Adv Comput Math (2022) 48: 12
https://doi.org/10.1007/s10444-022-09927-5

An adaptive kernel-split quadrature method for parameter-dependent layer potentials

Fredrik Fryklund[1] · Ludvig af Klinteberg[1] · Anna-Karin Tornberg[1]

Received: 2 August 2021 / Accepted: 18 January 2022 / Published online: 9 March 2022

Abstract
Panel-based, kernel-split quadrature is currently one of the most efficient methods available for accurate evaluation of singular and nearly singular layer potentials in two dimensions. However, it can fail completely for the layer potentials belonging to the modified Helmholtz, modified biharmonic, and modified Stokes equations. These equations depend on a parameter, denoted α, and kernel-split quadrature loses its accuracy rapidly when this parameter grows beyond a certain threshold. This paper describes an algorithm that remedies this problem, using per-target adaptive sampling of the source geometry. The refinement is carried out through recursive bisection, with a carefully selected rule set. This maintains accuracy for a wide range of the parameter α, at an increased cost that scales as $\log \alpha$. Using this algorithm allows kernel-split quadrature to be both accurate and efficient for a much wider range of problems than previously possible.

Keywords Integral equations · Partial differential equations · Layer potentials · Modified Helmholtz equation · Modified Stokes equation

Mathematics Subject Classification (2010) 65D05 · 65D30 · 65R20 · 45D99

1 Introduction

This paper presents an extension of the panel-based, kernel-split quadrature scheme by Helsing and Ojala [13], for evaluating singular and nearly singular layer potentials

Communicated by: Michael O'Neil

This article belongs to the Topical Collection: *Advances in Computational Integral Equations*
Guest Editors: Stephanie Chaillat, Adrianna Gillman, Per-Gunnar Martinsson, Michael O'Neil, Mary-Catherine Kropinski, Timo Betcke, Alex Barnett

✉ Fredrik Fryklund
ffry@kth.se

[1] Department of Mathematics, KTH Royal Institute of Technology, Stockholm, Sweden

in two dimensions. It is one of the current state of the art methods for maintaining low errors when solving homogeneous elliptic partial differential equations (PDEs) in two dimensions using integral equation methods [11, 13, 17]. However, there exists a set of problems for which this scheme can fail completely. This includes the following PDEs in $\mathbb{R}^2$:

$$(\Delta - \alpha^2)u = 0, \qquad \text{modified Helmholtz,} \tag{1}$$
$$\Delta(\Delta - \alpha^2)u = 0, \qquad \text{modified biharmonic,} \tag{2}$$
$$(\Delta - \alpha^2)u - \nabla p = 0, \qquad \text{modified Stokes (subject to } \nabla \cdot u = 0\text{),} \tag{3}$$

where α is a positive real number. For brevity, we refer to them as the *modified* PDEs. Note that they are not consistently named in the literature. For example, the modified Helmholtz equation is also known as the screened Poisson equation, the Yukawa equation, the linearized Poisson-Boltzmann equation, and the Debye-Hückel equation. Meanwhile, the modified Stokes equations are also known as the Brinkman equations. These PDEs appear in many different applications: electrostatic interactions in protein and related biological functions, macroscopic electrostatics, and fluid flow on the microscale, to mention a few [7, 12, 14, 15, 19, 20].

A common trait of the modified PDEs is that their associated layer potentials have kernels that either decay exponentially or have components that decay exponentially, with a rate that is proportional to α. This decay presents a problem for the abovementioned kernel-split quadrature. In short, the quadrature method is based on writing the kernel on a form with smooth functions multiplying explicit singularities, and then integrating each term separately. In order to be accurate, these smooth functions have to be locally well approximated by polynomials, which is increasingly difficult for larger values of α.

Large values of α are of interest in the context of elliptic marching. In elliptic marching, a semi-implicit temporal discretization is applied to the governing equations. A time-step then involves solving a sequence of elliptic equations, such as the modified PDEs: e.g., the heat equation, the time-dependent Stokes and Navier-Stokes equations correspond to the modified Helmholtz equation, the modified biharmonic equation, and the modified Stokes equations, respectively [3, 6, 8, 9]. Regardless of the specifics of the time-discretization scheme, the resulting equations involve the parameter α, where α^2 is inversely proportional to the time-step size. The smaller the time-step, the lower the temporal error, but the spatial problem becomes harder to solve accurately. This is a new problem in the context of elliptic marching, combined with boundary integral methods applied to the resulting modified PDEs. Earlier, high-order and accurate schemes were not available; thus, small time-steps were redundant since the spatial error dominated, and the low-error regimes were not feasible to explore on standard desktop computers. Today, efficient methods are available that give higher precision and allow for more complicated problems. It is evident that the restriction of the time-step is a significant bottleneck, as the kernel-split quadrature may fail for high temporal resolutions.

We have developed a robust quadrature scheme, based on adaptive refinement, that maintains high accuracy for any α, without sacrificing efficiency. It applies to target points both on and close to the boundary, where regular quadrature is insufficient.

In this context, refinement refers simply to an interpolation of known quantities to a locally refined discretization, as opposed to increasing the number of degrees of freedom in the discretized integral equation. The additional cost, in terms of assembly time per panel, scales as $\mathcal{O}(\log \alpha)$.

The remainder of this paper is organized as follows. In Section 2, we give an outline of the kernel-split quadrature. Section 3 describes the problem that we are trying to solve, using the modified Helmholtz equation as an example. In Section 4, we present error analysis, again with the modified Helmholtz equation as template. The new algorithm we propose is presented in Section 5, followed by numerical results in Section 6.

2 Background

Our goal is to evaluate layer potentials in $\mathbb{R}^2$ of the form

$$u(x) = \int_{\partial\Omega} G(x, y)\sigma(y)\,\mathrm{d}S(y). \tag{4}$$

The layer density $\sigma(y)$ is assumed to be smooth, implying that the boundary is assumed to be smooth as well. The kernel $G(x, y)$ is singular at $x = y$ and can be expressed with explicit singularities as

$$G(x, y) = G^S(x, y) + G^L(x, y)\log\left|y - x\right| + G^C(x, y)\frac{(y - x)\cdot \hat{n}(y)}{\left|y - x\right|^2}, \tag{5}$$

where G^S is a smooth function. The functions G^L and G^C are smooth functions that multiply a log-type singularity and a Cauchy-type singularity, respectively. We refer to this decomposition as *kernel-split.*

The boundary $\partial\Omega$ is discretized using a composite Gauss–Legendre quadrature. It is subdivided into intervals Γ_i, denoted *panels*,

$$\partial\Omega = \bigcup_i \Gamma_i. \tag{6}$$

Each panel Γ_i is described by a parametrization γ_i,

$$\Gamma_i = \left\{\gamma_i(t) \in \mathbb{R}^2 | \quad t \in [-1, 1]\right\}. \tag{7}$$

We refer to x in (4) as a *target point* and a point y as a *source point*. A panel to which a source point belongs is referred to as a *source panel*. Associated with the parametrization is a speed function $s_i(t) = \left|\gamma_i'(t)\right|$, a normal vector $\hat{n}_i(t)$, and the curvature $\kappa_i(t)$. Introducing the convenience notation $\sigma_i(t) = \sigma(\gamma_i(t))$, the layer potential from a panel Γ_i becomes

$$\int_{\Gamma_i} G(x, y)\sigma(y)\,\mathrm{d}S(y) = \int_{-1}^{1} G\big(x, \gamma_i(t)\big)\sigma_i(t)s_i(t)\,\mathrm{d}t. \tag{8}$$

Each panel is discretized in the parametrization variable t using the nodes and weights (t_j^G, λ_j^G) of an n-point Gauss–Legendre quadrature rule on the canonical interval $[-1,\ 1]$, which is of order $2n$, such that on each panel we have the discrete quantities

$$y_{ij} = \gamma_i(t_j^G), \tag{9}$$

$$\hat{n}_{ij} = \hat{n}_i(t_j^G), \tag{10}$$

$$\sigma_{ij} = \sigma\Big(\gamma_i(t_j^G)\Big), \tag{11}$$

$$s_{ij} = \Big|\gamma_i'(t_j^G)\Big|, \tag{12}$$

$$\kappa_{ij} = \kappa_i(t_j^G). \tag{13}$$

Omitting the panel index i, the layer potential contribution from a panel Γ is then computed using the approximation

$$\int_\Gamma G(x,y)\sigma(y)\,\mathrm{d}S(y) \approx \sum_{j=1}^{n} G\big(x,y_j\big)\sigma_j s_j \lambda_j^G. \tag{14}$$

Due to the singularities in G, the above formula requires x to be *well-separated* from Γ in order to be accurate (see Section 5, paragraph Near evaluation criterion). Otherwise, the scheme of [11] is used, known as *product integration*. With that, target-specific quadrature weights w^L and w^C of order n are computed for the known singularities where needed, such that

$$\int_\Gamma f(x,y)\log\big|y-x\big|\,\mathrm{d}S(y) \approx \sum_{j=1}^{n} f(x,y_j)w_j^L(x), \tag{15}$$

$$\int_\Gamma f(x,y)\frac{(y-x)\cdot\hat{n}(y)}{\big|y-x\big|^2}\,\mathrm{d}S(y) \approx \sum_{j=1}^{n} f(x,y_j)w_j^C(x). \tag{16}$$

Substituting (5) into (4) and applying the above product integration give the so-called *kernel-split quadrature scheme*. Depending on the location of the target point x relative to the source panel Γ, the evaluation can be divided up into three different cases:

1. **Singular, with self-interaction.** If $x \in \Gamma$ is one of the quadrature nodes, $x = y_i$, then the term multiplying G^C is smooth, with the limit

$$\lim_{\substack{x\to y\\ x,y\in\partial\Omega}} \frac{(y-x)\cdot\hat{n}(y)}{\big|y-x\big|^2} = -\frac{\kappa(y)}{2}, \tag{17}$$

where $\kappa(y)$ is the curvature of $\partial\Omega$ at y. Applying product integration to the G^L term, we get

$$\begin{aligned}\textstyle\int_\Gamma G(y_i,y)\sigma(y)\,\mathrm{d}S(y) \approx & \sum_{\substack{j=1\\ j\neq i}}^{n}\Big[G(y_i,y_j)\sigma_j s_j\lambda_j^G + G^L(y_i,y_j)\sigma_j\Big(w_j^L(y_i) - s_j\log\big|y_j-y_i\big|\lambda_j^G\Big)\Big] \\ & + G^S(y_i,y_i) + G^L(y_i,y_i)\sigma_i w_i^L(y_i) - G^C(y_i,y_i)\tfrac{\kappa(y_i)}{2}.\end{aligned} \tag{18}$$

Note that this is the only case where G^S is evaluated explicitly, and only in one point.

2. **Singular, without self-interaction.** If $x \in \partial\Omega$ is either on the source panel Γ but not a quadrature node, or on a neighboring panel, then the G^C term is still smooth, and we do not have to take the limit at $x \to y$ into account. This lets us simplify the above to

$$\int_\Gamma G(x,y)\sigma(y)\,\mathrm{d}S(y) \approx \sum_{j=1}^{n} \left[G(x,y_j)\sigma_j s_j \lambda_j^G + G^L(x,y_j)\sigma_j \left(w_j^L(x) - s_j \log\left|y_j - x\right| \lambda_j^G \right) \right]. \quad (19)$$

3. **Nearly singular case.** If x is close to Γ, but not on a neighboring panel, then we need to address both singularities in (5). This case occurs when either $x \in \Omega$ is close to $\partial\Omega$, or when $x \in \partial\Omega$ is on a section of the boundary that is distant in arc length or disjoint. The layer potential is then evaluated as

$$\begin{aligned} \int_\Gamma G(x,y)\sigma(y)\,\mathrm{d}S(y) \approx \sum_{j=1}^{n} \Bigg[& G(x,y_j)\sigma_j s_j \lambda_j^G \\ & + G^L(x,y_j)\sigma_j \left(w_j^L(x) - s_j \log\left|y_j - x\right| \lambda_j^G \right) \\ & + G^C(x,y_j)\sigma_j \left(w_j^C(x) - s_j \frac{(y_j - x)\cdot \hat{n}(y_j)}{\left|y_j - x\right|^2} \lambda_j^G \right) \Bigg]. \end{aligned} \quad (20)$$

Given a target point x, the panels on $\partial\Omega$ are partitioned into two sets: *far* panels, that can be evaluated directly using (14), and *near* panels, that must be evaluated (or corrected), using either (18), (19), or (20).

3 Problem statement

The kernel-split quadrature scheme outlined above is both efficient and accurate when applied to the single- and double-layer potentials of several PDEs, such as the Laplace, Helmholtz, and Stokes equations [11, 13, 17] on geometries with a smooth boundary. However, for the modified PDEs (1), (2), and (3), the scheme can fail completely. All of these equations have layer potential kernels including second-kind modified Bessel functions, of the forms $K_0(\alpha|y-x|)$ and/or $K_1(\alpha|y-x|)$. As we shall see, this is problematic for the kernel-split quadrature.

As an illustrating example, we study the single-layer kernel of the modified Helmholtz equation (1). The associated Green function is

$$G(x,y) = K_0(\alpha|y-x|), \quad (21)$$

where the scaling factor $1/2\pi$ has been omitted. The split of this kernel is based on using a standard decomposition [16, §10.31] to explicitly write out the singularities in K_0,

$$K_0(\rho) = K_0^S(\rho) - I_0(\rho)\log\rho, \quad \rho \in \mathbb{R}^+, \quad (22)$$

where K_0^S is the smooth remainder, and I_0 is the modified Bessel function of the first kind. For future reference, we have by [16, §10.25,§10.29] that

$$I_\nu(\rho) = \left(\frac{1}{2}\rho\right)^\nu \sum_{k=0}^{\infty} \frac{\left(\frac{1}{4}\rho^2\right)^k}{k!(\nu+k)!}, \quad \nu \in \mathbb{N},\ \rho \in \mathbb{R}, \tag{23}$$

and the derivatives of I_ν can be expressed as

$$I_\nu^{(n)}(\rho) = 2^{-n} \sum_{m=0}^{n} \binom{n}{m} I_{2m+(\nu-n)}(\rho), \quad \nu,\ n \in \mathbb{N},\ \rho \in \mathbb{R}, \tag{24}$$

which makes them easy to evaluate using standard numerical libraries,

Inserting (22) into the kernel (21), after also splitting the logarithm, we identify the terms in (5) as

$$G^S(x, y) = K_0^S(\alpha|y-x|) - I_0(\alpha|y-x|)\log\alpha, \tag{25}$$

$$G^L(x, y) = -I_0(\alpha|y-x|), \tag{26}$$

$$G^C(x, y) = 0. \tag{27}$$

Product integration is a semi-analytical method. This means that the target-specific quadrature weights in (15) are found by an analytic treatment of the singularity or near singularity. The accuracy of the method for a given evaluation point x is basically limited by how well a function $f(x, y)$ can be resolved by an $(n-1)$th-degree Legendre polynomial. For a well-resolved function, the integration error is usually only at most a few orders of magnitude larger than round-off, independent of the location of the evaluation point. We now want to evaluate the layer potential (4) with the layer density σ, using the split above. With this G^L, $f(x, y)$ in (15) will be $-I_0(\alpha|y-x|)\sigma(y)$. Hence, for good accuracy in integrating the logarithmic singularity, it is this product that must have a small interpolation error.

The function $G(x, y)$ goes to zero as $|y-x|$ goes to infinity, since $K_\nu(\rho) \sim \sqrt{\pi/(2\rho)}e^{-\rho}$ as $\rho \to \infty$ for $\nu = 1, 2$, but its components $G^S(x, y)$ and $G^L(x, y)$ actually grow exponentially as $e^{\alpha|y-x|}$, with opposing signs, following the asymptotic $I_\nu(\rho) \sim e^\rho/\sqrt{2\pi\rho}$ as $\rho \to \infty$ for $\nu = 1, 2$ [16, §10.25,§10.30]. As α gets larger, this makes G^L an increasingly bad candidate for polynomial interpolation, which will render the product integration inaccurate. In addition, when evaluated in limited precision, this split is prone to numerical cancellation, due to the limiting forms of K_0 and I_1. We will argue that the error mainly is a function of the quantity αh, which can be heuristically motivated by Fig. 1. To ensure that this error remains below some tolerance ϵ, we therefore suggest that α and panel length h must satisfy a criterion on the form

$$\alpha h \le C_\varepsilon, \tag{28}$$

for some constant C_ε, which can be determined using numerical experiments. The above criterion can also be reformulated as follows: In order to achieve a tolerance ϵ, panel lengths must satisfy

$$h \le h_{\max} := C_\varepsilon/\alpha. \tag{29}$$

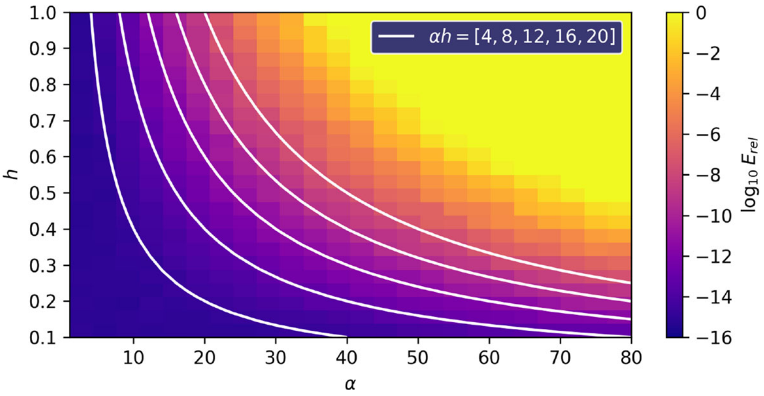


Fig. 1 Relative error $E_{\text{rel}} = \|u - \tilde{u}\|_\infty / \|u\|_\infty$, where $\tilde{u}$ is the result of evaluating the modified Helmholtz single layer potential (30) over a flat panel of length h, using kernel-split quadrature with $n = 32$ Gauss–Legendre points. The norm is taken over 100 values of target points x randomly drawn from the box $[-h/2, h/2] \times [0, h/2]$. The white lines are contours of the quantity αh, providing a heuristic motivation for why a criterion of the form (28) is suitable

The naive way of achieving this, for a given α, is to discretize $\partial\Omega$ using sufficiently short panels. However, this can result in a discretization with orders of magnitude more points than necessary to resolve the geometry and the layer density. Global refinement is also redundant for another reason: as stated above the functions $K_0(\rho)$ and $K_1(\rho)$ decay as $\sqrt{\pi/(2\rho)}e^{-\rho}$, meaning that they are very localized for large α, i.e., when polynomial interpolation might fail, and are almost zero in finite precision away from $\rho = 0$. Thus, only a small portion of the boundary needs refinement.

4 Error estimates

As was indicated above, there are two main sources of errors in the kernel split quadrature that grow as α increases. The first part arises from an interpolation error, and the second is due to numerical cancellation. In order to better understand these errors, we will perform a limited analysis for the single-layer potential of the modified Helmholtz equation on a flat panel in Section 4.1.

This understanding of the error structure is useful also for the associated kernels for the other modified PDEs since they have a similar singularity structure to the single-layer modified Helmholtz potential; see Appendix 2. They are all combinations of K_0 and/or K_1; thus, in the form of explicit singularities, they contain I_0 and/or I_1 multiplying a log-type singularity.

Based on our simple analysis in Section 4.1, we will in Section 4.2 formulate an error estimate and illustrate its performance for the single-layer modified Helmholtz potential. The error estimates can be used to set C_ϵ in (28), as discussed in Section 4.4. Other kernels are discussed in Section 4.3.

In this section as well as the next, we will identify vectors $x, y \in \mathbb{R}^2$ with the corresponding complex numbers $x, y \in \mathbb{C}$.

4.1 Error analysis for a flat panel

We consider the single-layer potential for the modified Helmholtz equation with unit density, evaluated using kernel-split quadrature from a flat panel of length h,

$$u(x) = \int_{-h/2}^{h/2} K_0(x, y)\,\mathrm{d}y. \tag{30}$$

To evaluate (30) using the kernel-split correction (20), we first write

$$u(x) \approx \sum_{j=1}^{n} \Big(G(x, y_j) - G^L(x, y_j)\log\big|y_j - x\big|\Big)\lambda_j - \int_{-h/2}^{h/2} I_0(\alpha|y - x|)\log|y - x|\,\mathrm{d}y, \tag{31}$$

using $G^C(x, y) = 0$ (27). In order to evaluate the remaining integral, the function $I_0(\alpha|y - x|)$ is approximated by an $(n - 1)$th-degree polynomial p_{n-1} that interpolates $I_0(\alpha|y - x|)$ at the n nodes y_j,

$$I_0(\alpha|y - x|) = p_{n-1}(x, y) + r_n(x, y), \tag{32}$$

$$p_{n-1}(x, y) = \sum_{k=0}^{n-1} c_k(x) y^k, \tag{33}$$

$$p_{n-1}(x, y_j) = I_0(\alpha|y_j - x|), \quad j = 1, \ldots, n, \tag{34}$$

where r_n is the polynomial interpolation error. We consider only even values for n. The integral on the right-hand side of (31) can be written as a sum: the approximation by integrating the polynomial p_{n-1}, and an integral over the polynomial interpolation error

$$\int_{-h/2}^{h/2} I_0(\alpha|y - x|)\log|y - x|\,\mathrm{d}y = \mathrm{Re}\left(\int_{-h/2}^{h/2} I_0(\alpha|y - x|)\log(y - x)\,\mathrm{d}y\right) \tag{35}$$

$$= \mathrm{Re}\left(\sum_{k=0}^{n-1} c_k(x) q_k(x)\right) + \mathrm{Re}\left(\int_{-h/2}^{h/2} r_n(x, y)\log(y - x)\,\mathrm{d}y\right). \tag{36}$$

Here, we have used the definition of the complex logarithm. We have that

$$q_k(x) = \int_{-h/2}^{h/2} y^k \log(y - x)\,\mathrm{d}y, \quad k = 0, \ldots, n - 1, \tag{37}$$

are integrals that can be computed recursively, starting from exact formulas [10].

The error in the kernel-split quadrature has two sources. The first is the integral over the interpolation error from the integral in the right-hand side of (31), namely

$$R_n(\alpha, x, h) = \int_{-h/2}^{h/2} r_n(x, y)\log|y - x|\,\mathrm{d}y = \mathrm{Re}\left(\int_{-h/2}^{h/2} r_n(x, y)\log(y - x)\,\mathrm{d}y\right). \tag{38}$$

For brevity, we also refer to the integral over the interpolation error as the interpolation error. It will be clear from context which one that is referred to.

The second is a cancellation error due to the limiting forms of G and G^L. Recall that $G^S = G - G^L$; this function grows exponentially as $-e^{\alpha|y-x|}/\sqrt{2\pi\alpha|y-x|}$, which is the same asymptotics as for G^L, but with opposite sign. The function G is a decaying function as $\alpha|y-x|$ goes to infinity; thus, G^S and G^L have to cancel. As the quadrature terms in (31) grow in magnitude, catastrophic cancellation errors follow.

We will now perform the analysis of the interpolation error for target points x along the real axis. In Section 4.2, we will consider the performance of this estimate for target points in the plane.

4.1.1 Interpolation error

Our goal is to estimate the magnitude of the interpolation error $R_n(\alpha, x, h)$ in (38), and we will do so for $x \in \mathbb{R}$. The polynomial p_{n-1} (33) is formed by interpolating at the Legendre nodes, $y_j = t_j^G h/2$. According to [18], the remainder is given by

$$r_n(x, y) = \prod_{j=1}^{n}(y - t_j^G h/2)\frac{1}{n!}\frac{\mathrm{d}^n}{\mathrm{d}\xi^n} I_0\big(\alpha(\xi - x)\big), \quad y, \xi \in [-h/2, h/2], \tag{39}$$

where the absolute value of the argument of I_0 can be removed, since I_0 is an even function. This is convenient, as it simplifies differentiation.

The n Legendre nodes are the roots of the Legendre polynomial of order n, denoted P_n, orthogonal on $[-h/2, h/2]$. Hence

$$\ell_n \prod_{j=1}^{n}(y - t_j^G h/2) = P_n(2y/h), \quad y \in [-h/2, h/2], \tag{40}$$

where ℓ_n is the coefficient of the leading order monomial term in P_n. From Rodrigues' formula [16, §18.5], it can be shown that $\ell_n = 2^{-n}(2n)!/(n!)^2$. After rescaling ξ, we have

$$r_n(x, y) = P_n(2y/h)\frac{(\alpha h)^n n!}{(2n)!} I_0^{(n)}\big(\alpha(\xi h/2 - x)\big) \quad \xi \in [-1, 1]. \tag{41}$$

To present an error bound for (41), we can apply

$$0 \le I_0^{(n)}\big(\alpha(\xi h/2 - x)\big) \le I_0^{(n)}\big(\alpha(h/2 + |x|)\big), \quad \forall \xi \in [-1, 1]. \tag{42}$$

However, the resulting error bound greatly overestimates the error, and thus is not of practical use. We pursue an alternative approach by estimating r_n by choosing a value for ξ for all pairs (x, y). For this purpose, heuristics suggest $\xi = 0$, as demonstrated below.

Combining (38) and (41) for $\xi = 0$, we get the estimate

$$\left|R_n(\alpha, x, h)\right| \approx \left|\mathrm{Re}\left(\Lambda_S(\alpha, n, x, h)\right)\right| \le \left|\Lambda_S(\alpha, n, x, h)\right|, \tag{43}$$

with Λ_S defined as

$$\Lambda_S(\alpha, n, x, h) = \tfrac{n!}{(2n)!}(\alpha h)^n I_0^{(n)}\left(\alpha|x|\right)\tfrac{h}{2} g_n\left(2x/h\right), \tag{44}$$

where the subscript S denotes single-layer potential. Note that $I_0^{(n)}$ is an even function for even values of n; by the relation (24) $I_0^{(n)}$ is a linear combination of I_ν with even ν, and it follows from the definition (23) that these terms are even. Thus, we may take the absolute value of the argument of $I_0^{(n)}$ without altering the result. This is important, as later on we will consider complex target points x.

In (44), we have

$$g_n(\tilde{x}) = \int_{-1}^{1} P_n(\rho)\log(\rho - \tilde{x})\,\mathrm{d}\rho \tag{45}$$

obtained by

$$\int_{-1}^{1} P_n(\rho)\log(\rho h/2 - x)\,\mathrm{d}\rho = \int_{-1}^{1} P_n(\rho)\log(\rho - 2x/h)\,\mathrm{d}\rho = g_n(2x/h), \tag{46}$$

where the integral is rewritten using $\log(h/2(\rho-2x/h)) = \log(h/2)+\log(\rho-2x/h)$ and the fact that each $P_n(\rho)$ integrates to zero over the interval.

This “Legendre-log integral” can be evaluated using the recursion formulas in Appendix 1.

For $h = 2$, both the absolute value as well as the real part of Λ_S in (44) are shown as a function of x in Fig. 2. The absolute value of Λ_S is a smooth function that achieves its maximum for x close to the endpoints of the interval. This is due to the log-factor in the integrand being singular for target points x in $[-1, 1]$, combined with the growth of the factor I_0 towards the edges of the panel. For target points x outside the source panel, the Legendre-log integral g_n decreases rapidly.

This plot is only illustrating the error estimate derived for this simplified case. How well it actually estimates the error will be discussed after we have considered also the second part of the error.

4.1.2 Numerical cancellation error

In addition to the interpolation error just discussed, the kernel-split (31) can also suffer from numerical cancellation when α is large. To see this, note that $G(x,y) - G^L(x,y)\log(\alpha|y-x|) = G^S(x,y)$, which has the asymptotics $-e^{\alpha|y-x|}/\sqrt{2\pi\alpha|y-x|}$ as $\alpha|y-x|$ goes to infinity. This is the same asymptotics as

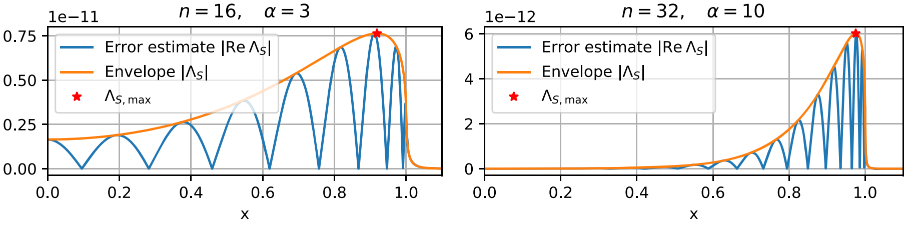


Fig. 2 Plot of the error estimate (43) with (n, α) valued (16, 3) and (32, 10), and $h = 2$. The estimate $|\operatorname{Re}\Lambda_S|$ oscillates in x, and is enveloped by the smooth upper bound $|\Lambda_S|$, which has a maximum to the left of $x = 1$

for I_0, but with an opposing sign. These two terms have to cancel, since K_0, i.e., G, is a decaying function. The sum of terms with a large magnitude, but opposing signs, is prone to numerical cancellation in limited precision.

The cancellation error is straightforward to estimate by

$$\Xi_S(\alpha, x, h) = \epsilon_{\text{mach}} h I_0(\alpha d), \quad d = \max_{y\in[-h/2,h/2]} |y - x| = h/2 + |x|, \tag{47}$$

where ϵ_{mach} is the machine epsilon.

4.2 Error estimate

We now have an error estimate $|\Lambda_S(\alpha, n, x, h)|$ (44) for the interpolation error $R_n(\alpha, x, h)$ (38), and an estimate $\Xi_S(\alpha, x, h)$ (47) for the cancellation error. Both were derived for target points along the real axis, but we now evaluate the estimated maximum error over a set D of discrete target points in the plane, and compare it to the maximum measured actual error. There are two goals: to study the errors' and error estimates' dependence on αh, and to construct a combination of the error estimates to predict the total error.

Let D be the set 100 sampled complex points with positive imaginary part uniformly within a Bernstein ellipse with foci ±1, and denote its elements as $\tilde{x}$. The radius for the Bernstein ellipse is set to be equal to $3^{16/n}$, as points farther away from the panel do not need the kernel-split quadrature scheme to be accurate [4]. This set is visualized in Fig. 3. By setting $\tilde{x} = 2x/h$, the estimate (44) can be rewritten as

$$\Lambda_S(\alpha, n, \tilde{x}, h) = \frac{n!}{(2n)!}(\alpha h)^n I_0^{(n)}\left(\frac{\alpha h}{2}|\tilde{x}|\right)\frac{h}{2} g_n(\tilde{x}), \tag{48}$$

and similarly the cancellation error (47) becomes

$$\Xi_S(\alpha, \tilde{x}, h) = \epsilon_{\text{mach}} h I_0\left(\frac{\alpha h}{2} d\right), \quad d = 1 + |\tilde{x}|. \tag{49}$$

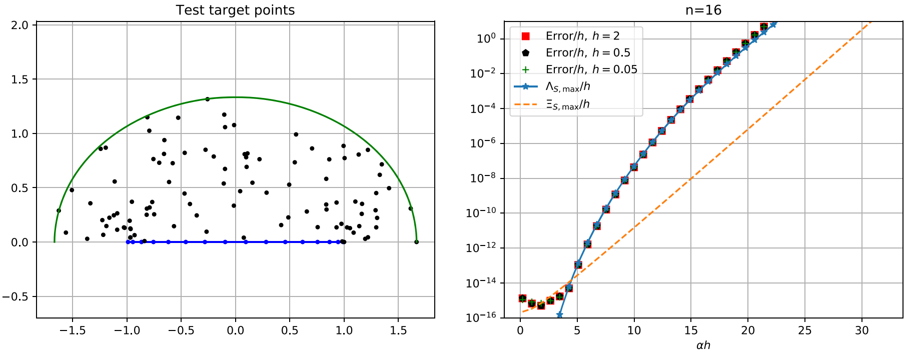


Fig. 3 Left: Distribution of 100 target points in the upper half of the Bernstein ellipse with foci at ±1, forming the set D, and a flat panel. Right: Plot of measured actual total errors scaled with $1/h$ for different h, error bound, and error estimates $\Lambda_{S,\max}/h$ and $\Xi_{S,\max}/h$ for the target points in the left plot for the single-layer kernel for the modified Helmholtz Eq. (21), with $n = 16$ over a range of αh

We define

$$\Lambda_{S,\max} = \max_{\tilde{x}\in D} |\Lambda_S(\alpha, n, \tilde{x}, h)|, \tag{50}$$

$$\Xi_{S,\max} = \max_{\tilde{x}\in D} \Xi_S(\alpha, \tilde{x}, h). \tag{51}$$

The maximum of these two quantities gives the error estimate

$$E_S(\alpha, n, h) = \max\left(\Lambda_{S,\max}, \Xi_{S,\max}\right), \tag{52}$$

for given values of α, n, and h. From (48) and (49), it is clear that E_S is a function with αh and h as two separate arguments, motivating the formulation

$$E_S(\alpha, n, h) = h\tilde{E}_S(\alpha h, n). \tag{53}$$

Hence, we expect the error divided by h to be a function of αh only, and this is confirmed in Figs. 3 and 4. Here, we plot the error scaled with $1/h$ for different values of h versus αh, for different values of n. We see that the error curves collapse for different values of h, just as predicted. Furthermore, from the plots, it is clear that the interpolation error and cancellation error dominate in different regimes. As n is increased, the interpolation error decreases, and the cancellation error will continue to be dominant for larger values of αh. Considering the left plot of Fig. 4, we can most clearly see how the measured error follows the cancellation error estimate $\Xi_{S,\max}$ first, and then shifts to follow the interpolation error estimate $\Lambda_{S,\max}$ as it becomes dominant. We can conclude that the error estimate $\tilde{E}_S$ predicts the actual error, scaled with $1/h$, quite well.

4.3 Other kernels

We now discuss error estimates for the other kernels of interest for the modified PDEs. The double-layer kernel for modified Helmholtz is discussed in detail, which

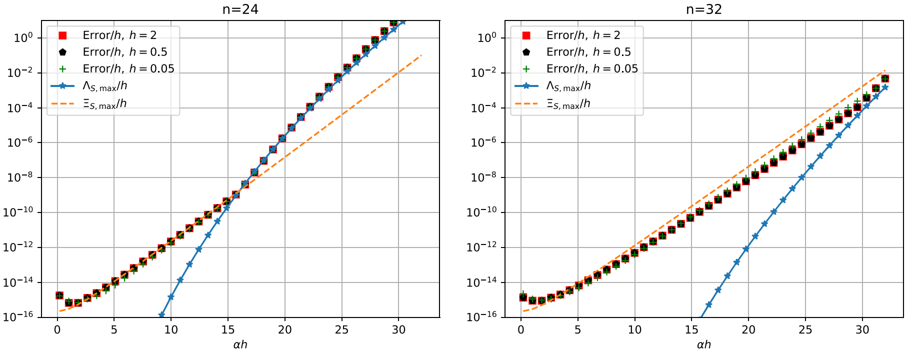


Fig. 4 Plots of measured actual total errors scaled with $1/h$ for different h, and error estimates $\Lambda_{S,\max}/h$ and $\Xi_{S,\max}/h$ for the target points in Fig. 3 for the single-layer kernel for the modified Helmholtz Eq. 21, with $n = 24$ (left) and 32 (right), over a range of αh

is used to motivate a similar treatment of the error estimates for the modified biharmonic equation and the modified Stokes equations. Their corresponding kernels, and the decompositions thereof into explicit singularities, are shown in Appendix 2.

The kernels for layer potentials associated with the modified Helmholtz equation (double layer), modified biharmonic, and modified Stokes are harder to study analytically than the single-layer kernel for the modified Helmholtz equation; G^L for all of them contain I_0 and/or I_1 multiplying a smooth function, which we collectively denote F. This means that the interpolation error (39) includes the nth derivative of a product. However, we observe that the derived error estimates of the interpolation error for the single-layer kernel can be applied to the these kernels as well, with a few modifications. This is not surprising, as the difficulties lie in properly resolving I_0 and/or I_1, not resolving the function F. The cancellation error (51) is also straightforward to modify.

Before proceeding with the error estimates, we present the split of the double-layer kernel for the modified Helmholtz equation into explicit singularities. The associated kernel is, again omitting the scaling $1/2\pi$,

$$G(x,y)=\frac{\partial}{\partial\hat{n}(x)}K_0(\alpha|y-x|)=-\alpha K_1(\alpha|y-x|)\frac{(y-x)\cdot\hat{n}(y)}{|y-x|}, \tag{54}$$

and by [16, §10.31] one has

$$K_1(\rho)=K_1^S(\rho)+\frac{1}{\rho}+I_1(\rho)\log\rho, \quad \rho\in\mathbb{R}_+. \tag{55}$$

The contents of the functions G^0, G^L, and G^C depend on the location of the target point x. If x belongs to the boundary, i.e., $x\in\partial\Omega$, then the kernel-split is

$$\begin{aligned}
G^S(x,y) &= -\alpha\left(K_1^S(\alpha|y-x|)+\frac{1}{\alpha|y-x|}+I_0(\alpha|y-x|)\log\alpha\right)\frac{(y-x)\cdot\hat{n}(y)}{|y-x|}, &(56)\\
G^L(x,y) &= -\alpha I_1(\alpha|y-x|)\frac{(y-x)\cdot\hat{n}(y)}{|y-x|}, &(57)\\
G^C(x,y) &= 0. &(58)
\end{aligned}$$

For off-boundary x, the kernel-split is

$$\begin{aligned}
G^S(x,y) &= -\alpha\left(K_1^S(\alpha|y-x|)+I_1(\alpha|y-x|)\log\alpha\right)\frac{(y-x)\cdot\hat{n}(y)}{|y-x|}, &(59)\\
G^L(x,y) &= -\alpha I_1(\alpha|y-x|)\frac{(y-x)\cdot\hat{n}(y)}{|y-x|}, &(60)\\
G^C(x,y) &= -1. &(61)
\end{aligned}$$

The main difference between the two decompositions is that $1/|y-x|$ in (55) goes into G^S for $x\in\partial\Omega$, due to the known limit value (17), while for $x\notin\partial\Omega$ it results in a non-zero G^C. For both kernel splits, the function G^L is smooth and consists of the modified Bessel function I_1 multiplied with $F=(y-x)\cdot\hat{n}(y)/|y-x|$. Here, F does not pose the difficulties with polynomial interpolation as I_1 does.

We now create an error estimate for the interpolation error. Again identifying the vectors x and y in $\mathbb{R}^2$ with complex numbers x and y in $\mathbb{C}$, and following the same steps as for the single-layer potential, we get analogous to (41)

$$r_n(x, y) = P_n(2y/h)\frac{(\alpha h)^n n!}{(2n)!}\frac{\mathrm{d}^n}{\mathrm{d}\xi^n}\left(\alpha I_1\big(\alpha|\xi h/2 - x|\big)\frac{\big(\xi h/2 - x\big)\cdot\hat{n}(\xi h/2)}{|\xi h/2 - x|}\right) \tag{62}$$

$$= P_n(2y/h)\frac{(\alpha h)^n n!}{(2n)!}\frac{\mathrm{d}^n}{\mathrm{d}\xi^n}\left(\operatorname{sgn}\big(\xi h/2 - x\big)\alpha I_1\Big(\alpha\big(\xi h/2 - x\big)\Big)\frac{\big(\xi h/2 - x\big)\cdot\hat{n}(\xi h/2)}{|\xi h/2 - x|}\right), \quad \xi \in [-1, 1], \tag{63}$$

since I_1 is an odd function. The derivation for the single-layer kernel that we based this result on was done for x in $\mathbb{R}$. We will now, as before, apply it to target points x in $\mathbb{C}$, for which $\big(\xi h/2 - x\big)\cdot\hat{n}(\xi h/2)$ is non-zero. To avoid the nth derivative of the product, we make the following argument. We exclude the value $x = \xi h/2$ since then r_n is equal to zero. Away from zero, the derivative of the sgn function is zero. Furthermore, it is I_1 that is difficult to represent accurately with polynomials and it contributes considerably more to the error than F does; thus, we keep only the dominant term obtained by the chain rule and write

$$r_n(x, y) \approx P_n(2y/h)\frac{(\alpha h)^n n!}{(2n)!}\operatorname{sgn}\big(\xi h/2 - x\big)\alpha I_1^{(n)}\big(\alpha(\xi h/2 - x)\big)\frac{\big(\xi h/2 - x\big)\cdot\hat{n}(\xi h/2)}{|\xi h/2 - x|}, \quad \xi \in [-1, 1]. \tag{64}$$

As for the single layer we set ξ to zero, which results in the estimate

$$\Lambda_D(\alpha, n, \tilde{x}, h) = \tfrac{n!}{(2n)!}(\alpha h)^{n+1} I_1^{(n)}\left(\tfrac{\alpha h}{2}|\tilde{x}|\right)\tfrac{-\mathrm{Im}(\tilde{x})}{|\tilde{x}|}\tfrac{1}{2}g_n(\tilde{x}), \tag{65}$$

where $g_n(\tilde{x})$ is given by (45) and $\tilde{x} = 2x/h$ as before. Again we have reinserted the absolute value in the argument of the modified Bessel function, by removing the sgn function, since by (23) and (24) $I_1^{(n)}$ is an odd function for even n.

For the cancellation error, similarly

$$\Xi_D(\alpha, \tilde{x}, h) = \epsilon_{\mathrm{mach}}\alpha h I_1\left(\frac{\alpha h}{2}d\right)\frac{\mathrm{Im}(\tilde{x})}{|\tilde{x}|}, \quad d = 1 + |\tilde{x}|, \tag{66}$$

where $(y - x)\cdot\hat{n}(y)/|y - x|$ has been replaced with $\mathrm{Im}(\tilde{x})/|\tilde{x}|$.

Clearly, the error estimates depend purely on αh, without any multiplying h-factor, in difference to the single-layer kernel. For the double-layer potential, the error estimate is

$$E_D(\alpha, n, h) = \tilde{E}_D(\alpha h, n) = \max\left(\Lambda_{D,\max}, \Xi_{D,\max}\right), \tag{67}$$

where $\Lambda_{D,\max}$ and $\Xi_{D,\max}$ are defined analogous to (50) and (51) for (65) and (66). The predicted pure dependence on αh for the measured numerical error can be seen in Fig. 5. The estimate for the cancellation error is not as tight as for the single-layer kernel for $n = 32$. Still, at its worst, the estimates are off by a digit of the actual error. Also, we see that it is possible to ignore the chain rule and only differentiate I_1, but not F, and obtain good estimates.

To derive error estimates for the kernels associated with the other modified PDEs, a similar approach as above can be taken. The error estimate E will have the form

$$E(\alpha, n, h) = H(h)\tilde{E}(\alpha h, n), \tag{68}$$

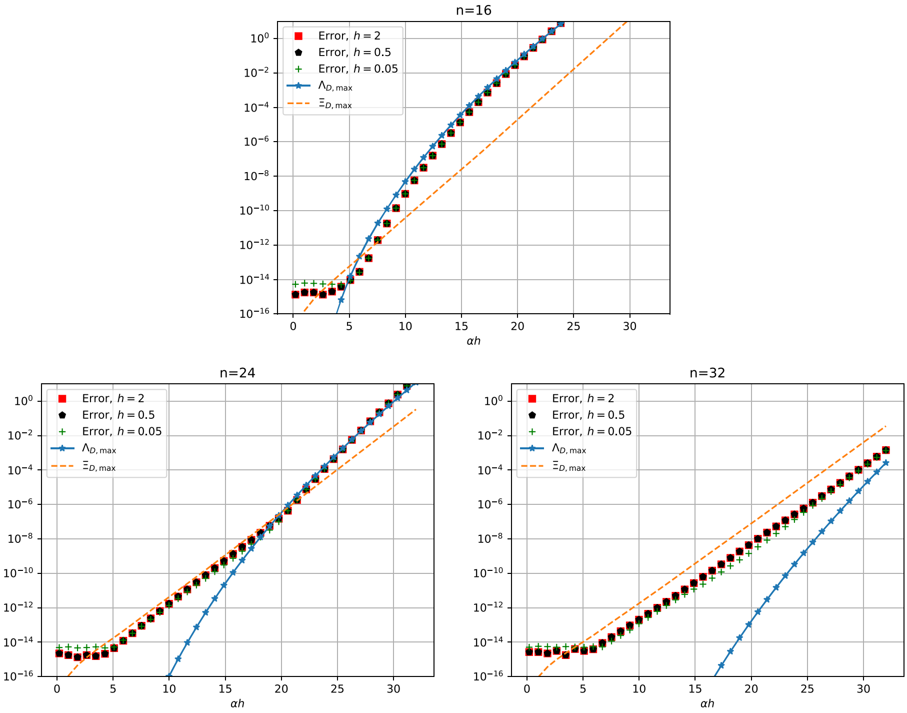


Fig. 5 Plots of measured actual total errors for different h, and error estimates $\Lambda_{D,\max}$ and $\Xi_{D,\max}$ for the target points in Fig. 3 for the double-layer kernel for the modified Helmholtz (54), with $n = 16$ (top), $n = 24$ (left), and 32 (right), over a range of αh

where simply $H(h) = h^p$ for some p.

The discussion that we have put forward has given an understanding of the two sources of errors. Practically, only a value for C_ε needs to be determined; it sets panel lengths in the subdivision algorithm, introduced in the next section. Instead of deriving an error estimate, one can simply use figures such as Figs. 3, 4, and 5 for the measured numerical errors for this purpose. Here, one needs to find the proper scaling $H(h)$ by which to scale the errors before plotting, such that they coincide for different h. In Fig. 6, such results are presented for the double-layer kernel for the modified Stokes (106), with $p = 1$. The error curves do not collapse as neatly as for the modified Helmholtz equation, indicating that there is a scaling factor other than h. Still, the results can be used to set C_ε, as demonstrated in Section 6.

4.4 Condition for αh

As was shown above, the error when evaluating the double layer modified Helmholtz kernel with an n point Gauss–Legendre quadrature rule over one panel with length h only depends on n and αh. That means that given the error tolerance ε, $C_\varepsilon = \alpha h$ that yield this error can be read off from a plot such as in Fig. 5 for the given n. Fixing that C_ε, the condition (as written in (28)) is $\alpha h \leq C_\varepsilon$.

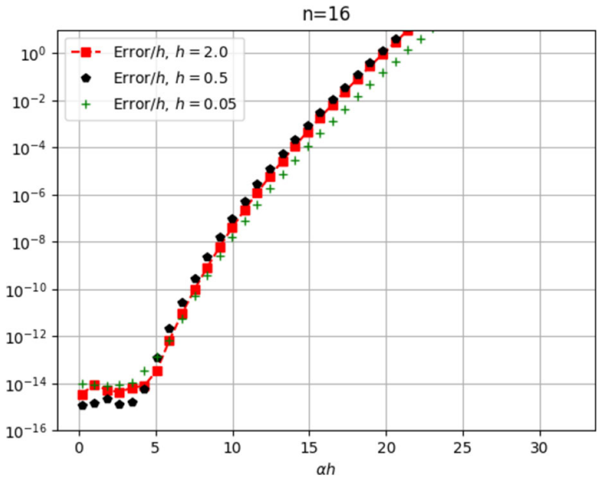


Fig. 6 Plot of measured actual total errors for different h, for the target points in Fig. 3 for the double-layer kernel for the modified Stokes (106), with $n = 16$, over a range of αh

For other kernels, such as the single-layer modified Helmholtz kernel, there is not a pure αh dependence on the error, as we have seen. However, to keep the procedure simple, we would suggest the following approach. Pick a typical value of the panel length h for the discretization at hand. Then, divide the given tolerance ε by this h, and use this scaled tolerance when determining C_ε from plots of the scaled error for the appropriate n, such as in Figs. 3 and 4.

In the subdivision algorithm, the reduction in error through this extra factor of h of refinement will then not be taken into account, and hence the panels might be refined to be somewhat smaller than needed. But this will allow us to keep the same simple structure of the subdivision algorithm for all different kernels.

5 Quadrature by recursive subdivision

To circumvent the problem of interpolatory quadrature failing for large α, we here introduce an algorithm for local refinement, based on panel subdivision. Given a single-source panel Γ, we assume that it is sufficiently short, relative to the quadrature order n, for both the geometry and the layer density to be well represented by a polynomial, interpolated at the n quadrature points. We say that it is *well resolved.* For a given target point x, we can then subdivide Γ into a set of M subpanels $\{\Gamma_i\}_{i=1}^{M}$, interpolate our known quantities from the quadrature nodes on Γ to the quadrature nodes on those subpanels, and then evaluate the layer potential at x using the subpanels. To ensure accuracy for a given tolerance, this subdivision is formed in a way that guarantees that all subpanels either are short enough to satisfy (28), obtained via (52) or similarly, or are sufficiently far away from x, relative to their own length, to not need kernel-split quadrature.

Before we can state our algorithm, a number of preliminaries are needed.

Preimage of target Let $\gamma(t) : \mathbb{R} \to \mathbb{C}$ be the mapping from the standard interval $[-1, 1]$ to the panel Γ. Then z, such that $\gamma(z) = x$, is the *preimage* of the target point x. The preimage z is real-valued if $x \in \partial\Omega$, and complex-valued otherwise. We here assume that we know the value of z, but $\gamma(t)$ need not be a known function; see [2] for a discussion on how to construct a numerical representation.

Subpanels and subintervals A subdivision of Γ is defined by a set of edges in the parametrization, $\{-1 = t_1, t_2, \dots, t_{M+1} = 1\}$, such that a subpanel Γ_i is given by the mapping of the subinterval $[t_i, t_{i+1}]$ under γ. We can, by a linear scaling, define the *local mapping* that maps the standard interval $[-1, 1]$ to Γ_i as

$$\gamma_i(t) = \gamma\left(t_i + \frac{\Delta t_i}{2}(t+1)\right), \tag{69}$$

where $\Delta t_i = t_{i+1} - t_i$. Given the preimage z, the *local preimage* z_i, such that $\gamma_i(z_i) = x$, is given by

$$z_i = \frac{2}{\Delta t_i}(z - t_i) - 1. \tag{70}$$

Near evaluation criterion Given the preimage z of a point x close to a panel (or subpanel) Γ, it is possible to compute an accurate estimate of the quadrature error when evaluating the layer potential using n-point Gauss–Legendre quadrature. Detailed discussions can be found in [1, 2]. To leading order, the error is proportional $\rho(z)^{-2n}$, where ρ is the elliptical radius of the Bernstein ellipse on which z lies,

$$\rho(z) = \left|z + \sqrt{z^2 - 1}\right|, \tag{71}$$

where $\sqrt{z^2 - 1}$ is defined as $\sqrt{z+1}\sqrt{z-1}$ with $-\pi < \arg(z \pm 1) \leq \pi$. For a given kernel G and error tolerance ε, it is then possible to introduce a cutoff radius R_ε, such that kernel-split quadrature must be used for

$$\rho(z) < R_\varepsilon, \tag{72}$$

and otherwise Gauss–Legendre quadrature is sufficiently accurate. We refer to this as the target point and the source points being *well-separated*; otherwise, they are considered to be *close*. See [2] on how to set R_ε for a given tolerance; we use $R_\varepsilon = 3.5$ for $n = 16$, which corresponds to a tolerance of 10^{-14}.

Later, we will use that the inverse of ρ has a particularly simple form in the special case when z lies on the imaginary axis,

$$\begin{aligned} z &= \pm \mathrm{i}b, \quad b > 0, \\ \rho(z) &= b + \sqrt{b^2 + 1}, \\ z(\rho) &= \pm\frac{\mathrm{i}(\rho^2 - 1)}{2\rho}. \end{aligned} \tag{73}$$

Interpolation and upsampling To interpolate data from the n original Gauss–Legendre nodes on $[-1, 1]$, to m new Gauss–Legendre nodes on a subinterval $[t_i, t_{i+1}] \subset [-1, 1]$, we use barycentric Lagrange interpolation [5]. By *upsampling*, we refer to the special case of interpolating from n to $2n$ Gauss–Legedre nodes, both on $[-1, 1]$.

Subinterval length criterion When a new subpanel Γ_i is formed on Γ, we need to check if it satisfies the kernel-split accuracy criterion (28), which requires knowledge of the arc length of the subpanel, denoted h_i. Assuming that $\gamma'(t)$ does not vary rapidly on Γ, a good approximation to h_i is $h_i \approx h\Delta t_i/2$, where h is the arc length of Γ. We can now combine this approximation with (28), to get an accuracy criterion formulated in subinterval size,

$$\Delta t_i \leq \frac{2C_\varepsilon}{\alpha h}. \tag{74}$$

In particular, this allows us to write down the maximum length of subintervals on which product integration can be used,

$$\Delta t_{\max} = \frac{2C_\varepsilon}{\alpha h}. \tag{75}$$

Here, C_ε can be obtained via (52), as a precomputation step.

5.1 Algorithm

Our algorithm proceeds with creating a division of $[-1, 1]$ into subintervals, which corresponds to a division of Γ into subpanels.

For a target point x with preimage z such that $\operatorname{Re} z \in (-1, 1)$, the first step is to create a subinterval centered on $\operatorname{Re} z$, with length set to satisfy both of the conditions (72) and (75). The centering ensures that the subpanels will not introduce new edges or quadrature nodes that are close enough to z to degrade precision.

If this initial subinterval has length Δt_c, then the preimage of x in that local frame will be $z_c = \mathrm{i}b_c$, with $b_c = 2 \operatorname{Im} z/\Delta t_c$ and (73) is applicable.

If we wish the local preimage z_c to be just beyond the limit where kernel-split quadrature is needed, then we must set Δt_c such that $\rho(z_c) = R_\varepsilon$. From (73), we can derive that this is satisfied when $\Delta t_c = \Delta t_{\text{direct}}$,

$$\Delta t_{\text{direct}} = |\operatorname{Im} z| \frac{4R_\varepsilon}{{R_\varepsilon}^2 - 1}. \tag{76}$$

For the subinterval to be contained within $[-1, 1]$, it may not be bigger than twice the distance between $\operatorname{Re} z$ and the closest interval edge,

$$\Delta t_{\text{edge}} = 2(1 - |\operatorname{Re} z|). \tag{77}$$

Now we set the initial subpanel as large as possible, while still ensuring that the quadrature from it is accurate, and that it falls within $[-1, 1]$,

$$\Delta t_c = \min\left(\Delta t_{\text{edge}}, \max\left(\Delta t_{\text{direct}}, \Delta t_{\max}\right)\right). \tag{78}$$

Here, Δt_{direct} depends on the location of the target point, while $\Delta t_{\max}$ does not. Thus, for points sufficiently far away $\Delta t_{\max}$ might be larger than Δt_{direct}.

By (78), we have the initial subdivision $\{-1, \operatorname{Re} z - \Delta t_c/2, \operatorname{Re} z + \Delta t_c/2, 1\}$. The center subinterval is now acceptable, and we proceed by recursively bisecting each remaining subinterval until either its length satisfies (74), or the local preimage of x satisfies (72).

For target points such that $\operatorname{Re} z \notin (-1, 1)$, we can skip the process of carefully selecting the length of the nearest subinterval, and proceed immediately with recursive bisection of $\{-1, 1\}$. This completes the algorithm, which we list in its entirety in Algorithm 1.

Algorithm 1 Given a panel of length h and a nearby target point with preimage z, create a subdivision of $[-1, 1]$ that allows the layer potential to be accurately evaluated, using either direct Gauss–Legendre quadrature or kernel-split quadrature.

```
function CREATE_SUBDIVISION($z, h, \alpha, C_\varepsilon, R_\varepsilon$)
    $\Delta t_{\text{max}} \leftarrow 2C_\varepsilon/(\alpha h)$
    if $|\operatorname{Re} z| \geq 1$ then        ▷ Preimage outside interval, recursively bisect all of it.
        return RECURSIVE_BISECTION($-1, 1, R_\varepsilon, \Delta t_{\text{max}}, z$).
    else
        $\Delta t_{\text{direct}} \leftarrow 4|\operatorname{Im} z|R_\varepsilon/({R_\varepsilon}^2 - 1)$
        $\Delta t_{\text{edge}} = 2(1 - |\operatorname{Re} z|)$
        $\Delta t_c \leftarrow \min\left(\Delta t_{\text{edge}}, \max\left(\Delta t_{\text{direct}}, \Delta t_{\text{max}}\right)\right)$
        $t_a \leftarrow \operatorname{Re} z - \Delta t_c/2$
        $t_b \leftarrow \operatorname{Re} z + \Delta t_c/2$
                ▷ Center interval is now acceptable, recursively bisect remainder subintervals.
        $S_1 \leftarrow$ RECURSIVE_BISECTION($-1, t_a, R_\varepsilon, \Delta t_{\text{max}}, z$)
        $S_2 \leftarrow \{t_a, t_b\}$
        $S_3 \leftarrow$ RECURSIVE_BISECTION($t_b, 1, R_\varepsilon, \Delta t_{\text{max}}, z$)
        return $S_1 \cup S_2 \cup S_3$
    end if
end function

function RECURSIVE_BISECTION($t_1, t_2, R_\varepsilon, \Delta t_{\text{max}}, z$)
    if $t_1 < t_2$ then
        $\Delta t_{\text{sub}} \leftarrow t_2 - t_1$
        $z_{\text{sub}} \leftarrow 2(z - t_1)/\Delta t_{\text{sub}} - 1$        ▷ From (70).
        if $\rho(z_{\text{sub}}) < R_\varepsilon$ and $\Delta t_{\text{sub}} > \Delta t_{\text{max}}$ then        ▷ Using (71).
                ▷ Kernel-split must be used, but interval still too large. Continue bisection.
            $t_{\text{mid}} \leftarrow t_1 + \Delta t_{\text{sub}}/2$
            $S_1 \leftarrow$ RECURSIVE_BISECTION($t_1, t_{\text{mid}}, R_\varepsilon, \Delta t_{\text{max}}, z$)
            $S_2 \leftarrow$ RECURSIVE_BISECTION($t_{\text{mid}}, t_2, R_\varepsilon, \Delta t_{\text{max}}, z$)
            return $S_1 \cup S_2$
        end if
    end if
    return $\{t_1, t_2\}$        ▷ Subinterval passed.
end function
```

6 Numerical results

To test the robustness of our method across a range of α values, we solve the modified Helmholtz equation and the modified Stokes equations. Based on the discussion in Section 4, the parameter C_ε is set to confirm that prescribed error tolerances are satisfied.

6.1 The modified Helmholtz equation

We solve the modified Helmholtz (1) inside the annulus centered at the origin, defined by a circle of radius 0.3, and a circle of radius 0.6, with Dirichlet boundary conditions given by a fundamental solution (21) centered in the inner circle,

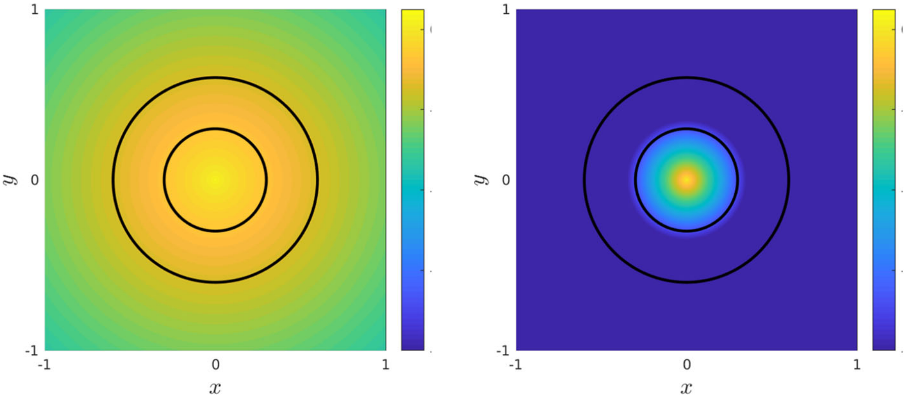
$$\begin{aligned} (\Delta - \alpha^2)u &= 0, & x &\in \Omega, \qquad (79)\\ u &= K_0(\alpha|x - x_0|), & x &\in \partial\Omega, \qquad (80)\\ x_0 &= (0.01, 0.01)^T. & & \qquad (81) \end{aligned}$$

The exact solution to this problem is equal to the expression for the Dirichlet boundary condition, evaluated in Ω. See Fig. 7 for a visualization of (80) for $\alpha = 10,\ 100$. We see that $K_0(\alpha|x - x_0|)$ decays rapidly for $\alpha = 100$.

To solve the above problem numerically, we represent the solution using the double-layer potential

$$u(x) = \int_{\partial\Omega} G(x, y)\sigma(y)\,\mathrm{d}S(y), \qquad (82)$$

where G is the double-layer kernel (54).



(a) $K_0(10\|x - x_0\|)$

(b) $K_0(100\|x - x_0\|)$

Fig. 7 Plots of $K_0(\alpha\|x - x_0\|)$, the modified Bessel function of the second kind of zeroth order, for $\alpha = 10, 100$ and $x_0 = (0.01, 0.01)^T$. The black circles of radius 0.3 and 0.6 define the annulus that is computational domain

Enforcing the boundary condition (80) gives a second kind integral equation in σ,

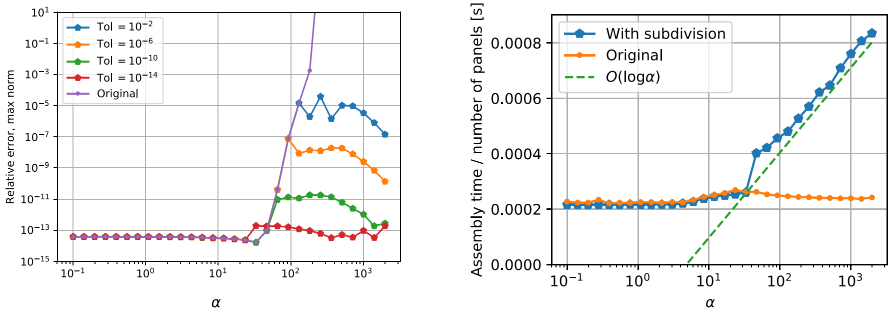
$$\sigma(x) + \int_{\partial\Omega} G(x, y)\sigma(y)\, \mathrm{d}S(y) = K_0(\alpha|x - x_0|), \quad x \in \partial\Omega. \tag{83}$$

We solve this using the Nyström method, discretizing the boundary using 16-point Gauss–Legendre panels, with 15 panels on the inner circle, and 30 panels on the outer. For the bound (76), we use use $R_\varepsilon = 3.5$, and for (74) we set $C_\varepsilon = 3.7$ by reading off Fig. 5 to achieve a tolerance of 10^{-14}. Following the notation of (5), the kernel-split of (54) is given by (56) – (61).

For a large range of α values, we solve the integral equation, and then evaluate the solution at 15 random points on a circle of radius 0.301 (very close to the inner boundary). Quadrature by recursive bisection is applied both when solving for the density (83) and for evaluating the solution (82).

The results, shown in Fig. 8, demonstrate that our subdivision algorithm is capable of avoiding the catastrophic loss of accuracy otherwise present above a threshold α, and that the additional cost incurred from it is proportional to $\log\alpha$. We do not satisfy the error tolerance of 10^{-14}; around one digit of accuracy appears to be lost, presumably due to the additional interpolation steps involved.

We also try other values for C_ε to confirm that we stay under a given tolerance. Based on Fig. 5, we have $C_\varepsilon = 18.7,\ 12.8,\ 8.2$ for the tolerances $10^{-2},\ 10^{-6},\ 10^{-10}$. The largest corresponding relative errors for each tolerance, taken over the plotted range of values of α, are $3.7 \cdot 10^{-5},\ 6.7 \cdot 10^{-8},\ 1.8 \cdot 10^{-11}$. Clearly the errors are at least one digit below the prescribed tolerance. It is not surprising that the set value for C_ε gives a lower error than Fig. 5 suggests. In the subdivision algorithm introduced in this paper, the first panel in the recursive scheme always has as its center the projection of the target point on the boundary. Therefore, target points are never close to panel edges, where errors from the kernel-split quadrature tend to be greater than for target points towards the panel's center [13].



(a) Error when evaluating the layer potential close to the boundary, relative error in max-norm computed as $\|u - u_{\text{ref}}\|_\infty / \|\sigma\|_\infty$.

(b) Time to assemble the matrix blocks that correspond to neighboring panels, which are where the log-correction need to be added for the tolerance 10^{-14}.

Fig. 8 Comparison of the original kernel-split algorithm, denoted "Original," and our adaptive algorithm, denoted "Tol = tolerance subdivision" for given tolerances, when solving our test problem for the modified Helmholtz equation for a large range of α. We test the solution up to $\alpha \approx 2000$; for larger values of α, the solution is about round-off in the entire domain

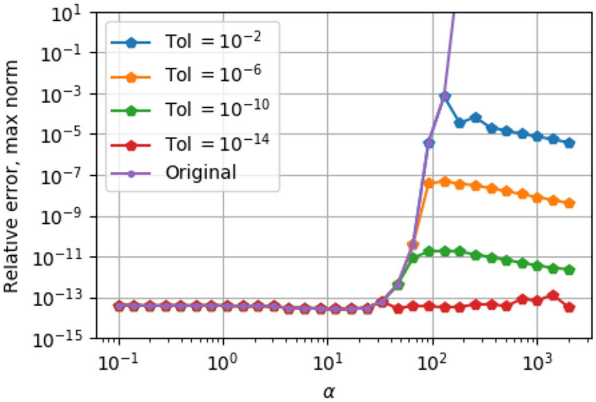


Fig. 9 Comparison of the original kernel-split algorithm, denoted "Original," and our adaptive algorithm, denoted "Tol = tolerance subdivision" for given tolerances, when solving our test problem for the modified Stokes equations for a large range of α

6.2 The modified Stokes equations

We solve the modified Stokes (3) in a setting analogous to (80); the geometry, and the discretization thereof, is the same, and the boundary data and the solution is given by the associated fundamental solution (106). As opposed to the modified Helmholtz equation, no error estimates are presented in this paper for the modified Stokes equation, which can be used to set C_ε. Instead, one can use the measured numerical errors for a flat panel for different values of h, each scaled with $1/h^p$, as suggested in Section 4.3. Here, p is chosen such that the error curves collapse. For the modified Stokes in a double-layer formulation, such results are shown in Fig. 6, with $p = 1$.

Following the methodology presented in Section 4.4, we set C_ε to achieve certain tolerances. For the tolerances 10^{-2}, 10^{-6}, 10^{-10}, 10^{-14}, we set $C_\varepsilon = 16.7, 11.5, 7.4, 4.4$ based on Fig. 6. The corresponding maximum errors over all alpha are $3.8 \cdot 10^{-5}$, $4.8 \cdot 10^{-8}$, $1.1 \cdot 10^{-11}$, $5.3 \cdot 10^{-14}$, shown in Fig. 9. The magnitudes of the errors, for a given tolerance, are consistent with the results for the modified Helmholtz equation, presented in the previous section. The errors are well below the prescribed tolerance, meaning more subdivisions are applied than necessary. The hypothesis is the same as for the modified Helmholtz; for the subdivision algorithm the target point, or the projection thereof, on the boundary is always centered on the new panel. Thus, it is never close to the panel edges, where the error tends to be greater.

7 Conclusions

We present a robust recursive algorithm that allows the method of Helsing et al. to be applied, for any α, to the modified Helmholtz equation, modified biharmonic equation, and modified Stokes equations on smooth geometries. Before, this was not possible for large α, which corresponds to small time-steps with semi-implicit marching schemes for the heat equation and the time-dependent Stokes and Navier-Stokes equations. Our algorithm is fully adaptive, and the additional computational time it requires scales as $\log \alpha$. Our choice of the parameters C_ε and R_ε is based on numerical observations and provide excellent results.

Appendix 1: Formulas for recursive computation of the Legendre-log integrals

1.1 Result

The integrals

$$g_n(x) = \int_{-1}^{1} P_n(\rho) \log(\rho - x) \mathrm{d}\rho, \quad n = 0, 1, 2, \ldots, \tag{84}$$

can be computed recursively using the formulas

$$\begin{aligned}
g_0(x) &= (1-x)\log(1-x) + (1+x)\log(-1-x) - 2, \\
g_1(x) &= \frac{1}{2}\left(\left(1-x^2\right)\log(1-x) - \left(1-x^2\right)\log(-x-1) - 2x\right), \\
g_2(x) &= \frac{1}{3}\big(3xg_1 + 2\big), \\
&\vdots \\
g_{n+1}(x) &= \frac{1}{n+2}\big((2n+1)xg_n(x) - (n-1)g_{n-1}(x)\big), \quad n \geq 2. \qquad (85)
\end{aligned}$$

The first two formulas, for $g_0(x)$ and $g_1(x)$, have finite limits as $x \to \pm 1$.

1.2 Proof

The formulas for g_0 and g_1 follow from direct integration of (84), with $P_0(\rho) = 1$ and $P_1(\rho) = \rho$. For $n > 1$, the following two results are useful:

$$(n+1)P_{n+1}(\rho) = (2n+1)\rho P_n(\rho) - nP_{n-1}(\rho), \tag{86}$$

$$(2n+1)P_n(\rho) = \frac{\mathrm{d}}{\mathrm{d}\rho}\big(P_{n+1}(\rho) - P_{n-1}(\rho)\big). \tag{87}$$

Insertion of the recursion formula (86) into (84) gives

$$\begin{aligned}
(n+1)g_{n+1}(x) &= (2n+1)\int_{-1}^{1} \rho P_n(\rho)\log(\rho-x)\mathrm{d}\rho - n\int_{-1}^{1} P_{n-1}(\rho)\log(\rho-x)\mathrm{d}\rho & (88) \\
&= (2n+1)\int_{-1}^{1} P_n(\rho)(\rho - x + x)\log(\rho-x)\mathrm{d}\rho - ng_{n-1}(x) & (89) \\
&= (2n+1)\int_{-1}^{1} P_n(\rho)(\rho-x)\log(\rho-x)\mathrm{d}\rho + (2n+1)xg_n(x) - ng_{n-1}(x). & (90)
\end{aligned}$$

Integration by parts of remaining integral, using (87) and $((\rho - x)\log(\rho - x))' = 1 + \log(\rho - x)$,

$$(2n+1)\int_{-1}^{1} P_n(\rho)(\rho - x)\log(\rho - x)\mathrm{d}\rho = \tag{91}$$

$$= \underbrace{\Big[\big(P_{n+1}(\rho) - P_{n-1}(\rho)\big)\log(\rho - x)\Big]_{-1}^{1}}_{I_1} - \underbrace{\int_{-1}^{1}\big(P_{n+1}(\rho) - P_{n-1}(\rho)\big)\big(1 + \log(\rho - x)\big)\mathrm{d}\rho}_{I_2}. \tag{92}$$

We have $P_n(1) = 1$ and $P_n(-1) = (-1)^n$, so it follows that $I_1 = 0$. Furthermore,

$$\int_{-1}^{1} P_n(\rho)\mathrm{d}\rho = 2\delta_{0n} = \begin{cases} 2, & \text{if} \quad n = 0 \\ 0, & \text{if} \quad n \geq 1. \end{cases} \tag{93}$$

Consequently, since $n \geq 0$,

$$\begin{aligned} I_2 &= -2\delta_{0(n-1)} + g_{n+1}(x) - g_{n-1}(x), & (94)\\ \Rightarrow (n+1)g_{n+1}(x) &= 2\delta_{0(n-1)} - g_{n+1}(x) + g_{n-1}(x) + (2n+1)xg_n(x) - ng_{n-1}(x), & (95)\\ \Rightarrow (n+2)g_{n+1}(x) &= (2n+1)xg_n(x) - (n-1)g_{n-1}(x) + 2\delta_{0(n-1)}, & (96) \end{aligned}$$

which leads to the recursion formulas (85).

Appendix 2: Kernel splits

Here, we complement the kernel split for the modified Helmholtz equation with kernel splits for the modified biharmonic equation and the modified Stokes equations.

2.1 Modified biharmonic equation

The single-layer kernel for the modified biharmonic equation is

$$G(x, y) = \frac{-1}{2\pi\alpha^2}(\log|y - x| + K_0(\alpha|y - x|)). \tag{97}$$

Inserting (22) the explicit split becomes

$$\begin{aligned} G^S &= \frac{-1}{2\pi\alpha^2}(K_0^S(\alpha|y - x|) + I_0(\alpha|y - x|)\log\alpha), & (98)\\ G^L &= \frac{-1}{2\pi\alpha^2}(1 + I_0(\alpha|y - x|)), & (99)\\ G^C &= 0. & (100) \end{aligned}$$

Here, G^L contains the modified Bessel function I_0, like the single-layer kernel for the modified Helmholtz equation. The double-layer kernel is

$$G(x, y) = \frac{\partial}{\partial \hat{n}(y)} \frac{-1}{2\pi\alpha^2} (\log|y-x| + K_0(\alpha|y-x|)) = \frac{-1}{2\pi\alpha^2} \left(\frac{1}{|y-x|} + \alpha K_1(\alpha|y-x|)\right) \frac{(y-x)\cdot\hat{n}(y)}{|y-x|} \tag{101}$$

$$= \frac{-1}{2\pi\alpha^2} \left(\frac{1}{|y-x|} + \alpha K_1^S(\alpha|y-x|) + \frac{1}{|y-x|} + \alpha I_1(\alpha|y-x|) \log(\alpha|y-x|)\right) \frac{(y-x)\cdot\hat{n}(y)}{|y-x|} \tag{102}$$

by (55). The resulting decomposition is

$$G^S = \frac{-1}{2\pi\alpha} (K_1^S(\alpha|y-x|) + I_1(\alpha|y-x|) \log\alpha) \frac{(y-x)\cdot\hat{n}(y)}{|y-x|}, \tag{103}$$

$$G^L = \frac{-1}{2\pi\alpha} I_1(\alpha|y-x|) \frac{(y-x)\cdot\hat{n}(y)}{|y-x|}, \tag{104}$$

$$G^C = \frac{-1}{\pi\alpha^2}. \tag{105}$$

2.2 The modified Stokes equations

We present only the double-layer kernel for the modified Stokes equations, also known as the stresslet. In the Einstein summation convention, it has the closed-form

$$G_{ijk} = \alpha^2 \mathcal{G}_1(\alpha\|\mathbf{r}\|)(\delta_{jk} r_i + \delta_{ik} r_j + \delta_{ij} r_k) + \alpha^4 \mathcal{G}_2(\alpha\|\mathbf{r}\|) r_i r_j r_k + \alpha^2 \mathcal{G}_3(\alpha\|\mathbf{r}\|)\delta_{ik} r_j, \tag{106}$$

where δ_{ij} is the Kronecker delta and $\mathbf{r} = \mathbf{x} - \mathbf{y}$. Here, the functions $\mathcal{G}_1$–$\mathcal{G}_3$ are

$$\mathcal{G}_1(\rho) = -\frac{2\rho^2 K_0(\rho) + (\rho^2 + 4)\rho K_1(\rho) - 4}{2\pi\rho^4}, \tag{107}$$

$$\mathcal{G}_2(\rho) = \frac{4\rho^2 K_0(\rho) + (\rho^2 + 8)\rho K_1(\rho) - 8}{\pi\rho^6}, \tag{108}$$

$$\mathcal{G}_3(\rho) = \frac{\rho K_1(\rho) - 1}{2\pi\rho^2}. \tag{109}$$

Since they are expressed in terms of the modified Bessel functions K_0 and K_1, we can use the decompositions (22) and (55) to write the expressions in explicit singularities. We have

$$\mathcal{G}_1(\rho) = \mathcal{G}_1^S(\rho) + \mathcal{G}_1^L(\rho) \log\rho, \tag{110}$$

$$\mathcal{G}_2(\rho) = \mathcal{G}_2^S(\rho) + \mathcal{G}_2^L(\rho) \log\rho + \frac{1}{8\pi\rho^2} - \frac{1}{\pi\rho^4}, \tag{111}$$

$$\mathcal{G}_3(\rho) = \mathcal{G}_S^3(\rho) + \mathcal{G}_3^L(\rho) \log\rho, \tag{112}$$

where

$$\mathcal{G}_1^S = -\frac{2\rho K_0^S(\rho) + (\rho^2+4)K_1^S(\rho) + \rho}{2\pi\rho^3}, \qquad \mathcal{G}_1^L = \frac{2\rho I_0(\rho) - (\rho^2+4)I_1(\rho)}{2\pi\rho^3}, \tag{113}$$

$$\mathcal{G}_2^S = \frac{32\rho K_0^S(\rho) + 8(\rho^2+8)K_1^S(\rho) - \rho(\rho^2-16)}{8\pi\rho^5}, \qquad \mathcal{G}_2^L = \frac{(\rho^2+8)I_1(\rho) - 4\rho I_0(\rho)}{\pi\rho^5}, \tag{114}$$

$$\mathcal{G}_3^S = \frac{K_1^S(\rho)}{2\pi\rho}, \qquad \mathcal{G}_L^3 = \frac{I_1(\rho)}{2\pi\rho}. \tag{115}$$

With these definitions the kernel-split form of the stresslet is

$$T_{ijk}(\mathbf{r}) = \left(T_{ijk}^S(\mathbf{r}) + T_{ijk}^L(\mathbf{r})\log(\alpha)\right) + T_{ijk}^L(\mathbf{r})\log\|\mathbf{r}\| + T_{ij}^C(\mathbf{r})\frac{r_k}{\|\mathbf{r}\|} + T^Q\frac{r_ir_jr_k}{\|\mathbf{r}\|^4}, \tag{116}$$

$$G_{ijk}(\mathbf{r}) = G_{ijk}^S(\mathbf{r}) + T_{ijk}^L(\mathbf{r})\log\|\mathbf{r}\| + T_{ij}^C(\mathbf{r})\frac{r_k}{\|\mathbf{r}\|} + T^Q\frac{r_ir_jr_k}{\|\mathbf{r}\|^4} \tag{117}$$

where

$$G_{ijk}^S(\mathbf{r}) = \alpha^2\mathcal{T}_1^S(\alpha\|\mathbf{r}\|)(\delta_{jk}r_i + \delta_{ik}r_j + \delta_{ij}r_k) + \alpha^4\mathcal{T}_2^S(\alpha\|\mathbf{r}\|)r_ir_jr_k + \alpha^2\mathcal{T}_3^S(\alpha\|\mathbf{r}\|)\delta_{ik}r_j, \tag{118}$$

$$T_{ijk}^S(\mathbf{r}) = \alpha^2\mathcal{T}_1^L(\alpha\|\mathbf{r}\|)(\delta_{jk}r_i + \delta_{ik}r_j + \delta_{ij}r_k) + \alpha^4\mathcal{T}_2^L(\alpha\|\mathbf{r}\|)r_ir_jr_k + \alpha^2\mathcal{T}_3^L(\alpha\|\mathbf{r}\|)\delta_{ik}r_j, \tag{119}$$

$$T_{ij}^C(\mathbf{r}) = \alpha^2 r_ir_j/8\pi, \tag{120}$$

$$T_{ij}^C(\mathbf{r}) = -1/\pi. \tag{121}$$

Funding Open access funding provided by Royal Institute of Technology. The authors received the support from the Knut and Alice Wallenberg Foundation under grant no. 2016.0410 (L.a.K.), and by the Swedish Research Council under Grant No. 2015-04998 (F.F. and A.K.T.)

Declarations

Competing interests The authors declare no competing interests.

References

1. af Klinteberg, L., Tornberg, A.-K.: Error estimation for quadrature by expansion in layer potential evaluation. Adv. Comput. Math. **43**(1), 195–234 (2017). https://doi.org/10.1007/s10444-016-9484-x
2. af Klinteberg, L., Tornberg, A.-K.: Adaptive quadrature by expansion for layer potential evaluation in two dimensions. SIAM J. Sci Comput. **40**(3), A1225–A1249 (2018). https://doi.org/10.1137/17M1121615

3. af Klinteberg, L., Askham, T., Kropinski, M.C.: A fast integral equation method for the two-dimensional navier-stokes equations. J. Comput. Phys. **409**, 109353 (2020). https://doi.org/10.1016/j.jcp.2020.109353
4. af Klinteberg, L., Sorgentone, C., Tornberg, A.-K.: Quadrature error estimates for layer potentials evaluated near curved surfaces in three dimensions. arXiv (2020)
5. Berrut, J.-P., Trefethen, L.N.: Barycentric Lagrange interpolation. SIAM Rev. **46**(3), 501–517 (2004). https://doi.org/10.1137/S0036144502417715
6. Kropinski, M., Catherine A., Quaife, B.: Fast integral equation methods for Rothe's method applied to the isotropic heat equation. Comput. Math. Appl. **61**, 2436–2446 (2011). https://doi.org/10.1016/j.camwa.2011.02.024
7. Chen, C.S., Jiang, X., Chen, W., Yao, G.: Fast solution for solving the modified Helmholtz equation with the method of fundamental solutions. Commun. Comput. Phys. **17**(3), 867–886 (2015). https://doi.org/10.4208/cicp.181113.241014a
8. Fryklund, F., Kropinski, M.C., Tornberg, A.-K.: An integral equation–based numerical method for the forced heat equation on complex domains. Adv. Comput. Math. **46**, 69 (2020). https://doi.org/10.1007/s10444-020-09804-z
9. Greengard, L., Kropinski, M.C.: An integral equation approach to the incompressible Navier–Stokes equations in two dimensions. SIAM J. Sci. Comput. **20**(1), 318–336 (1998). https://doi.org/10.1137/S1064827597317648
10. Helsing, J.: Integral equation methods for elliptic problems with boundary conditions of mixed type. J. Comput. Phys. **228**(23), 8892–8907 (2009). https://doi.org/10.1016/j.jcp.2009.09.004
11. Helsing, J., Holst, A.: Variants of an explicit kernel-split panel-based Nyström discretization scheme for Helmholtz boundary value problems. Adv. Comput. Math. **41**(3), 691–708 (2015). https://doi.org/10.1007/s10444-014-9383-y
12. Helsing, J., Jiang, S.: On integral equation methods for the first Dirichlet problem of the biharmonic and modified biharmonic equations in nonsmooth domains. SIAM J. Sci. Comput. **40**(4), A2609–A2630 (2018). https://doi.org/10.1137/17M1162238
13. Helsing, J., Ojala, R.: On the evaluation of layer potentials close to their sources. J. Comput. Phys. **227**(5), 2899–2921 (2008). https://doi.org/10.1016/j.jcp.2007.11.024
14. Jiang, S., Kropinski, M.C.A., Quaife, B.D.: Second kind integral equation formulation for the modified biharmonic equation and its applications. J. Comput. Phys. **249**, 113–126 (2013). https://doi.org/10.1016/j.jcp.2013.04.034
15. Kropinski, M.C.A., Quaife, B.D.: Fast integral equation methods for the modified Helmholtz equation. J. Comput. Phys. **230**(2), 425–434 (2011). https://doi.org/10.1016/j.jcp.2010.09.030
16. NIST. Digital library of mathematical functions. Release 1.0.16 of 2017-09-18. URL http://dlmf.nist.gov/
17. Ojala, R., Tornberg, A.-K.: An accurate integral equation method for simulating multi-phase Stokes flow. J. Comput. Phys. **298**, 145–160 (2015). https://doi.org/10.1016/j.jcp.2015.06.002
18. Powell, M.J.D.: Approximation Theory and Methods. Cambridge U.P., Cambridge (1981). ISBN 0-521-22472-1
19. Vorobjev, Y.N.: Modeling of electrostatic effects in macromolecules. In: L.A. (ed.) Comput. Methods to Study Struct. Dyn. Biomol. Process. From Bioinforma. to Mol. Quantum Mech., pp. 163–202. Springer International Publishing, Cham (2019). ISBN 978-3-319-95843-9. https://doi.org/10.1007/978-3-319-95843-9_6
20. Zhou, H.-X., Pang, X.: Electrostatic interactions in protein structure, folding, binding, and condensation. Chem. Rev. **118**(4), 1691–1741 (2018). https://doi.org/10.1021/acs.chemrev.7b00305

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