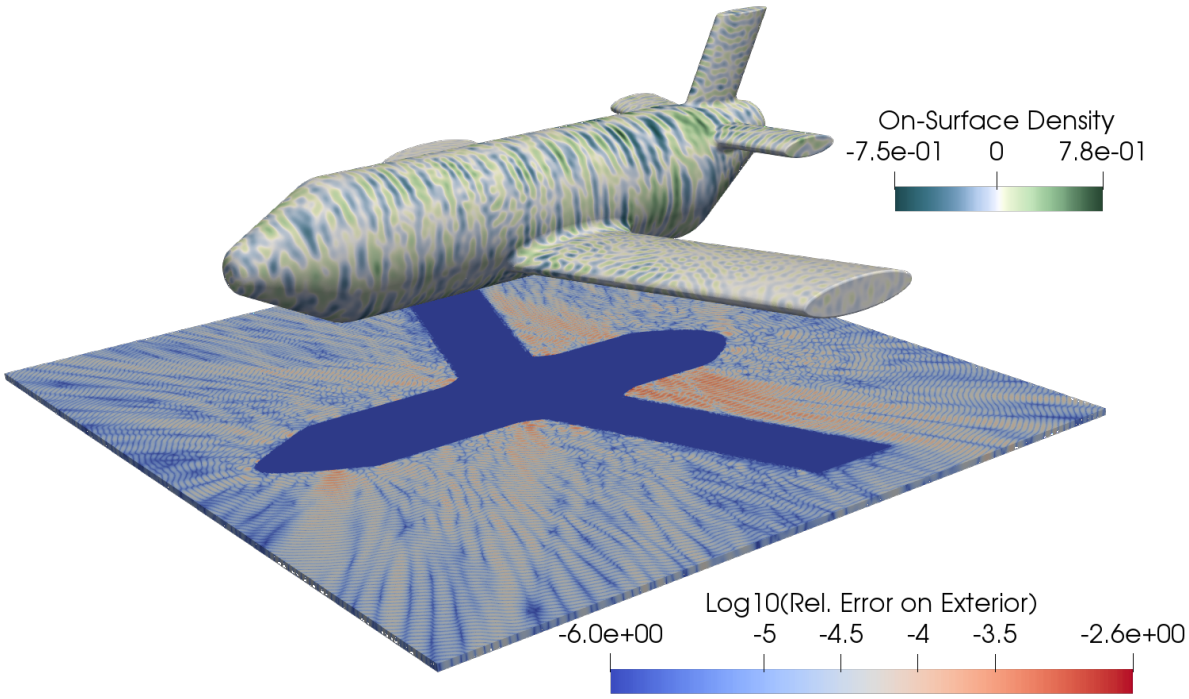


Figure 10: Visualization of the solution to an exterior Dirichlet problem for the Helmholtz equation solved on the 'plane' geometry. The surface geometry is shaded according to the real part of the solved-for density $\mu$. The slice positioned below the geometry in the visualization is taken from the level of the 'wing' of the plane and is shaded according to the logarithm of the observed relative error in the exterior of the volume. The maximum observed relative error anywhere in the exterior is about $2.5 \cdot 10^{-3}$.

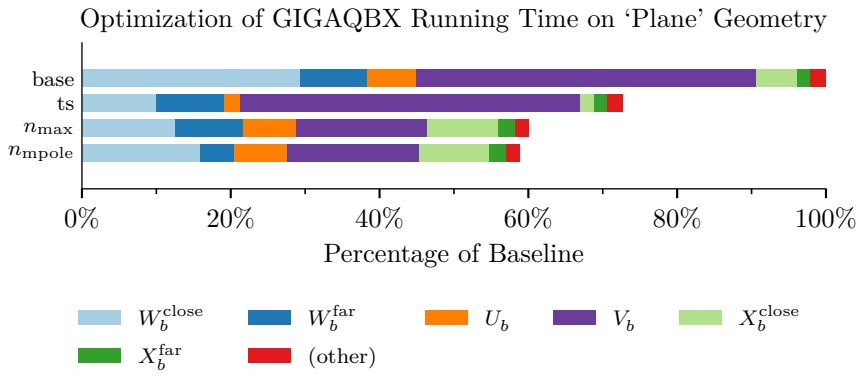


Figure 11: Cumulative impact of a sequence of optimizations, applied to the evaluation of the Helmholtz double-layer potential, $k = 20$, for the 'plane' geometry, on modeled process time as well as time for individual stages.

exterior Dirichlet problem for the Helmholtz equation

$$\begin{aligned}\left(\triangle + k^2\right) u(x) &= 0 \qquad && x \in \mathbb{R}^3 \setminus \Omega,\\ u(x) &= f(x) && x \in \Gamma,\\ \lim_{|x|\to\infty} |x| \left(\frac{\partial}{\partial |x|} - ik\right) u(x) &= 0,\end{aligned}$$

where $\Omega \subset \mathbb{R}^3$ is a closed, bounded region with smooth boundary $\Gamma = \partial\Omega$. The solution $u$ to the boundary value problem uses a Brakhage-Werner representation [3]

$$u \coloneqq i\alpha \mathcal{S}\mu - \mathcal{D}\mu,$$

where $\alpha \in \mathbb{R}$ and the double-layer operator $\mathcal{D}$ for a Green's function $\mathcal{K}$ is defined as

$$(\mathcal{D}\mu)(x) \coloneqq \int_\Gamma \frac{\partial \mathcal{K}(x,y)}{\partial \nu(y)} \mu(y)\, dS(y).$$

Our choice of geometry $\Omega$ is derived from `surface-3d/betterplane.brep` from [19]; this is the same source geometry as the example in [32, Sec. 6.2]. The surface mesh of the geometry consists of triangular elements of degree two mapping functions obtained with Gmsh [11]. The stage-1 discretization consists of 60,638 elements and the stage-2 discretization consists of 91,526 elements. We use 150 quadrature nodes per element, for a total of about 14 million source points, and $t_f = 0.9$.

The geometry, visualized in Figure 10, has a bounding box of size approximately $19 \times 20 \times 6.5$ units. We choose a Helmholtz parameter of $k = 20$. The reference solution is obtained using a point potential of sources places near the 'tail' of the geometry. We modify the quadrature scheme to use $L^2$-weighted degrees of freedom as a way to improve conditioning of the discrete problem [4].

Using the GIGAQBX FMM with target-specific expansions and near-optimal parameter values to drive the matrix-vector products in GMRES [28], our method attains a decrease in the residual norm by a factor of $10^{-6}$ in 120 iterations. We use $p_{\text{qbx}} = 4$ and we choose the FMM order to ensure a relative FMM error below $10^{-5}$. This entails a minimum FMM order of $p_{\text{fmm}} = 13$, but, as typical in an implementation of the Helmholtz FMM [15], also requires increasing $p_{\text{fmm}}$ with the box size to maintain accuracy tolerances. We use Helmholtz translation operators from FMMLIB3D [13, 12]. According to comparison with the reference solution using point evaluations in the exterior of the volume, the scheme obtains a relative $\ell^\infty$ error of about $2.5 \cdot 10^{-3}$.

We apply the optimization sequence described in Section 4.2.2 to obtain algorithmic parameters to minimize the modeled process time for the double-layer evaluation with target-specific expansions. The latter operator is the dominant cost of a GMRES iteration.

The parameter values that obtain near-optimal modeled cost for the 'baseline' version without target-specific expansions are $n_{\text{max}} = 100$ and $n_{\text{mpole}} = 40$. The baseline modeled process time of the double-layer operator is 28,900 seconds. Figure 11 shows the effect of the various optimizations on the modeled process time for the double-layer operator. Using target-specific expansions reduces the modeled process time to 73% of the baseline, shown under the label 'ts'. An empirically determined value of $n_{\text{max}} = 300$ reduces the time to 60% of the baseline, shown under the label '$n_{\text{max}}$'. Lastly, an empirically determined value of $n_{\text{mpole}} = 150$ reduces the time to 59% of the baseline, shown under the label '$n_{\text{mpole}}$'. The cost decreases in this example are smaller compared with the examples of Sections 4.2.2 and 4.2.3, since the QBX expansion order is lower and the cost of direct evaluations is not as significant in the baseline.

We also measured the 'wall time', or the actual duration from start to finish of the operator, using the baseline and optimized versions. The baseline wall time of the double-layer operator is about 2300 seconds, and the wall time for the single layer is about 1500 seconds. The sequence of optimizations above reduces the two times to about 1600 seconds and 900 seconds respectively. Hence, the use of target-specific expansions reduces the solve time for this problem to 66% of the baseline. Compared with process time, the wall time is not reduced as much because not every stage of our implementation is efficiently parallelized. This remains as future work.

# 5 Conclusion

This paper examines optimizations to reduce the cost of the GIGAQBX FMM in three dimensions. The main cost, as reported in [32] and verified in this paper, is typically due to the increased number of near-field interactions as compared with the 'point' version of the FMM. To reduce the impact of the cost of the near-field interactions, we consider an acceleration strategy using target-specific expansions. We develop a version of the GIGAQBX FMM that uses target-specific expansions for near-field evaluation. We also demonstrate a cost model for the GIGAQBX FMM that accurately estimates the total process time used on a shared-memory system. This model aids in an efficient assessment of the impact of various choices of algorithmic parameters affecting computational cost. We find that, in order to make the best use of target-specific expansions, algorithmic parameters should be modified to effectively shift more of the work onto target-specific near-field interactions. We demonstrate that this strategy produces speedups of 1.7–3.3× in practice on examples involving the Laplace and Helmholtz kernels using a variety of QBX and FMM orders.

A key strength of target-specific expansions is their general applicability. However, the requisite modification to the algorithm, including the mathematical derivation of target-specific expansions for specific kernels and the optimization of algorithmic parameters as done in this paper, requires manual intervention. A technical question remains as to what extent these tasks can be automated. One current subject of our investigation is the use of techniques from symbolic computing and/or numerical linear algebra to develop automated ways to generate optimal complexity target-specific expansions for various kernels. A second subject is a better combination of automation with the cost model in this paper in order to enable efficient selection of parameters, such as the maximum number of particles per box, for arbitrary input geometries.

# Acknowledgments

The authors' research was supported by the National Science Foundation under grants DMS-1418961 and DMS-1654756 as well as the Department of Computer Science at the University of Illinois at Urbana-Champaign. Any opinions, findings, and conclusions, or recommendations expressed in this article are those of the authors and do not necessarily reflect the views of the National Science Foundation; NSF has not approved or endorsed its content. Part of the work was performed while the authors were participating in the HKUST-ICERM workshop 'Integral Equation Methods, Fast Algorithms and Their Applications to Fluid Dynamics and Materials Science' held in 2017. The authors would also like to thank the anonymous reviewer for helpful comments.

# A Target-Specific Expansions

For completeness, we describe how to obtain target-specific expansions for the kernels of the single-layer potential $\mathcal{S}\mu$, its normal derivative $\mathcal{S}'\mu$, and the double-layer potential $\mathcal{D}\mu$ for a Green's function $\mathcal{K}$, where $\mathcal{K}$ is the Laplace or Helmholtz kernel:

$$
\begin{aligned}
(\mathcal{S}\mu)(x) &= \int_\Gamma \mathcal{K}(x,y)\mu(y)\,dS(y),\\
(\mathcal{S}'\mu)(x) &= \int_\Gamma \frac{\partial \mathcal{K}(x,y)}{\partial \nu(x)}\mu(y)\,dS(y),\\
(\mathcal{D}\mu)(x) &= \int_\Gamma \frac{\partial \mathcal{K}(x,y)}{\partial \nu(y)}\mu(y)\,dS(y).
\end{aligned}
$$

Consider a source $s \in \mathbb{R}^3$, center $c \in \mathbb{R}^3$, and target $t \in \mathbb{R}^3$ satisfying $|t-c| \le |s-c|$. Let $\gamma$ be the angle between $s-c$ and $t-c$ (Figure 3).

In the case of the Laplace kernel $\mathcal{G}$ (5), the $p$-th order target-specific expansion $\mathcal{G}^{(p)}$ takes the form

$$
\mathcal{G}^{(p)}(t,s) = \frac{1}{4\pi}\sum_{n=0}^{p} \frac{|t-c|^n}{|s-c|^{n+1}} P_n(\cos\gamma). \tag{19}
$$

To obtain the target-specific expansion of the normal derivative of the single-layer kernel, we require the gradient with respect to the target $t$. For the double-layer kernel, the gradient should be taken with respect to the source $s$. Recalling that $\cos\gamma = (t-c)^\top(s-c)/(|t-c||s-c|)$, we use the chain rule to obtain

$$\begin{aligned}\nabla_t \mathcal{G}^{(p)}(t,s) = \frac{1}{4\pi}\sum_{n=1}^{p}\frac{|t-c|^{n-1}}{|s-c|^{n+1}}\left(n\frac{t-c}{|t-c|}P_n(\cos\gamma)\right.\\ \left.+\left[\frac{s-c}{|s-c|}-\frac{t-c}{|t-c|}\cos\gamma\right]P_n'(\cos\gamma)\right),\end{aligned} \tag{20}$$

$$\begin{aligned}\nabla_s \mathcal{G}^{(p)}(t,s) = \frac{1}{4\pi}\sum_{n=0}^{p}\frac{|t-c|^{n}}{|s-c|^{n+2}}\left(-(n+1)\frac{s-c}{|s-c|}P_n(\cos\gamma)\right.\\ \left.+\left[\frac{t-c}{|t-c|}-\frac{s-c}{|s-c|}\cos\gamma\right]P_n'(\cos\gamma)\right).\end{aligned} \tag{21}$$

In the Helmholtz case, the kernel is

$$\mathcal{G}_k(t,s) = \frac{e^{ik|t-s|}}{4\pi|t-s|}.$$

For $|t| < |s|$, the addition for the Helmholtz kernel takes the form (see for instance [9, eq. (10.60.1, 10.60.2)]):

$$\mathcal{G}_k(t,s) = \frac{ik}{4\pi}\sum_{n=0}^{\infty}(2n+1)j_n(k|t|)h_n(k|s|)P_n(\cos\theta),$$

where the functions $j_n$ and $h_n$ are spherical Bessel and Hankel functions of the first kind (see [9, eq. (10.47.3)] and [9, eq. (10.47.5)]). This implies that the $p$-th order target-specific expansion of the Helmholtz kernel takes the form

$$\mathcal{G}_k^{(p)}(t,s) = \frac{ik}{4\pi}\sum_{n=0}^{p}(2n+1)j_n(k|t-c|)h_n(k|s-c|)P_n(\cos\gamma). \tag{22}$$

The gradients with respect to the target $t$ and the source $s$ are

$$\begin{aligned}\nabla_t \mathcal{G}_k^{(p)}(t,s) = \frac{ik}{4\pi}\sum_{n=0}^{p}(2n+1)\frac{h_n(k|s-c|)}{|t-c|}\left(k(t-c)j_n'(k|t-c|)P_n(\cos\gamma)\right.\\ \left.+\left[\frac{s-c}{|s-c|}-\frac{t-c}{|t-c|}\cos\gamma\right]j_n(k|t-c|)P_n'(\cos\gamma)\right),\end{aligned} \tag{23}$$

$$\begin{aligned}\nabla_s \mathcal{G}_k^{(p)}(t,s) = \frac{ik}{4\pi}\sum_{n=0}^{p}(2n+1)\frac{j_n(k|t-c|)}{|s-c|}\left(k(s-c)h_n'(k|s-c|)P_n(\cos\gamma)\right.\\ \left.+\left[\frac{t-c}{|t-c|}-\frac{s-c}{|s-c|}\cos\gamma\right]h_n(k|s-c|)P_n'(\cos\gamma)\right).\end{aligned} \tag{24}$$

Recurrences are available for computing the functions $P_n$, $j_n$, $h_n$, and their derivatives rapidly [9, Sec. 10.51, 14.10].

# B Software and Reproducibility

The results in this paper were obtained using **Pytential** [20], a Python package for the evaluation of layer potentials in two and three dimensions and the solution of related integral equations. We used revision `54360f5` (tagged `tsqbx`), available at `https://github.com/inducer/pytential`.

In addition, we have prepared a Docker image from which all experiments in this paper can be automatically reproduced with included software and scripts, available at `https://doi.org/10.5281/zenodo.3523410`. The code for the experiments is available at `https://doi.org/10.5281/zenodo.3542253`.

# References


[1] E. Agullo et al., *Modeling Irregular Kernels of Task-based codes: Illustration with the Fast Multipole Method*, Research Report RR-9036, INRIA Bordeaux, Feb. 2017, p. 35.

[2] C. R. Anderson, *An implementation of the fast multipole method without multipoles*, SIAM J. Sci. Statist. Comput. 13.4 (1992), pp. 923–947, DOI: 10.1137/0913055.

[3] H. Brakhage and P. Werner, *Über das Dirichletsche Außenraumproblem für die Helmholtzsche Schwingungsgleichung*, Archiv der Mathematik 16.1 (1965), pp. 325–329, DOI: 10.1007/BF01220037.

[4] J. Bremer, *On the Nyström discretization of integral equations on planar curves with corners*, Applied and Computational Harmonic Analysis (2011), DOI: 10.1016/j.acha.2011.03.002.

[5] J. Carrier, L. Greengard, and V. Rokhlin, *A fast adaptive multipole algorithm for particle simulations*, SIAM J. Sci. Statist. Comput. 9.4 (1988), pp. 669–686, DOI: 10.1137/0909044.

[6] A. Chandramowlishwaran et al., *Brief Announcement: Towards a Communication Optimal Fast Multipole Method and Its Implications at Exascale*, in: *Proceedings of the Twenty-fourth Annual ACM Symposium on Parallelism in Algorithms and Architectures*, SPAA '12, Pittsburgh, Pennsylvania, USA: ACM, 2012, pp. 182–184, ISBN: 978-1-4503-1213-4, DOI: 10.1145/2312005.2312039.

[7] H. Cheng, L. Greengard, and V. Rokhlin, *A fast adaptive multipole algorithm in three dimensions*, J. Comput. Phys. 155.2 (1999), pp. 468–498, DOI: 10.1006/jcph.1999.6355.

[8] J. Choi et al., *A CPU-GPU Hybrid Implementation and Model-Driven Scheduling of the Fast Multipole Method*, in: *Proceedings of Workshop on General Purpose Processing Using GPUs*, GPGPU-7, Salt Lake City, UT, USA: ACM, 2014, 64:64–64:71, ISBN: 978-1-4503-2766-4, DOI: 10.1145/2576779.2576787.

[9] *NIST Digital Library of Mathematical Functions*, https://dlmf.nist.gov/, Release 1.0.18 of 2018-03-27, F. W. J. Olver, A. B. Olde Daalhuis, D. W. Lozier, B. I. Schneider, R. F. Boisvert, C. W. Clark, B. R. Miller and B. V. Saunders, eds., URL: https://dlmf.nist.gov/.

[10] C. L. Epstein, L. Greengard, and A. Klöckner, *On the convergence of local expansions of layer potentials*, SIAM J. Numer. Anal. 51.5 (2013), pp. 2660–2679, DOI: 10.1137/120902859.

[11] C. Geuzaine and J.-F. Remacle, *Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities*, International Journal for Numerical Methods in Engineering 79.11 (2009), pp. 1309–1331, DOI: 10.1002/nme.2579.

[12] Z. Gimbutas and L. Greengard, *A fast and stable method for rotating spherical harmonic expansions*, Journal of Computational Physics 228.16 (Sept. 2009), pp. 5621–5627, DOI: 10.1016/j.jcp.2009.05.014.

[13] Z. Gimbutas and L. Greengard, *FMMLIB3D*, Retrieved at revision 339e93bbc, URL: https://github.com/zgimbutas/fmmlib3d.

[14] N. A. Gumerov and R. Duraiswami, *Comparison of the efficiency of translation operators used in the fast multipole method for the 3D Laplace equation*, tech. rep., University of Maryland Institute of Advanced Computer Studies, 2005.

[15] N. A. Gumerov and R. Duraiswami, *Fast Multipole Methods for the Helmholtz Equation in Three Dimensions*, Elsevier Series in Electromagnetism, Amsterdam: Elsevier Science, 2004, DOI: 10.1016/B978-0-08-044371-3.X5000-5.

[16] Y. C. Hu and S. L. Johnsson, *On the Accuracy of Poisson's Formula Based N-Body Algorithms*, tech. rep., Center for Research in Computing Technology, Harvard University, 1996.

[17] O. D. Kellogg, *Foundations of Potential Theory*, New York, NY: Dover Publications, 2000.

[18] L. af Klinteberg and A.-K. Tornberg, *Error estimation for quadrature by expansion in layer potential evaluation*, Adv. Comput. Math. 43.1 (2017), pp. 195–234, DOI: 10.1007/s10444-016-9484-x.

[19] A. Klöckner, *A Repository of Sample Geometries*, Retrieved at revision a869fc3ad, URL: https://github.com/inducer/geometries/.

[20] A. Klöckner and M. Wala, *Pytential: a software package for the evaluation of layer potentials*, Accessed October 2018, URL: https://github.com/inducer/pytential.

[21] A. Klöckner et al., *Quadrature by expansion: a new method for the evaluation of layer potentials*, J. Comput. Phys. 252 (2013), pp. 332–349, DOI: 10.1016/j.jcp.2013.06.027.

[22] R. Kress, *Linear integral equations*, Third edition, vol. 82, Applied Mathematical Sciences, Springer, New York, 2014, pp. xvi+412, ISBN: 978-1-4614-9593-2, DOI: 10.1007/978-1-4614-9593-2.

[23] K. Nabors et al., *Preconditioned, Adaptive, Multipole-Accelerated Iterative Methods for Three-Dimensional First-Kind Integral Equations of Potential Theory*, SIAM Journal on Scientific Computing 15.3 (1994), pp. 713–735, DOI: 10.1137/0915046.

[24] H. G. Petersen, E. R. Smith, and D. Soelvason, *Error estimates for the fast multipole method. II. The three-dimensional case*, en, Proc. R. Soc. Lond. A 448.1934 (Mar. 1995), pp. 401–418, DOI: 10.1098/rspa.1995.0024.

[25] H. Pouransari and E. Darve, *Optimizing the adaptive fast multipole method for fractal sets*, SIAM J. Sci. Comput. 37.2 (2015), A1040–A1066, DOI: 10.1137/140962681.

[26] M. Rachh, A. Klöckner, and M. O'Neil, *Fast algorithms for Quadrature by Expansion I: Globally valid expansions*, J. Comput. Phys. 345 (2017), pp. 706–731, DOI: 10.1016/j.jcp.2017.04.062.

[27] A. Rahimian, A. Barnett, and D. Zorin, *Ubiquitous evaluation of layer potentials using Quadrature by Kernel-Independent Expansion*, BIT Numerical Mathematics (Nov. 2017), DOI: 10.1007/s10543-017-0689-2.

[28] Y. Saad and M. H. Schultz, *GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems*, SIAM Journal on Scientific and Statistical Computing 7.3 (July 1986), pp. 856–869, DOI: 10.1137/0907058.

[29] M. Siegel and A.-K. Tornberg, *A local target specific quadrature by expansion method for evaluation of layer potentials in 3D*, Journal of Computational Physics 364 (2018), pp. 365–392, DOI: 10.1016/j.jcp.2018.03.006.

[30] J. K. Singer, *Parallel implementation of the fast multipole method with periodic boundary conditions*, East-West J. Numer. Math. 3.3 (1995), pp. 199–216.

[31] B. Vioreanu and V. Rokhlin, *Spectra of Multiplication Operators as a Numerical Tool*, SIAM Journal on Scientific Computing 36.1 (Jan. 2014), A267–A288, DOI: 10.1137/110860082.

[32] M. Wala and A. Klöckner, *A fast algorithm for Quadrature by Expansion in three dimensions*, Journal of Computational Physics 388 (2019), pp. 655–689, DOI: 10.1016/j.jcp.2019.03.024.

[33] M. Wala and A. Klöckner, *A fast algorithm with error bounds for Quadrature by Expansion*, Journal of Computational Physics 374 (2018), pp. 135–162, DOI: 10.1016/j.jcp.2018.05.006.

[34] H. Xiao and Z. Gimbutas, *A numerical algorithm for the construction of efficient quadrature rules in two and higher dimensions*, Computers & Mathematics with Applications 59.2 (Jan. 2010), pp. 663–676, DOI: 10.1016/j.camwa.2009.10.027.

[35] F. Zhao and S. L. Johnsson, *The parallel multipole method on the Connection Machine*, SIAM J. Sci. Statist. Comput. 12.6 (1991), pp. 1420–1437, DOI: 10.1137/0912077.