



Accurate quadrature of nearly singular line integrals in two and three dimensions by singularity swapping

Ludvig af Klinteberg¹ · Alex H. Barnett²

Received: 28 October 2019 / Accepted: 10 June 2020
© The Author(s) 2020

Abstract

The numerical method of Helsing and co-workers evaluates Laplace and related layer potentials generated by a panel (composite) quadrature on a curve, efficiently and with high-order accuracy for arbitrarily close targets. Since it exploits complex analysis, its use has been restricted to two dimensions (2D). We first explain its loss of accuracy as panels become curved, using a classical complex approximation result of Walsh that can be interpreted as “electrostatic shielding” of a Schwarz singularity. We then introduce a variant that swaps the target singularity for one at its *complexified parameter preimage*; in the latter space the panel is flat, hence the convergence rate can be much higher. The preimage is found robustly by Newton iteration. This idea also enables, for the first time, a near-singular quadrature for potentials generated by smooth curves in 3D, building on recurrences of Tornberg–Gustavsson. We apply this to accurate evaluation of the Stokes flow near to a curved filament in the slender body approximation. Our 3D method is several times more efficient (both in terms of kernel evaluations, and in speed in a C implementation) than the only existing alternative, namely, adaptive integration.

Keywords Nearly singular · Quadrature · Layer potential · Singularity swap

Mathematics Subject Classification 65D30 · 65D32 · 65R20

Communicated by Gunilla Kreiss.

✉ Ludvig af Klinteberg
ludvigak@kth.se

¹ Department of Mathematics, Simon Fraser University, Burnaby, BC, Canada

² Center for Computational Mathematics, Flatiron Institute, New York, NY, USA

1 Introduction

Integral equation methods enable efficient numerical solutions to piecewise-constant coefficient elliptic boundary value problems, by representing the solution as a layer potential generated by a so-called “density” function defined on a lower-dimensional source geometry [4,31]. This work is concerned with accurate evaluation of such layer potentials close to their source, when this source is a one-dimensional curve embedded in 2D or 3D space. In 2D, this is a common occurrence: the curve is the boundary of the computational domain or an interface between different materials. In 3D, such line integrals represent the fluid velocity in non-local slender-body theory (SBT) for filaments in a viscous (Stokes) flow [17,26,28], and also may represent the solution fields in the electrostatic [42] or electromagnetic [11] response of thin wire conductors. Applications in the Stokes case include simulation of straight [45], flexible [35,46] or closed-loop [33] fiber suspensions in fluids. SBT has recently been placed on a more rigorous footing as the small-radius asymptotic limit of a surface boundary integral formulation [30,33,34]. In this work we focus on non-oscillatory kernels arising from Laplace (electrostatic) and Stokes applications, although we expect that by singularity splitting (e.g. [20]) the methods we present could be adapted for oscillatory or Yukawa kernels.

For numerical solutions, the density function must first be discretized [31,44]. A variety of methods are used in practice. If the geometry is smooth enough, and the interacting objects are far enough away that the density is also smooth, then a non-adaptive density representation is sufficient. Such representations include low-order splines on uniform grids [46], global spectral discretization by the periodic trapezoid rule for closed curves [6,18,31], and, for open curves, global Chebychev [11,35] or Legendre [45] expansions. However, if the density has regions where refinement is needed (such as corners or close interactions with other bodies), a composite (“panel”) scheme is better, as is common with Galerkin boundary-element methods [44]. On each panel a high-order representation (such as the Lagrange basis for a set of Gauss–Legendre nodes) is used; panels may then be split in either a graded [24,36,44] or adaptive [37,41,50] fashion.

Our goal is accurate and efficient evaluation of the potential due to a density given on a panel, at arbitrarily close target points. Not only is this crucial for accurate solution evaluation once the density is known, but it is also key to constructing matrix elements for a Nyström or Galerkin solution [31,44] in the case when curves are close to each other [6,23,36]. Specifically, let the panel Γ be an open smooth curve in \mathbb{R}^d , where $d = 2$ or $d = 3$. We seek to evaluate the layer potential

$$u(\mathbf{x}) = \int_{\Gamma} K(\mathbf{x}, \mathbf{y}) f(\mathbf{y}) \, ds(\mathbf{y}), \quad \mathbf{x} \notin \Gamma \subset \mathbb{R}^d, \quad (1)$$

at the target point $\mathbf{x} \notin \Gamma$. Here the density f is a smooth function defined on Γ , and K is a kernel that has a singularity as $\mathbf{x} \rightarrow \mathbf{y}$. In two dimensions the dominant singularity can be either logarithmic,

$$K(\mathbf{x}, \mathbf{y}) \sim k(\mathbf{x} - \mathbf{y}) \log|\mathbf{x} - \mathbf{y}|, \quad (2)$$

or of power-law form

$$K(\mathbf{x}, \mathbf{y}) \sim \frac{k(\mathbf{x} - \mathbf{y})}{|\mathbf{x} - \mathbf{y}|^m}, \quad m = 1, 2, \dots, \tag{3}$$

where m is generally even. Here k denotes smooth (possibly tensor-valued) functions on \mathbb{R}^d . In 3D only the latter form arises, for m generally odd. We will assume that Γ is described by the parametrization $\mathbf{g} : \mathbb{R} \rightarrow \mathbb{R}^d$ that maps the standard interval $[-1, 1]$ to Γ ,

$$\Gamma = \{\mathbf{g}(t) \mid t \in [-1, 1]\}. \tag{4}$$

Thus the parametric form of the desired integral (1) is

$$u(\mathbf{x}) = \int_{-1}^1 K(\mathbf{x}, \mathbf{g}(t)) \tilde{f}(t) |\mathbf{g}'(t)| dt, \quad \mathbf{x} \notin \Gamma \subset \mathbb{R}^d, \tag{5}$$

where $\tilde{f}(t) := f(\mathbf{g}(t))$ is the pullback of the density to parameter space.

The difficulty of evaluation of a layer potential with given density is essentially controlled by the distance of the target point \mathbf{x} from the curve. In particular, if \mathbf{x} is far from Γ , the kernel $K(\mathbf{x}, \mathbf{y})$ varies, as a function of \mathbf{y} , no more rapidly than the curve geometry itself. Thus a fixed quadrature rule, once it integrates accurately the product of density f and any geometry functions (such as the “speed” $|\mathbf{g}'|$), is accurate for all such targets. Here, “far” can be quantified relative to the local quadrature node spacing h and desired accuracy: e.g. in the 2D periodic trapezoid rule setting the distance should be around $(\log 10)h/2\pi \approx 0.37 h$ times the desired number of correct digits (a slight extension of [7, Remark 2.6]). However, as \mathbf{x} approaches Γ , the kernel $K(\mathbf{x}, \mathbf{y})$ becomes increasingly non-smooth in the region where \mathbf{y} is near \mathbf{x} , and any fixed quadrature scheme loses accuracy. This is often referred to as the layer potential integral being *nearly singular*.

There are several approaches to handling the nearly singular case. The most crude is to increase the number of nodes used to discretize the density; this has the obvious disadvantage of growing the linear system size beyond that needed to represent the density, wasting computation time and memory. Slightly less naive is to preserve the original discretization nodes, but interpolate from these nodes onto a refined quadrature scheme (with higher order or subdivided panels [46]) used only for potential evaluations. However, to handle arbitrarily close targets one has to refine the panels in an adaptive fashion, *afresh for each target point*. As a target approaches Γ an increasing level of adaptive refinement is demanded. The large number of conditionals and interpolations needed in such a scheme can make it slow.

This has led to more sophisticated “special” rules which do not suffer as the target approaches the curve. They exploit properties of the PDE, specifically, in 2D, the relation of harmonic to analytic functions. In the 2D high-order setting, several important contributions have been made in the last decade, including panel-based kernel-split quadrature [19–21,23,36], globally compensated spectral quadrature [6,23],

quadrature by expansion (QBX) [7,29,48], harmonic density interpolation [39], and asymptotic expansion [12]. In 3D, little work has been done on high-order accurate evaluation of nearly singular line integrals. Many 3D applications are in the context of active or passive filaments in viscous fluids, but current quadrature methods are limited to either using analytical formulae for straight segments [45], or regularizing the integrand using an artificial length scale [13,25,46].

In this paper, we propose a new panel-based quadrature method for nearly singular line integrals in both 2D and 3D. The method incurs no loss of efficiency for targets arbitrarily close to the curve. (Note that, in the 2D case it is usual for one-sided limits on Γ to exist, but in 3D limits on Γ do not generally exist.) Our method builds on the panel-based monomial recurrences introduced by Helsing–Ojala [23] and extended by Helsing and co-workers [19,20,22,36] [21, Sec. 6], with a key difference that, instead of interpolating in the physical plane (which is associated with the complex plane), we interpolate in the (real) parametrization variable of the panel. Rather than exploiting Cauchy’s theorem in the physical plane, the singularity is “swapped” by cancellation for one in the parameter plane; this requires a nearby root-finding search for the (analytic continuation of the) function describing the distance between the panel and the target. It builds on recent methods for computing quadrature error estimates in two dimensions [3]. We include a robust and efficient method for such root searches.

This new formulation brings two major advantages:

1. The rather severe loss of accuracy of the Helsing–Ojala method for curved panels—which we quantify for the first time via classical complex approximation theory—is much ameliorated. As we demonstrate, more highly curved panels may be handled with the same number of nodes, leading to higher efficiency.
2. The method extends to arbitrary smooth curves in 3D. Remarkably, the search for a root in the complex plane is unchanged from the 2D case.

As a side benefit, our method automatically generates the information required to determine whether it is needed, or whether direct quadrature using the existing density nodes is adequate, using results in [2,3].

The structure of this paper is as follows. Section 2 is concerned with the 2D problem; we give the necessary background and introduce our new “singularity swap” method. Section 3 then extends the method to deal with problems in 3D. Section 4 summarizes the entire proposed algorithm, unifying the 2D and 3D cases. Section 5 presents further numerical tests of accuracy in 2D, and accuracy and efficiency in a 3D Stokes slender body application, with a comparison against standard adaptive quadrature. We conclude in Sect. 6.

2 Two dimensions

This section gives the necessary background and introduces our method for problems in 2D. Section 2.1 reviews the convergence rate for direct evaluation of the potential, i.e. using a direct quadrature scheme. This also serves to introduce complex notation and motivate special quadratures. Section 2.2 summarizes the complex monomial interpolatory quadrature of Helsing–Ojala [19,23] for evaluation of a given density

close to its 2D source panel. In Sect. 2.3 we explain and quantify the loss of accuracy in the aforementioned method due to panel bending. Finally in Sect. 2.4 we use the “singularity swap” to make a real monomial version that is less sensitive to bending, and which will form the basis of the 3D method.

2.1 Summary of the direct evaluation error near a panel

Here we review known results about the approximation of the integral (5) directly using a fixed high-order quadrature rule. In the analytic panel case this is best understood by analytic continuation in the parameter. Let t_j be the nodes and w_j the weights for an n -node Gauss–Legendre scheme on $[-1, 1]$, at which we assume the density is known.¹ Substituting the rule into (5) gives the direct approximation

$$u(\mathbf{x}) \approx \sum_{j=1}^n W_j K(\mathbf{x}, \mathbf{y}_j) f_j, \tag{6}$$

where the nodes are $\mathbf{y}_j = \mathbf{g}(t_j)$, the density samples $f_j = f(\mathbf{y}_j)$, and the modified weights $W_j = w_j |\mathbf{g}'(t_j)|$. For any integrand $F(t)$ analytic in a complex neighborhood of $[-1, 1]$, the error

$$R_n[F] := \sum_{j=1}^n w_j F(t_j) - \int_{-1}^1 F(t) dt \tag{7}$$

vanishes exponentially with n , with a rate which grows with the size of this neighborhood. Specifically, define E_ρ , the Bernstein ρ -ellipse, as the image of the disc $|z| < \rho$ under the Joukowski map

$$t = \frac{z + z^{-1}}{2}. \tag{8}$$

(For example, Fig. 1a shows this ellipse for $\rho \approx 1.397$.) Then, if F is analytic and bounded in E_ρ , there is a constant C depending only on ρ and F such that

$$|R_n[F]| \leq C \rho^{-2n}, \quad \text{for all } n = 1, 2, \dots \tag{9}$$

An explicit bound [47, Thm. 19.3] is $C = [64/15(\rho^2 - 1)] \sup_{t \in E_\rho} |F(t)|$.

Returning to (5), the integrand of interest is $F(t) = K(\mathbf{x}, \mathbf{g}(t)) \tilde{f}(t) |\mathbf{g}'(t)|$, for which we seek the largest ρ such that F is analytic in E_ρ . Bringing \mathbf{x} near Γ induces a singularity near $[-1, 1]$ in the analytic continuation in t of $K(\mathbf{x}, \mathbf{g}(t))$, a claim justified as follows. Each of the kernels $K(\mathbf{x}, \mathbf{g}(t))$ under consideration has a singularity when

¹ We choose this rule since it is the most common; however, any other high-order rule with an asymptotic Chebyshev density on $[-1, 1]$ would behave similarly.

the squared distance

$$R(t)^2 := |\mathbf{x} - \mathbf{g}(t)|^2 = (x_1 - g_1(t))^2 + (x_2 - g_2(t))^2 \quad (10)$$

goes to zero, which gives $x_1 - g_1(t) = \pm i(x_2 - g_2(t))$. Taking the negative case, the singularity (denoted by t_0) thus obeys

$$x_1 + ix_2 = g_1(t_0) + ig_2(t_0), \quad (11)$$

and one may check by Schwarz reflection that the positive case gives its conjugate \bar{t}_0 . This suggests identifying \mathbb{R}^2 with \mathbb{C} and introducing

$$\zeta = x_1 + ix_2, \quad (\text{target point}) \quad (12)$$

$$\tau = y_1 + iy_2, \quad (\text{source point on } \Gamma) \quad (13)$$

$$\gamma(t) = g_1(t) + ig_2(t), \quad (\text{complex parameterization of } \Gamma). \quad (14)$$

Thus, in this complex notation, the Eq. (11) for t_0 is written

$$\zeta = \gamma(t_0). \quad (15)$$

For now we assume that $\gamma : \mathbb{C} \rightarrow \mathbb{C}$ is analytic in a sufficiently large complex neighborhood of $[-1, 1]$, so the nearest singularity is $t_0 = \gamma^{-1}(\zeta)$, the preimage of ζ under γ ; see upper and lower panels of Fig. 1a. Thus our claim is proved.

Inverting (8) gives the elliptical radius ρ of the Bernstein ellipse on which any $t \in \mathbb{C}$ lies,

$$\rho(t) := |t \pm \sqrt{t^2 - 1}|, \quad \text{with sign chosen such that } \rho > 1. \quad (16)$$

We then have, for any of the kernels under study, that (9) holds for any $\rho < \rho(\gamma^{-1}(\zeta))$. Thus images of the Bernstein ellipses, $\gamma(E_\rho)$, control contours of equal exponential convergence rate, and thus (assuming that the prefactor C does not vary much with the target point), at fixed n , also the contours of equal error magnitude. A more detailed analysis for the Laplace double-layer case in [2] [3, Sec. 3.1] improves² the rate to $\rho = \rho(\gamma^{-1}(\zeta))$, predicts the constant, and verifies that the error contours very closely follow this prediction. Indeed, the lower panel of our Fig. 1a illustrates, for an analytic choice of density f , that the contour of constant error (modulo oscillatory “fingering” [3]) well matches the curve $\gamma(E_\rho)$ passing through the target ζ . The top-left plots in Figs. 2 and 3 show the shrinkage of these error contours (images of Bernstein ellipses) as n , the number of nodes on the panel, grows.

For kernels other than the Laplace double-layer, error results are similar [2,29]. This quantified breakdown in accuracy as the target approaches Γ motivates the special quadratures to which we now turn.

² In that work the tool was instead the asymptotics of the characteristic remainder function of Donaldson–Elliott [15], and $\rho(t_0)$ was left in the form (16).

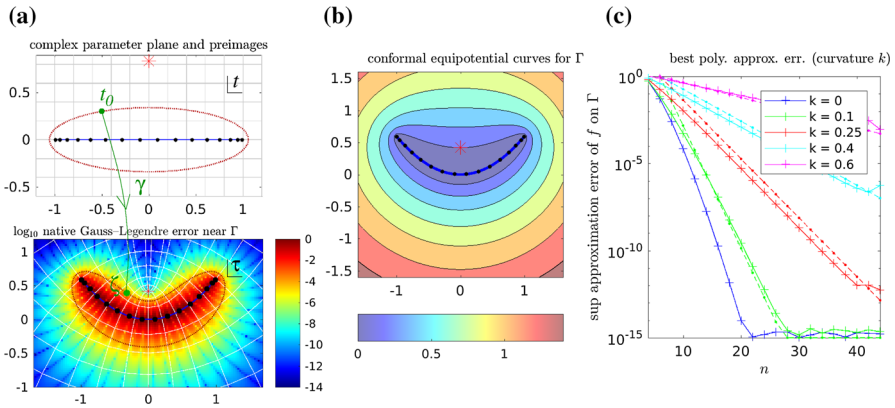


Fig. 1 Direct and complex-interpolatory evaluation errors for a simple parabolic panel Γ , for I_1 (complex double-layer) with density $\tilde{f}(t) = \sin(1 + 3t)$ as a function of parameter t . **a** t -plane preimage $[-1, 1]$ (top plot) mapping to panel Γ (bottom plot) via $\tau = \gamma(t) = t + ik t^2$ with curvature $k = 0.6$, hence minimum radius of curvature $R = 1/2k \approx 0.83$. Also shown are: $n = 16$ Gauss–Legendre points and (bottom with colormap) their error in potential evaluation using the direct scheme; an example target ζ and its preimage $t_0 = \gamma^{-1}(\zeta)$ (large green dots); Bernstein ellipse for t_0 and its image (dotted red); Schwarz singularity $\tau_* = i/4k$ and its preimage $t_* = i/2k$ (red *’s); and Cartesian grid lines (grey) in the t -plane and their images (white). **b** Equipotential curves of $g(z)$ for the same panel Γ , showing “shielding” of the Schwarz singularity (red *) thus small $g \approx 0.2$ and $\rho \approx 1.22$. Contrast $\rho \approx 2.14$ for the Schwarz preimage in the t -plane. **c** Convergence in n of polynomial approximation error of f over Γ (solid lines) for several curvatures k , compared to $C\rho^{-n}$ (dotted lines) predicted by Sect. 2.3 (color figure online)

2.2 Interpolatory quadrature using a complex monomial basis

We now review the special quadrature of Helsing and coworkers [19,23,36] that approximates (1) in the 2D case. In the complex notation of (14), given a smooth density defined on Γ , for all standard PDEs of interest (Laplace, Helmholtz, Stokes, etc), the integrals that we need to evaluate can be written as the contour integrals

$$I_L = I_L(\zeta) := \int_{\Gamma} f(\tau) \log(\tau - \zeta) d\tau, \tag{17}$$

$$I_m = I_m(\zeta) := \int_{\Gamma} \frac{f(\tau)}{(\tau - \zeta)^m} d\tau, \quad m = 1, 2, \dots \tag{18}$$

“Appendix A” reviews how to rewrite common boundary integral kernels as such contour integrals; note that f may include other smooth factors than merely the density. Other 2nd-order elliptic 2D PDE kernels may be split into terms of the above form, as pioneered by Helsing and others in the Helmholtz [20], axisymmetric Helmholtz [22], elastostatics [19, Sec. 9], Stokes [37], and (modified) biharmonic [21, Sec. 6] cases.

An innovation of the method was to interpolate the density in the complex coordinate τ rather than the parameter t . Thus,

$$f(\tau) \approx \sum_{k=1}^n c_k \tau^{k-1}, \quad \tau \in \Gamma. \tag{19}$$

This is most numerically stable if for now we assume that Γ has been transformed by rotation and scaling to connect the points ± 1 . Here the (complex) images of the quadrature nodes are $\{\tau_j\}_{j=1}^n := \{\gamma(t_j)\}_{j=1}^n$. The monomial coefficients c_k can be found by collocation at these nodes, as follows. Let \mathbf{c} be the column vector with entries $\{c_k\}$, \mathbf{f} be the vector of values $\{f(\tau_j)\}$, and A be the Vandermonde matrix with entries $A_{ij} = \tau_i^{j-1}$, for $i, j = 1, \dots, n$. Then enforcing (19) at the nodes results in the $n \times n$ linear system

$$A\mathbf{c} = \mathbf{f} \quad (20)$$

which can be solved for the vector \mathbf{c} . Note that working in this *monomial* basis, even in the case of τ_j on the real axis, is successful even up to quite high orders, although this is rarely stated in textbooks.

Remark 1 (Backward stability) It is known for $\{\tau_j\}$ being any set of real or complex nodes that is not close to the roots of unity that the condition number of the Vandermonde matrix A in (20) must grow at least exponentially in n [16,38]. However, as pointed out in [23, App. A], extreme ill-conditioning in itself introduces no loss of interpolation accuracy, at least for $n < 50$. Specifically, as long as a vector \mathbf{c} has small relative residual $\|A\mathbf{c} - \mathbf{f}\|/\|\mathbf{f}\|$, then the resulting monomial is close to the data at the nodes. Assuming weak growth in the Lebesgue constant with n , the interpolant is thus uniformly accurate. Such a small residual norm, if it exists for the given right-hand side, is found by a solver that is backward stable. We recommend either partially pivoted Gaussian elimination (e.g. as implemented by `mldivide` in MATLAB) which is $\mathcal{O}(n^3)$, or the Björck–Pereyra algorithm [10] which is $\mathcal{O}(n^2)$.

As with any interpolatory quadrature, once the coefficients $\{c_k\}$ have been computed, the integrals (17) and (18) can be approximated as

$$I_L \approx \sum_{k=1}^n c_k q_k(\zeta), \quad (21)$$

$$I_m \approx \sum_{k=1}^n c_k p_k^m(\zeta), \quad (22)$$

where the values $\{q_k\}$ and $\{p_k^m\}$ are the exact integrals of the monomial densities $\{\tau^{k-1}\}$ multiplied with the logarithmic and power-law kernels respectively,

$$q_k(\zeta) = \int_{\Gamma} \tau^{k-1} \log(\tau - \zeta) \, d\tau, \quad k = 1, \dots, n, \quad (23)$$

$$p_k^m(\zeta) = \int_{\Gamma} \frac{\tau^{k-1}}{(\tau - \zeta)^m} \, d\tau, \quad k = 1, \dots, n. \quad (24)$$

The benefit of this (somewhat unusual) complex monomial basis is that these integrals can be efficiently evaluated through *recursion formulae*, as follows.

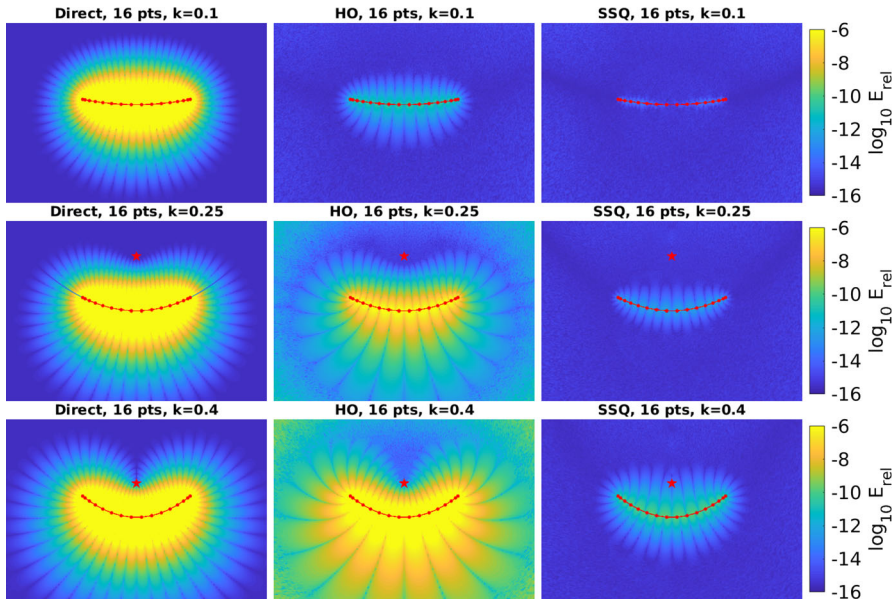


Fig. 2 Spatial plot of relative errors in 2D quadratures for the Laplace double-layer (42) on Γ a parabolic panel $g(t) = (t, kt^2)$, $-1 \leq t \leq 1$, with smooth density $\rho(y) = y_1 y_2$, using $n = 16$ nodes. Each row shows a different curvature k . Left column: direct quadrature of Sect. 2.1. Middle column: Helsing–Ojala (HO) quadrature of Sect. 2.2. Right column: proposed singularity swap quadrature (SSQ) of Sect. 2.4. The red $*$ is the Schwarz singularity of Γ

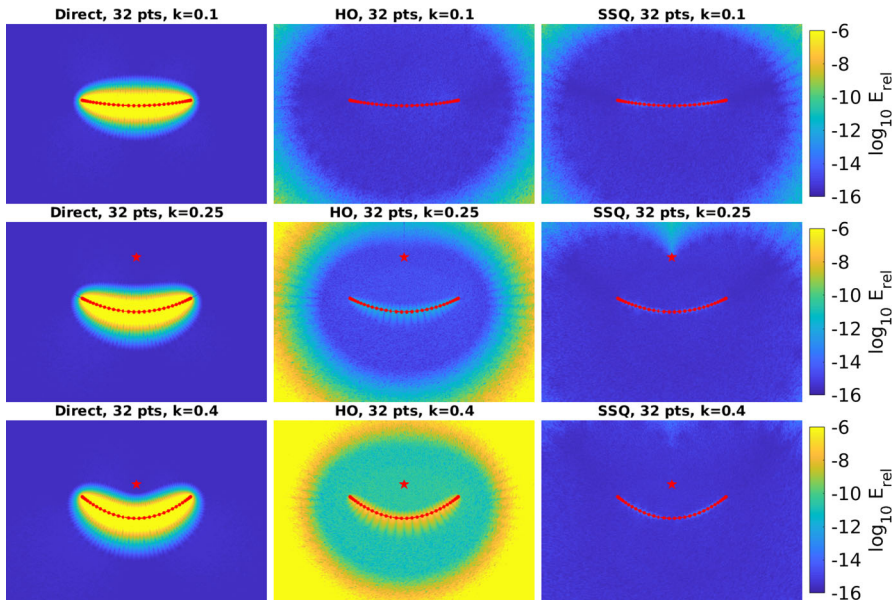


Fig. 3 Same as Fig. 2, after interpolating panel data to $n = 32$ nodes

2.2.1 Exact evaluation of p_k^m and q_k by recurrence

Recall that Γ has been transformed by rotation and scaling to connect the points ± 1 . We first review the Cauchy kernel $m = 1$ case [23, Sec. 5.1]. The contour integral for $k = 1$ exploits independence of the path Γ :

$$p_1^1(\zeta) = \int_{\Gamma} \frac{d\tau}{\tau - \zeta} = \log(1 - \zeta) - \log(-1 - \zeta) \pm 2\pi i \mathcal{N}_{\zeta}. \quad (25)$$

Here $\mathcal{N}_{\zeta} \in \mathbb{Z}$ is a winding number that, for the standard branch cut of the logarithm, is zero if ζ is outside the domain enclosed by the oriented closed curve comprising Γ traversed forwards plus $[-1, 1]$ traversed backwards, while $\mathcal{N}_{\zeta} = +1$ (-1) when ζ is inside a part of this domain that is enclosed counterclockwise (clockwise) [23, Sec. 7]. The following 2-term recurrence is derived by adding and subtracting $\zeta \tau^{k-1}$ from the numerator of the formula (24):

$$p_{k+1}^1(\zeta) = \zeta p_k^1(\zeta) + \frac{1 - (-1)^k}{k}, \quad k > 1. \quad (26)$$

One can then recur upwards to get p_m^1 for all k . For $m > 1$ we get analogously,

$$p_1^m(\zeta) = \frac{(1 - \zeta)^{1-m} - (-1 - \zeta)^{1-m}}{1 - m}, \quad (27)$$

$$p_{k+1}^m(\zeta) = \zeta p_k^m(\zeta) + p_k^{m-1}(\zeta), \quad k > 1. \quad (28)$$

Thus once all values for $m - 1$ are known, they can be found for m by upwards recurrence. Note that in [19, Sec. 2.2], a different recursion was used for the case $m = 2$. Finally, recursive computation of the complex logarithmic kernel follows directly, following [19, Sec. 2.3],

$$q_k(\zeta) = \frac{1}{k} \left(\log(1 - \zeta) - (-1)^k \log(-1 - \zeta) - p_{k+1}^1(\zeta) \right). \quad (29)$$

We emphasize that these simple exact path-independent formulae are enabled by the choice of the complex monomial basis in the coordinate τ .

Remark 2 (Transforming for general panel endpoints) We now review the changes that are introduced to the answers I_L and I_m when rescaling and rotating a general panel Γ parameterized by $\gamma(t)$, $t \in [-1, 1]$, into the required one which connects ± 1 , and similarly transforming the target. Define the complex scale factor $s_0 := (\gamma(1) - \gamma(-1))/2$ and origin $\tau_0 = (\gamma(1) + \gamma(-1))/2$. Then the affine map from a given location τ to the transformed location $\tilde{\tau}$, given by

$$\tilde{\tau} = s(\tau) := \frac{\tau - \tau_0}{s_0}, \quad (30)$$

can be applied to all points on the given panel and to the target to give a transformed panel $\tilde{\Gamma}$ and target $\tilde{\zeta}$. From this, applying the formulae above one can evaluate \tilde{I}_L and \tilde{I}_m . By inserting the change of variables one then gets the desired potentials due to the density on Γ at target ζ ,

$$I_L = s_0 \tilde{I}_L + (\log s_0) \int_{\Gamma} f(\tau) \, d\tau, \quad \text{and} \quad I_m = s_0^{1-m} \tilde{I}_m, \quad m = 1, 2, \dots$$

Here a new integral is needed (the total “charge”), easily approximated by $\sum_{j=1}^n w_j |\gamma'(t_j)| f(\tau_j)$, where w_j are the Gauss–Legendre weights for $[-1, 1]$.

2.2.2 The adjoint method for weights applying to arbitrary densities

The above procedure computes I_L and I_m of (17)–(18) for a specific function f (such as the density) on Γ , using its samples \mathbf{f} on the nodes to solve first for a coefficient vector \mathbf{c} . In practice it is more useful instead to compute the weight vectors $\boldsymbol{\lambda}^L := \{\lambda_j^L\}_{j=1}^n$ and $\boldsymbol{\lambda}^m := \{\lambda_j^m\}_{j=1}^n$ which, for any vector \mathbf{f} , approximate I_L and I_m when their inner products are taken against \mathbf{f} . That is, $I_L \approx \mathbf{f}^T \boldsymbol{\lambda}^L$ and $I_m \approx \mathbf{f}^T \boldsymbol{\lambda}^m$, where T indicates non-conjugate transpose. One application is for filling near-diagonal blocks of the Nyström matrix arising in a boundary integral formulation: $(\boldsymbol{\lambda}^L)^T$ and $(\boldsymbol{\lambda}^m)^T$ form row blocks of the matrix.

We review a result presented in [23, Eq. (51)] (where a faster numerical method was also given in the $m = 1$ case). Let $\boldsymbol{\lambda}$ be either $\boldsymbol{\lambda}^L$ or $\boldsymbol{\lambda}^m$, let I be the corresponding integral (17) or (18), and let \mathbf{p} be the corresponding vector $\{q_k\}$ or $\{p_k^m\}$ as computed in Sect. 2.2.1. Writing our requirement for $\boldsymbol{\lambda}$, we insert (20) to get

$$I \approx \mathbf{f}^T \boldsymbol{\lambda} = \mathbf{c}^T A^T \boldsymbol{\lambda} = \mathbf{c}^T \mathbf{p}, \tag{31}$$

where the last form simply expresses (21) or (22). But this last equality must hold for all $\mathbf{c} \in \mathbb{C}^n$, thus

$$A^T \boldsymbol{\lambda} = \mathbf{p}, \tag{32}$$

which is a linear system whose solution is the weight vector $\boldsymbol{\lambda}$. Notice that this is an adjoint method [27] to compute $\boldsymbol{\lambda}$.

2.3 The effect of panel curvature on Helsing–Ojala convergence rate

The Helsing–Ojala method reviewed above is easy to implement, computationally cheap, and can achieve close to full numerical precision for target points very close to Γ . However, there is a severe loss of accuracy as Γ becomes curved. This is striking in the central column of plots in Fig. 2. This observation led Helsing–Ojala [23] and later users to recommend upsampling from a baseline $n = 16$ to a larger n , such as $n = 32$, even though the density and geometry were already accurately resolved at $n = 16$. The “HO” $k = 0.4$ plot in Fig. 3 shows that even using $n = 32$ has loss of accuracy

near a moderately curved panel. This forces one to split panels beyond what is needed for the density representation, just to achieve good evaluation accuracy, increasing the cost.

The error in this method is essentially entirely due to the error of the polynomial approximation (19) of f on Γ , because each monomial is integrated exactly (up to rounding error) as in Sect. 2.2.1. For this error there is a classical result that the best polynomial approximation converges geometrically in the degree, with rate controlled by the largest *equipotential curve* of Γ inside of which f can be analytically continued. Namely, given the set Γ , let g be the unique solution to the potential problem in \mathbb{R}^2 (which we identify with \mathbb{C}),

$$\begin{aligned} \Delta g &= 0 && \text{in } \mathbb{R}^2 \setminus \Gamma \\ g &= 0 && \text{on } \Gamma \\ g(\mathbf{x}) &\sim \log |\mathbf{x}| && \text{as } |\mathbf{x}| \rightarrow \infty. \end{aligned}$$

For each $\rho > 1$, define³ the level curve $C_\rho := \{\tau \in \mathbb{C} : g(\tau) = \log \rho\}$. An example g and its equipotential curves are shown in Fig. 1b; note that they are very different from the Bernstein ellipse images in the lower plot of Fig. 1a.

Theorem 3 (Walsh [49, §4.5]) *Let $\Gamma \subset \mathbb{C}$ be a set whose complement (including the point at infinity) is simply connected. Let $f : \Gamma \rightarrow \mathbb{C}$ extend analytically to a function analytic inside and on C_ρ , for some $\rho > 1$. Then there is a sequence of polynomials $p_n(z)$ of degree $n = 0, 1, \dots$ such that*

$$|p_n(z) - f(z)| \leq C\rho^{-n} \quad z \in \Gamma, \text{ for all } n = 0, 1, \dots$$

where C is a constant independent of n and z .

To apply this to (19), we need to know the largest region in which f may be continued as an analytic function, which is in general difficult. In practical settings the panels will have been refined enough so that on each panel preimage \tilde{f} is interpolated in the parameter $t \in [-1, 1]$ to high accuracy, thus we may assume that any singularities of \tilde{f} are distant from $[-1, 1]$. Yet, in moving from the t to τ plane, the *shape of the panel Γ itself* can introduce new singularities in f which will turn out to explain well the observed loss of accuracy, as follows.

Proposition 4 (geometry-induced singularity in analytic continuation of the density) *Let the pullback \tilde{f} be a generic function analytic at t_* . Let $f(\tau) = \tilde{f}(t)$ where $\tau = \gamma(t)$. Let $t_* \in \mathbb{C}$ be such that $\gamma'(t_*) = 0$ and γ is analytic at t_* . Then generically $f(\tau)$ is not analytic at $\tau_* = \gamma(t_*)$.*

Proof Since the derivative vanishes, the Taylor expansion of the parameterization at t_* takes the form $\gamma(t) = \tau_* + a_2(t - t_*)^2 + \dots$. If f were analytic at τ_* , expanding gives $f(\tau) = f_0 + f_1(\tau - \tau_*) + \dots$, then inserting $\tau = \gamma(t)$ gives $f_0 + f_1 a_2(t - t_*)^2 + \dots$ which must match, term by term, the Taylor expansion of the pullback $\tilde{f}(t) = \tilde{f}_0 + \tilde{f}_1(t - t_*) + \dots$. But generically $\tilde{f}_1 \neq 0$, in which case there is a contradiction. \square

³ Note that the reuse of the radius symbol ρ from Sect. 2.1 is deliberate, since in the special case $\Gamma = [-1, 1]$, C_ρ is the boundary of the Bernstein ellipse E_ρ .

Similar (more elaborate) arguments are known for singularities in the extension of Helmholtz solutions [32]. Here τ_* is an example of a singularity of the Schwarz function [14,43] for the curve Γ . Recall that the Schwarz function G is defined by $\overline{G(\tau)} = \gamma(\gamma^{-1}(\tau))$, which is the analytic reflection of τ through the arc Γ . At points where $\gamma' = 0$, the inverse map γ^{-1} , hence the Schwarz function, has a square-root type branch singularity. Figure 1a shows that for a moderately bent parabolic panel, the Schwarz singularity is quite close to the concave side. Note that the above argument relies on \tilde{f} having a generic expansion, which we believe is typical; we show below that its convergence rate prediction holds well in practice.

In summary, smooth but curved panels often have a nearby Schwarz singularity, which generically induces a singularity at this same location in the analytic continuation of the density f ; then the equipotential for Γ at this location controls the best rate of polynomial approximation, placing a fundamental limit on the Helsing–Ojala convergence rate in n . Figure 1b shows that, for a moderately bent panel, the close singularity is effectively *shielded electrostatically* by the concave panel, thus the convergence rate $\rho = e^{g(\tau)}$ is very small (close to 1). In Fig. 1c we test this quantitatively: for a parabolic panel and simple analytic pullback density \tilde{f} we compare the maximum error in n -term polynomial approximation on Γ with the prediction from the above Schwarz–Walsh analysis. The agreement in rate is excellent over a range of curvatures. This also matches the loss of digits seen in the central column of Figs. 2 and 3: e.g. for even the mild curvature $k = 0.25$, $n = 16$ achieves only 5 digits, and $n = 32$ only 10 digits.

Conjecture 5 (Schwarz singularities at focal points) *Note that in the parabola example of Fig. 1, a square-root type Schwarz singularity appears near Γ 's highest curvature point, at distance $R/2$ on the concave side, where R is the radius of curvature. This is also approximately true for ellipses, whose foci induce singularities [43, §9.4]. We also always observe this numerically for other analytic curves. Hence we conjecture that it is a general result for analytic curves, in an approximate sense that becomes exact as $R \rightarrow 0$.*

2.4 Interpolatory quadrature using a real monomial basis

We now propose a “singularity swap” method which pushes the interpolation problem to the panel preimage $t \in [-1, 1]$, thus bypassing the rather severe effects of curvature just explained. Recalling that the target point is ζ , we first write the integrals (17) and (18) in parametric form

$$I_L = \int_{-1}^1 h(t) \log Q(t) \, dt, \tag{33}$$

$$I_m = \int_{-1}^1 \frac{h(t)}{Q(t)^m} \, dt, \quad m = 1, 2, \dots, \tag{34}$$

where we have introduced the displacement function Q and the smooth function h ,

$$Q(t) := \gamma(t) - \zeta, \quad (35)$$

$$h(t) := \tilde{f}(t) |\gamma'(t)|. \quad (36)$$

Now let t_0 be the root of Q nearest to $[-1, 1]$. (Unless the panel is very curved, there is only one such nearby root and it is simple.) Then $Q(t)/(t - t_0)$ has a removable singularity at t_0 , and hence it and its reciprocal are analytic in a larger neighborhood of $[-1, 1]$ than the above integrands. By multiplying and dividing by $(t - t_0)$,

$$I_L = \int_{-1}^1 h(t) \log \frac{Q(t)}{t - t_0} dt + \int_{-1}^1 h(t) \log(t - t_0) dt, \quad (37)$$

$$I_m = \int_{-1}^1 \frac{h(t)(t - t_0)^m}{Q(t)^m} \frac{1}{(t - t_0)^m} dt, \quad m = 1, 2, \dots \quad (38)$$

which, as we detail below, can be handled as special cases of (17) and (18) for a new flat panel $\Gamma = [-1, 1]$ written in the t plane. We have “swapped” the singularity from the τ to the t plane. As Fig. 1 illustrates, the preimage of the Schwarz singularity has a much higher conformal distance from $[-1, 1]$ than the actual singularity in the τ -plane has from Γ , indicating much more rapid convergence.

Thus we now apply the Helsing–Ojala methods of Sect. 2.2, but to $[-1, 1]$ in the t plane. Namely, as before, let t_j and w_j , $j = 1, \dots, n$ be the Gauss–Legendre quadrature for $[-1, 1]$. The first integral in (37) is smooth, so direct quadrature is used. The second integral in (37) is evaluated via (21) and (29), setting $\Gamma = [-1, 1]$, with the coefficients c_k for the function $h(t)$ found by solving the Vandermonde system as in Sect. 2.2. The first term $h(t)(t - t_0)^m/Q(t)^m$ in (38) is smooth and plays the role of $f(\tau)$ in (18), so its coefficients are found similarly. Equation (38) is then approximated by (22) with (25)–(28).

Remark 6 Since in this scheme the new flat “panel” $[-1, 1]$ is fixed, one may LU factorize the Vandermonde matrix once and for all, then use forward and back-substitution to solve for the coefficients $\{c_k\}$ given the n samples of each of the two smooth functions, namely $\{h(t_j)\}$ and $\{h(t_j)(t_j - t_0)^m/Q(t_j)^m\}$, with only $\mathcal{O}(n^2)$ effort.

Remark 7 With a very curved panel it is possible that more than one root of $Q(t)$ is relevant; see Fig. 4. In such a case where poles z_1 and z_2 need to be cancelled, we have derived recursion formulae for $p_k^m(z_1, z_2)$ and $q_k(z_1, z_2)$ which generalize those in Sect. 2.2.1. However, we have not found these to be needed in practice, so omit them.

2.4.1 Finding the roots of $Q(t)$

The only missing ingredient in the scheme just described is to find the nearest complex root t_0 satisfying

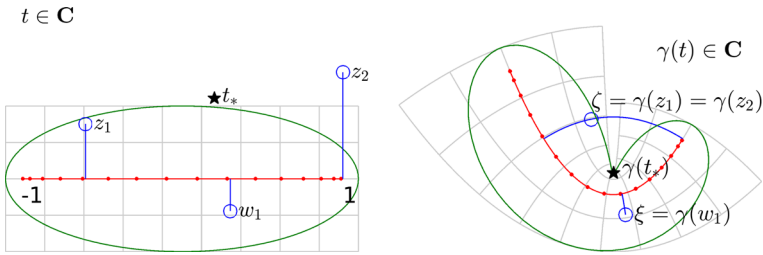


Fig. 4 Illustration of the relationship between the roots of the displacement $Q(t)$, to the left, and target points in 2D physical space, to the right. The point ξ corresponds to only one nearby root, w_1 , while the point ζ corresponds to two nearby roots, z_1 and z_2 . Note that only z_1 lies inside the Bernstein ellipse of radius ρ_ϵ , marked green, such that it requires singularity swapping. The point $\gamma(t_*)$ is the Schwarz singularity of the curve, marking the point where γ^{-1} stops being single-valued

$$Q(t_0) := \gamma(t_0) - \zeta = 0, \tag{39}$$

i.e., the preimage of the (complex) target point ζ under the complexification of the panel map γ ; see Fig. 4. We base our method on that of [3] (where roots were needed only for the purposes of error estimation). Let $P_n[\gamma](t)$ denote a degree $n - 1$ polynomial approximant to $\gamma(t)$ on $[-1, 1]$. Using the data at the Legendre nodes $\{t_j\}$, forming $P_n[\gamma](t)$ is both well-conditioned and stable if we use a Legendre (or Chebyshev) expansion, and has an $\mathcal{O}(n^2)$ cost. See [3] for details on how to use a Legendre expansion. Continuing each basis function to \mathbb{C} gives a complex approximation

$$\tilde{Q}(t) := P_n[\gamma](t) - \zeta, \tag{40}$$

for which we seek roots nearest $[-1, 1]$. Given a nearby starting guess, a single root of \tilde{Q} can be found using Newton’s method, which converges rapidly and costs $\mathcal{O}(n)$ per iteration. A good starting guess is $t_{\text{init}} \approx s(\zeta)$, recalling that s defined by Eq. (30) maps the panel endpoints to ± 1 . With this initialization, Newton iterations converge to the nearest root in most practical cases. The iterations can, however, be sensitive to the initial guess when two roots are both relatively close to $[-1, 1]$, and sometimes converge to the second-nearest root. As illustrated in Fig. 4, this typically happens only when the target point is on the concave side of a very curved panel.

A more robust, but also more expensive, way of finding the nearest root is to first find all $n - 1$ roots of \tilde{Q} at the same time, and then pick the nearest one. This can be done using a matrix-based method, which finds the roots as the eigenvalues to a generalized companion (or “comrade”) matrix [8]. This has a cost that is $\mathcal{O}(n^3)$ if done naively, and $\mathcal{O}(n^2)$ if done using methods that exploit the matrix structure [5].

In order to determine whether Newton iterations or a matrix-based method should be used for finding the root t_0 , we suggest a criterion based on the distance to the nearest Schwarz singularity preimage of the panel, t_* , as this is a measure of the size

of the region where γ^{-1} is single-valued. Let ρ_ϵ be a Bernstein radius beyond which direct Gauss–Legendre quadrature is expected to have relative error ϵ . Then, ignoring the prefactor in (9) and solving for ρ ,

$$\rho_\epsilon := \epsilon^{-1/2n} \quad (41)$$

is a useful estimate. Recalling (16), then $\rho(t_*) \leq C\rho_\epsilon$, where we choose the constant $C = 1.1$, indicates that there may be more than one nearby root, and that it is safer to use a matrix-based root finding method. For each panel, t_* can be found at the time of discretization, by applying Newton’s method to the polynomial $P_n[\gamma'](t) = 0$, with the initial guess $t_* \approx 0$.

Remark 8 By switching to matrix-based root finding when there is a nearby Schwarz singularity, we get a root finding process that is robust even in the case of a very curved panel. However, in practical applications we have not observed it to be necessary, e.g. for the results of Sect. 5. It is on the other hand necessary in the examples in Figs. 2 and 3, because there we test the specialized quadrature at distances well *beyond* where it is required.

Remark 9 The root finding process can be made efficient by computing the interpolants $P_n[\gamma](t)$ and $P_n[\gamma'](t)$ for each segment at the time of discretization, such that they can be reused for every target point ζ .

2.4.2 Tests of improved convergence rate for curved panels

We now compare the three methods: (i) direct Gauss–Legendre quadrature, (ii) Helsing–Ojala complex interpolatory quadrature, and (iii) the proposed real singularity swap quadrature. We evaluate the Laplace double layer potential from a single panel,

$$u(\mathbf{x}) = \int_\Gamma \rho(\mathbf{y}) \frac{(\mathbf{y} - \mathbf{x}) \cdot \mathbf{n}}{|\mathbf{y} - \mathbf{x}|^2} ds(\mathbf{y}). \quad (42)$$

The panel is the same parabola as in Fig. 1, i.e. $\mathbf{g}(t) = (t, kt^2)$, or in complex form, $\gamma(t) = t + ikt^2$, and we explore the dependence on curvature k . In complex form, which is necessary to apply the quadratures, the layer potential is (see “Appendix A”)

$$u(\zeta) = -\operatorname{Im} \int_{-1}^1 \frac{\rho(\gamma(t))\gamma'(t) dt}{\gamma(t) - \zeta}. \quad (43)$$

Figures 2 and 3 compare the three schemes for various curvatures, for $n = 16$ and $n = 32$ respectively. For $n = 32$ all data is upsampled by Lagrange interpolation from $n = 16$ (which is already adequate to represent it to machine precision). The

error is in all cases measured against a reference solution computed using adaptive quadrature (`integral` in MATLAB with `abstol` and `reltol` set to `eps`), and the relative error E_{rel} computed against the maximum value of u on the grid. To see how they behave when pushed to their limits, we deliberately apply the quadratures much further away than necessary, i.e., also in the region where direct quadrature achieves full numerical accuracy.

The direct errors (left column of each figure) have magnitudes controlled by the images of Bernstein ellipses under the parameterization, as explained in Sect. 2.1. The Helsing–Ojala scheme (middle column) is much improved over direct quadrature, but, because of its convergence rates dependence (Sect. 2.3) the reduction in accuracy is severe as curvature increases, although it can to some extent be ameliorated by using $n = 32$. In addition, there is an error in the quadrature that grows with the distance from the panel, particularly for $n = 32$, due to amplification of roundoff error in the upward recurrence for the integrals $\{p_1, \dots, p_n\}$. This was also noted in [23], where the suggested fix was to instead run the recurrence backwards for distant points, starting from a value of p_n computed using the direct quadrature. This avoids the problem of catastrophic accuracy loss, but only because specialized quadrature is used where direct quadrature would be sufficient. The bottom middle plot of Fig. 3 shows at best 10 accurate digits, and that only over quite a narrow range of distances, making the use of the HO algorithm to give a uniform accuracy for all target locations rather brittle.

For the proposed singularity swap quadrature (right columns), there is still a loss of accuracy with increased curvature, but the improvement over Helsing–Ojala is striking. The instability of upwards recurrence is also much more mild.

Remark 10 The loss of accuracy with increased curvature can for the singularity swap quadrature be explained by considering the roots of the displacement function $Q(t)$. We have “swapped” out the nearest root t_0 , so the region of analyticity of the regularized integrand $h(t)(t - t_0)/Q(t)$ is bounded by the second nearest root of $Q(t)$, illustrated by z_2 in Fig. 4. This limits the convergence rate of the polynomial coefficients $\{c_k\}$, such that we may need to upsample the panel in order for the coefficients to fully decay. In the case of our parabolic panel, the second nearest root gets closer with increased curvature, which explains why upsampling to 32 points is necessary to achieve full accuracy in some locations. Note that for the upsampling to be beneficial, the coefficients for $k > n$ must be nonzero. This is only the case if the components of the integrand (ρ, γ, γ') are upsampled separately, before the integrand is evaluated at the new nodes.

3 Line integrals on curved panels in three dimensions

Since the logarithmic kernel is irrelevant in 3D, we care about the parametric form of (1),

$$I_m = I_m(\mathbf{x}) = \int_{-1}^1 \frac{\tilde{f}(t)}{|\mathbf{g}(t) - \mathbf{x}|^m} |\mathbf{g}'(t)| dt, \quad m = 1, 3, 5, \dots, \quad (44)$$

where $\Gamma = \mathbf{g}([-1, 1])$ is an open curve in \mathbb{R}^3 , and \tilde{f} incorporates the density and possibly other smooth denominator factors in the kernel K . Fixing the target \mathbf{x} , then introducing the 3D squared distance function R^2 [analogous to (10)] and the smooth function h ,

$$R(t)^2 := |\mathbf{g}(t) - \mathbf{x}|^2 = (g_1(t) - x_1)^2 + (g_2(t) - x_2)^2 + (g_3(t) - x_3)^2, \quad (45)$$

$$h(t) := \tilde{f}(t) |\mathbf{g}'(t)|, \quad (46)$$

we can write (44) as

$$I_m = \int_{-1}^1 \frac{h(t)}{(R(t)^2)^{m/2}} dt. \quad (47)$$

This integrand has singularities at the roots of $R(t)^2$, which, since the function is real for real t , come in complex conjugate pairs $\{t_0, \bar{t}_0\}$. How to find these roots is discussed shortly in Sect. 3.2; for now we assume that they are known. As in 2D, if Γ is not very curved and the target is nearby there is only one nearby pair of roots.

We construct a quadrature for (47) as in the 2D case (38), except that now there is a conjugate pair of near singularities to cancel. Thus we write (47) as

$$I_m = \int_{-1}^1 H(t) \cdot \frac{1}{((t - t_0)(t - \bar{t}_0))^{m/2}} dt,$$

$$\text{where } H(t) := \frac{h(t) ((t - t_0)(t - \bar{t}_0))^{m/2}}{(R(t)^2)^{m/2}}. \quad (48)$$

$H(t)$ can be expected to be analytic in a much larger neighborhood of $[-1, 1]$ than the integrand. As in Sect. 2.4, we now represent $H(t)$ in a real monomial basis,

$$H(t) = \sum_{k=1}^n c_k t^{k-1}, \quad (49)$$

getting the coefficients c_k by solving the Vandermonde system

$$A\mathbf{c} = \mathbf{h}$$

with the matrix entries $A_{ij} = t_i^{j-1}$, where, as before, $\{t_i\}_{i=1}^n$ are the panel's parameter Legendre nodes in $[-1, 1]$. We fill \mathbf{h} by evaluating its entries $\{H(t_i)\}_{i=1}^n$. By analogy

with (24), we set

$$P_k^m(t_0) = \int_{-1}^1 \frac{t^{k-1}}{\left((t-t_0)(\bar{t}-t_0)\right)^{m/2}} dt = \int_{-1}^1 \frac{t^{k-1}}{|t-t_0|^m} dt, \quad k = 1, \dots, n, \tag{50}$$

which one can evaluate to machine accuracy using recurrence relations as described in the next section. Combining (50) and (49) gives the final quadrature approximation to (44),

$$I_m \approx \sum_{k=1}^n c_k P_k^m(t_0). \tag{51}$$

The adjoint method of Sect. 2.2.2 can again be used to solve for a weight vector λ^m whose inner product with \mathbf{h} approximates I_m . As in Sect. 2.4, the error in (51) is essentially due to the convergence rate of the best polynomial representation (49) on $[-1, 1]$, which is likely to be rapid because of the absence of curvature effects (Sect. 2.3).

3.1 Recurrence relations for 3D kernels on a straight line

The above singularity swap has transformed the integral on a curved line Γ in 3D to (48), whose denominator corresponds to that of a *straight line* in 3D. Such upward recursions are available in [45, App. B], where they were used for Stokes quadratures near straight segments. We will here present them with some improvements which increase their stability for t_0 in certain regions of \mathbb{C} . We present the case of a single root pair $\{t_0, \bar{t}_0\} = t_r \pm it_i$, for orders $m = 1, 3, 5$, noting that, while we have derived formulae for double root pairs, we have found that they are never needed in practice. To simplify notation (matching notation for I_n^β in [45, App. B]), we write

$$b := -2t_r, \quad c := t_r^2 + t_i^2, \quad d := t_i^2, \tag{52}$$

and for the distances to the parameter endpoints we write

$$u_1 = \sqrt{(1+t_r)^2 + t_i^2} = |1+t_0|, \quad u_2 = \sqrt{(1-t_r)^2 + t_i^2} = |1-t_0|. \tag{53}$$

Beginning with $m = 1$ and $k = 1$, the integral (50) is

$$P_1^1(t_0) = \int_{-1}^1 \frac{dt}{\sqrt{(t-t_r)^2 + t_i^2}} = \log(1-t_r+u_2) - \log(-1-t_r+u_1). \tag{54}$$

This expression suffers from cancellation for t_0 close to $[-1, 1]$, and must be carefully evaluated. To begin with, it is more accurate in the left half plane, even though the

integral is symmetric in t_r , so the accuracy can be increased by evaluating it after the substitution $t_r \rightarrow -|t_r|$. In addition, the argument of the second logarithm of (54) suffers from cancellation when $|t_r| < 1$ and $t_i^2 \ll (1 - |t_r|)^2$, even after the substitution. In this case we evaluate it using a Taylor series in t_i ,

$$\begin{aligned} & -(1 - |t_r|) + \sqrt{(1 - |t_r|)^2 + t_i^2} \\ &= (1 - |t_r|) \sum_{n=1}^{\infty} \frac{(-1)^n (2n)!}{(1 - 2n)(n!)^2 (4^n)} \left(\frac{t_i}{1 - |t_r|}\right)^{2n} =: S_1(t_0). \end{aligned} \quad (55)$$

We achieve sufficient accuracy by applying this series evaluation to points inside the rhombus described by $4|t_i| < 1 - |t_r|$, evaluating the series to $n = 11$ terms. For $k > 1$ the upward recursions of [45] are stable, such that

$$P_1^1(t_0) = \log(1 + |t_r| + \sqrt{(1 + |t_r|)^2 + t_i^2}) \quad (56)$$

$$- \begin{cases} \log S_1(t_0), & \text{if } 4|t_i| < 1 - |t_r|, \\ \log(-1 + |t_r| + \sqrt{(-1 + |t_r|)^2 + t_i^2}), & \text{otherwise,} \end{cases} \quad (57)$$

$$P_2^1(t_0) = u_2 - u_1 - \frac{b}{2} P_1^1(t_0), \quad (58)$$

$$P_{k+1}^1(t_0) = \frac{1}{k} \left(u_2 - (-1)^{n-1} u_1 + \frac{(1 - 2k)b}{2} P_k^1(t_0) - (k - 1)c P_{k-1}^1(t_0) \right). \quad (59)$$

For $m = 3$, the formula from [45] for $k = 1$ contains a conditional statement for points on the real axis ($d = 0$), where the pair of two conjugate roots $\{t_0, \bar{t}_0\}$ merge into a double root. In finite precision, this property causes a region around the real axis where the formula is inaccurate, namely two cones extending outwards from the endpoints ± 1 . To get high accuracy there, we consider the integral with shifted limits,

$$P_1^3(t_0) = \int_{-1-t_r}^{1-t_r} (s^2 + t_i^2)^{-3/2} ds = [S_3(s)]_{-1-t_r}^{1-t_r}, \quad (60)$$

where the antiderivative S_3 is evaluated by forming a Maclaurin series of the integrand in t_i , and then integrating that series exactly in s , thus

$$S_3(s) = \int (s^2 + t_i^2)^{-3/2} ds = \frac{|s|}{s^3} \sum_{n=0}^{\infty} \left(-\frac{1}{4}\right)^{n+1} \frac{(2n+2)!}{((n+1)!)^2} \left(\frac{t_i}{s}\right)^{2n}. \quad (61)$$

We find that we get sufficiently high accuracy in the region of interest by truncating this series to 30 terms, denoted by $S_3(s)$, and finally evaluating the integrals for $m = 3$ as

$$P_1^3(t_0) = \begin{cases} \tilde{S}_3(1 - t_r) - \tilde{S}_3(-1 - t_r), & \text{if } 0 < \frac{|t_i|}{|t_r|-1} < 0.6, \\ \frac{1}{2d} \left(\frac{b+2}{u_2} - \frac{b-2}{u_1} \right), & \text{otherwise,} \end{cases} \tag{62}$$

$$P_2^3(t_0) = \frac{1}{u_1} - \frac{1}{u_2} - \frac{b}{2} P_1^3(t_0), \tag{63}$$

$$P_{k+1}^3(t_0) = P_{k-1}^1(t_0) - bP_k^3(t_0) - cP_{k-1}^3(t_0). \tag{64}$$

The series evaluation can be optimized by defining a succession of narrower cones, using fewer terms in each cone.

For $m = 5$ and $k = 1$ we need to repeat the process of finding a power series that we can use in cones around the real axis, extending from the endpoints. We now consider

$$P_1^5(t_0) = \int_{-1-t_r}^{1-t_r} (s^2 + t_i^2)^{-5/2} ds, \tag{65}$$

and the antiderivative of the Maclaurin series of the integrand is

$$S_5(s) = \int (s^2 + t_i^2)^{-5/2} ds = \frac{|s|}{s^5} \sum_{n=0}^{\infty} \frac{(-1)^{n+1} (4n^2 + 8n + 3)}{3(n + 2)2^{2n+1}} \frac{(2n)!}{(n!)^2} \left(\frac{t_i}{s} \right)^{2n}. \tag{66}$$

Truncating this series to a maximum of 50 terms, denoted by $\tilde{S}_5(s)$, we now evaluate the integrals for $m = 5$ as

$$P_1^5(t_0) = \begin{cases} \tilde{S}_5(1 - t_r) - \tilde{S}_5(-1 - t_r), & \text{if } 0 < \frac{|t_i|}{|t_r|-1} < 0.7, \\ \frac{1}{3d} \left(\frac{b+2}{2u_2^3} - \frac{b-2}{2u_1^3} + 2P_1^3(t_0) \right), & \text{otherwise,} \end{cases} \tag{67}$$

$$P_2^5(t_0) = \frac{1}{3u_1^3} - \frac{1}{3u_2^3} - \frac{b}{2} P_1^5(t_0), \tag{68}$$

$$P_{k+1}^5(t_0) = P_{k-1}^3(t_0) - bP_k^5(t_0) - cP_{k-1}^5(t_0). \tag{69}$$

Note that our expression for P_2^5 is different from the one found in [45]; ours avoids a conditional.

3.2 Finding the roots of $R(\mathbf{t})^2$

In three dimensions, the nearby roots of the squared distance function (45) do not correspond to a single point in space, as was the case in 2D (see Fig. 4). Instead, each complex conjugate pair of roots $\{t_0, \bar{t}_0\}$ near $[-1, 1]$ corresponds to a circle in space looped around Γ . To see this, let \mathbf{g}_r and \mathbf{g}_i be the real and imaginary parts of the

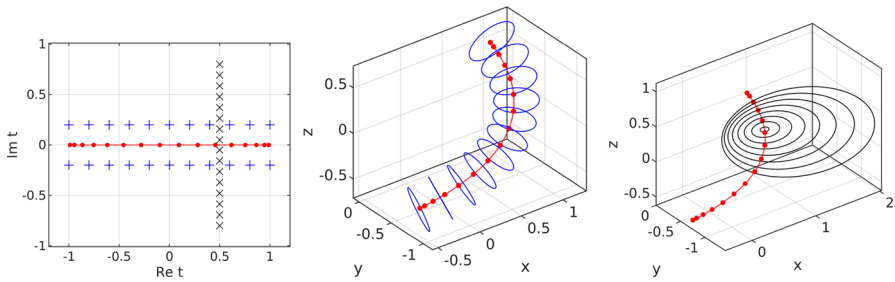


Fig. 5 Correspondence between complex roots of the 3D squared distance function $R(t)^2$ in (45) and target points in \mathbb{R}^3 , close to a curve. The red dots (left) are the Legendre nodes on $[-1, 1]$, mapped to a curve in space (center and right). The roots marked with a blue + in the left figure correspond to the blue circles in the middle figure, while the roots marked with a black x in the left figure correspond to the black circles in the right figure (color figure online)

complexification of the parametrization,

$$g(t) = g_r(t) + i g_i(t). \tag{70}$$

Then, the roots of (45) satisfy

$$R(t)^2 = |g_r(t) - x|^2 + 2i g_i(t) \cdot (g_r(t) - x) - |g_i(t)|^2 = 0. \tag{71}$$

For this to hold in both real and imaginary components for a given t , the point x must satisfy

$$\begin{cases} g_i(t) \cdot (g_r(t) - x) = 0, \\ |g_r(t) - x|^2 = |g_i(t)|^2. \end{cases} \tag{72}$$

The above equations describe the intersection of a plane with a sphere, and are satisfied by the circle of radius $|g_i(t)|$ that is centered at $g_r(t)$, and lies in the plane normal to $g_i(t)$. All points in \mathbb{R}^3 corresponding to the root t lie on this circle. For an illustration of this, see Fig. 5.

In order to construct a quadrature for a given target point x , we need to find the roots t of (45), which are the points satisfying the complex equation

$$R(t)^2 = (g_1(t) - x_1)^2 + (g_2(t) - x_2)^2 + (g_3(t) - x_3)^2 = 0. \tag{73}$$

We form a polynomial approximation $P_n[g](t)$ of $g(t)$ using the existing real Gauss–Legendre nodes $\{g(t_j)\}$. We then need to find the roots of the polynomial

$$\tilde{R}(t)^2 := (P_n[g_1](t) - x_1)^2 + (P_n[g_2](t) - x_2)^2 + (P_n[g_3](t) - x_3)^2. \tag{74}$$

These can be found using either of the root-finding methods discussed in Sect. 2.4.1. We here limit ourselves to finding the root pair $\{t_0, \bar{t}_0\}$ closest to $[-1, 1]$, using Newton’s

method. We find that a stable initial guess can be obtained using a linear mapping in the plane containing \mathbf{x} and the two closest discretization points on Γ , which we denote by $\mathbf{g}_j := \mathbf{g}(t_j)$ and $\mathbf{g}_k := \mathbf{g}(t_k)$. The initial guess t_{init} is then set to one of the two points satisfying

$$\frac{\operatorname{Re} t_{\text{init}} - t_j}{t_k - t_j} = \frac{(\mathbf{x} - \mathbf{g}_j) \cdot (\mathbf{g}_k - \mathbf{g}_j)}{\|\mathbf{g}_k - \mathbf{g}_j\|^2}, \quad \text{and} \quad \frac{|t_{\text{init}} - t_j|}{|t_k - t_j|} = \frac{\|\mathbf{x} - \mathbf{g}_j\|}{\|\mathbf{g}_k - \mathbf{g}_j\|}. \tag{75}$$

This is the exact root for the case of Γ a straight line.

Compared to the 2D case, we find that the accuracy of our method in 3D is more sensitive to the implementation details of the root-finding algorithm, especially for roots very close to the interval $[-1, 1]$. Below are a number of noteworthy observations:

- The convergence of Newton’s method deteriorates for roots that are close to the real axis (or on it, when $|\operatorname{Re} t_0| > 1$). This is because the root t_0 and its conjugate \bar{t}_0 then lie very close to each other, and in the case of $\operatorname{Im} t_0 = 0$ form a double root. To ameliorate this problem, we let our root-finding routine switch to Muller’s method [40] if Newton has not converged within a certain number of iterations (we use 20). In our tests, this appears to be a stable root-finding process for all points requiring special quadrature, such that we can apply direct quadrature to points where neither Newton’s nor Muller’s methods converge. For additional robustness, one could also switch to the comrade matrix method in such cases.
- Just as in the 2D case, once the root t_0 is known we can determine if special quadrature is needed using the Bernstein radius $\rho(t_0)$ criterion at the end of Sect. 2.4.1.
- We find that it is important for accuracy to find the roots using the approximation (74) of R^2 . If instead R^2 is approximated as $P_n[R^2](t)$, then 1–2 digits of accuracy are lost close to $[-1, 1]$. If roots are computed from this form, for example if the comrade matrix method is used, then they can be polished by taking one or two Newton steps of (74), thereby recovering the lost accuracy.
- For panels discretized with more than $n = 16$ points, we find that the best accuracy is achieved if the roots are found using Legendre expansions truncated to 16 terms. In general, our recommendation is to not use an underlying discretization with n larger than 16, and then upsample to more points when evaluating the SSQ, if necessary (see Remark 10).

4 Summary of algorithm

Below is a step by step summary of our algorithm for evaluating the potential generated by a single open curve panel Γ , in 2D or 3D, discretized using n Gauss–Legendre points. Specifically, we describe how to get the weight vector λ as defined in Sect. 2.2.2, from which one of the line integrals (17), (18) or (44) can be approximated by $I = \lambda^T \mathbf{f}$ for any sample vector \mathbf{f} of densities (or density times smooth geometric factors). Recall that potentials are defined by such line integrals, possibly in the 2D case via

complexification as shown in “Appendix A”. (We omit the optional finding of the panel’s Schwarz singularity, and use of the robust comrade-matrix method given in Sect. 2.4.1, since we find that it is not needed in practice.)

There is an overall algorithm option (flag) which may take the values “no upsampling”, “upsampling” or “upsampling with upsampled direct”. Either variety of upsampling is more expensive, but can increase accuracy; the second variety avoids some special weight computations so is the cheaper of the two.

Precomputations for panel Γ :

1. Choose a suitable polynomial basis (e.g. Legendre or Chebyshev; we use the former). The analytic continuation of the chosen basis functions must be simple to evaluate (as is the case for Legendre). Form the degree $n - 1$ polynomial approximation of the panel parameterization $P_n[\mathbf{g}](t)$ in this basis; this can be done by solving a $n \times n$ linear system with right-hand side the coordinates of the nodes $\mathbf{y}_j = \mathbf{g}(t_j)$, $j = 1, \dots, n$.
2. Given the desired tolerance ϵ , set the critical Bernstein radius ρ_ϵ via (41).

For each given target point \mathbf{x} :

1. Use a cheap criterion to check if \mathbf{x} is a candidate for near evaluation, based on the minimum distance between the target point and the panel nodes:

$$\mathbf{x} \text{ candidate if } \min_{\mathbf{y}_i \in \Gamma} |\mathbf{y}_i - \mathbf{x}| < D, \quad (76)$$

where D is a multiple of the panel length h (at $n = 16$ we use $D = h$ for full accuracy). If not a candidate, use the direct rule (6) then exit; more specifically the weights are

$$\lambda_j = w_j |\mathbf{g}'(t_j)| K(\mathbf{x}, \mathbf{y}_j), \quad j = 1, \dots, n. \quad (77)$$

2. Find the complex target preimage t_0 from the basis approximation to $\mathbf{g}(t)$: In 2D, use Newton’s method applied to (40) to find the root t_0 . In 3D, use Newton (complemented by Muller’s method) applied to (74) to find the root pair $\{t_0, \bar{t}_0\}$.
3. If $\rho(t_0) \geq \rho_\epsilon$, use the direct rule (77) and exit. If no upsampling, set $\tilde{n} = n$, or if upsampling, $\tilde{n} = 2n$. If “upsampling with upsampled direct”, check if $\sqrt{\rho_\epsilon} \leq \rho(z) < \rho_\epsilon$, and if so use the degree $n - 1$ approximants to resample the kernel and density to the m new nodes and use the upsampled m -node version of the direct formula (77), then exit.
4. We now do special quadrature for evaluation at \mathbf{x} . More precisely, we use the singularity swap method with parameter target t_0 and panel $[-1, 1]$, as follows.

For 2D, compute the monomial integral vector \mathbf{p} , being $\{p_k^m(t_0)\}_{k=0}^{\tilde{n}}$, or for the logarithmic case $\{q_k(t_0)\}_{k=0}^{\tilde{n}}$, via the recurrences (25)–(29) applied to $[-1, 1]$. For 3D, it is $\{P_k^m(t_0)\}_{k=0}^{\tilde{n}}$, computed via the various recurrences of Sect. 3.1.

5. Solve the adjoint Vandermonde system (32) where the \tilde{n} -by- \tilde{n} matrix A involves the nodes on $[-1, 1]$, and the right-hand side is the \mathbf{p} just computed. We find that

the Björck–Pereyra algorithm [10] is even faster than using a precomputed LU decomposition of A .

6. Finally, one must apply corrections to turn λ into a set of weights that act on the samples of f , rather than the singularity-swapped smooth functions involving $h(t)$ or $H(t)$. First multiply each weight λ_j by the Jacobian factor $|\mathbf{g}'(t_j)|$, according to (36) or (46). Then:
 - For the log case in 2D, according to (37), add $w_j \log(Q(t_j)/(t_j - t_0))$ to each weight λ_j .
 - For the power-law case in 2D, according to (38), multiply each weight by $((t_j - t_0)/Q(t_j))^m$.
 - In 3D, according to (48), multiply each weight by $((t_j - t_0)\sqrt{(t_j - t_0)}/R(t_j)^2)^{m/2}$.

5 Numerical tests

In Sect. 2 we already performed basic comparisons of prior methods and the proposed singularity swap quadrature in the 2D case. Now we give results of more extensive 2D tests and the 3D case.

The 2D and 3D quadrature methods of this paper have been implemented in a set of MATLAB routines. These are available online [1], together with the programs used for all the numerical experiments reported here. The computationally intensive steps of the 3D quadrature have been implemented in C and interfaced to MATLAB using MEX. The code is called using a list of target points, and has rudimentary parallelization in the form of OpenMP worksharing, so that each thread works on a subset of the target points. This allows very rapid computation of quadrature weights: on the computer used for the numerical results of this section, which is a 3.6 GHz quad-core Intel i7-7700, we can find the roots (preimages) of 4.7×10^6 target points per second, and compute 3.2×10^6 sets of target-specific quadrature weights per second, for $n = 16$ (without upsampling).

5.1 Evaluating a layer potential in two dimensions

To study quadrature performance when evaluating an actual layer potential, we solve a simple test problem using an integral equation method, then evaluate the solution using either the Helsing–Ojala method or our proposed method. Following [23] we test the Laplace equation $\Delta u = 0$ in the interior of the starfish-shaped domain Ω given by $\gamma(t) = (1 + 0.3 \cos(5t))e^{it}$, $t \in [0, 2\pi)$, and we choose Dirichlet boundary conditions $u = u_e$ on $\partial\Omega$ with data $u_e(\zeta) = \log|3 + 3i - \zeta|$. We have picked this data to be very smooth, since the errors then are due to quadrature, rather than density resolution, making it easier to compare methods.

We discretize $\partial\Omega$ using composite Gauss–Legendre quadrature, with $n = 16$ points per panel. The panels are formed using the following adaptive scheme. The interval $[0, 2\pi)$ is recursively bisected into segments that map to panels, until all panels satisfy a resolution condition, and all pairs of neighboring segments differ in length by a ratio this is no more than 2 (i.e. level-restriction). The resolution condition is based on the

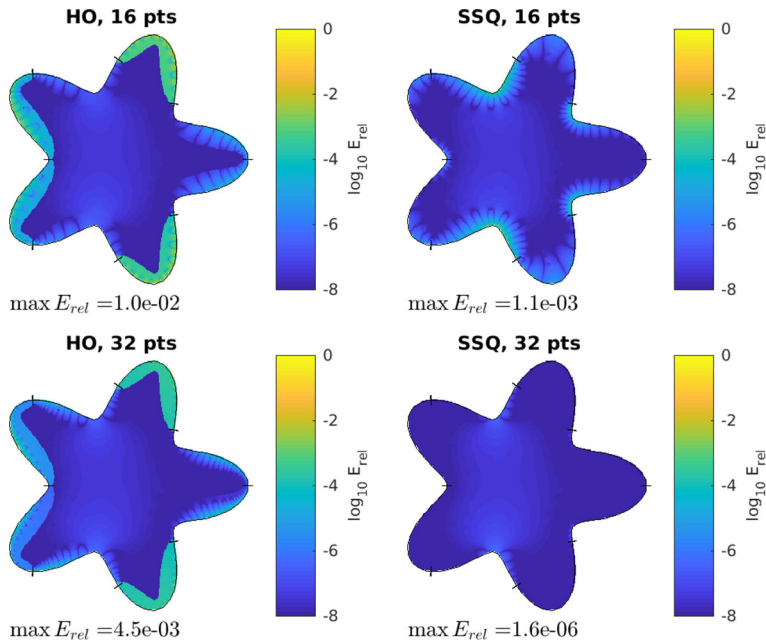


Fig. 6 Comparison of 2D quadratures when evaluating Laplace double layer potential using a coarse discretization (adaptive with $\epsilon = 10^{-6}$, giving 8 panels). Left column: Helsing–Ojala (HO). Right column: proposed singularity swap quadrature (SSQ). Top row: no upsampling. Bottom row: upsampling

Legendre expansion coefficients $\{\hat{\gamma}'_k\}$ of the panel derivative $\gamma'(t)$: a panel is deemed resolved to a tolerance ϵ if $\max(|\hat{\gamma}'_{n-1}|, |\hat{\gamma}'_n|) < \epsilon \|\hat{\gamma}'\|_\infty$. While somewhat ad-hoc, this produces a solution with a relative error that is comparable to ϵ , for a smooth problem (see Figs. 6 and 7).

The solution to the Laplace equation is represented using the double layer potential (43), which results in a second kind integral equation in ρ . This is solved using the Nyström method and the above discretization, which is straightforward due to the layer potential being smooth on $\partial\Omega$. For further details on this solution procedure, see, for example, [20] or [23]. Once we have ρ , we evaluate the layer potential using quadrature at sets of points in Ω , and compute the error by comparing with the exact solution, u_e . We compare two methods:

1. The Helsing–Ojala (HO) method summarized in Sect. 2.2.
2. The proposed singularity swap quadrature (SSQ), as outlined in Sect. 4.

In both methods we test “no upsampling” with $n = 16$, and “upsampling with upsampled direct” with $\tilde{n} = 32$.

In our first test, shown in Fig. 6, we solve the problem using a quite coarse discretization (adaptive with $\epsilon = 10^{-6}$), choose $\rho_\epsilon = 1.8$ (corresponding via (41) to a more conservative error of 10^{-8}), and evaluate the solution on a 300×300 uniform grid covering the domain. Far from the boundary the solution shows around the expected 6 digit accuracy. Notice that, since only eight panels are used, they are highly curved.

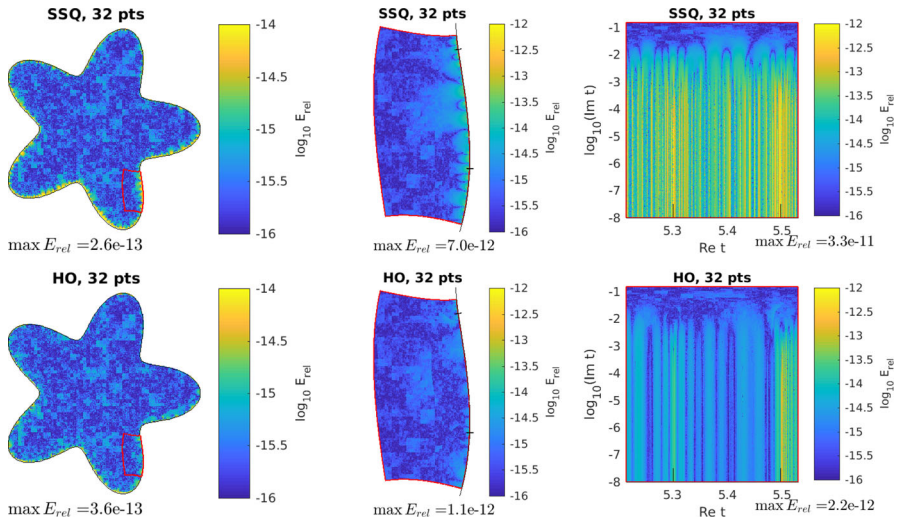


Fig. 7 Comparison of 2D quadratures when evaluating Laplace double layer potential using a high-accuracy discretization (adaptive with $\epsilon = 10^{-14}$, giving 32 panels) and upsampling. Top row: proposed singularity swap quadrature (SSQ). Bottom row: Helsing–Ojala (HO). Near region (center) is marked red in the left plots. Right plot shows error versus root location in near region, with exponential grading of the imaginary part, proportional to distance. Note the different color scale used in the near region

HO gets only 2 digits at $\tilde{n} = 16$ and scarcely more at $\tilde{n} = 32$, whereas SSQ gets 3 digits at $\tilde{n} = 16$ and the “full” 6 digits at $\tilde{n} = 32$. This shows that upsampled SSQ achieves the full expected accuracy using the efficient discretization chosen for this accuracy, whereas HO would require further panel subdivision beyond that needed to achieve the requested Nyström accuracy.

As a second test, we solve the same problem, this time using a fine discretization (adaptive with $\epsilon = 10^{-14}$), and pick $\rho_\epsilon = 3$, corresponding via (41) to around $\epsilon \approx 10^{-15}$ at $n = 16$. The result is 32 panels (similar in number to the 35 panels used in [23]). First we evaluate the solution on a uniform 300×300 grid covering all of Ω . Then we evaluate the solution on the slice of Ω defined by the mapping $\gamma(t)$ of the square in \mathbb{C} defined by $\text{Re } t \in [1.66\pi, 1.76\pi]$ and $\text{Im } t \in [d_{\min}, 0.15]$. Here we use two grids in t : one uniform 300×300 with $d_{\min} = 10^{-3}$, and one 300×300 that is uniform in $\text{Re } t$ and logarithmic in $\text{Im } t$, with $d_{\min} = 10^{-8}$. Here we only show the results when using upsampling, since both methods require that to achieve maximum accuracy (without upsampling, HO gets 8 digits and SSQ 9 digits). The results of this test are shown in Fig. 7. Both schemes are able to achieve high accuracy here: both get 13 digits close to the boundary, and both drop to 11–12 digits when tested extremely close to the boundary. However, contrary to the previous test, the proposed SSQ does lose around 1 digit relative to HO, in the distance range 10^{-3} to 10^{-8} . We do not have an explanation for this loss.

To summarize this experiment, SSQ is much better than HO for panels that are resolved to lower accuracies, because these panels may be quite curved. When panels

are resolved to close to machine precision the methods are about the same, with HO attaining a slightly smaller error.

5.2 Evaluating the field due to a slender body in three dimensions

To test our method in 3D, we evaluate the Stokes flow around a thin filament in the slender body approximation. In this approximation, the flow due to a fiber with centerline Γ and radius $\varepsilon \ll 1$ is given by the line integral (see e.g. [33])

$$\mathbf{u}(\mathbf{x}) = \int_{\Gamma} \left(\mathcal{S}(\mathbf{x} - \mathbf{y}) + \frac{\varepsilon^2}{2} \mathcal{D}(\mathbf{x} - \mathbf{y}) \right) \mathbf{f}(\mathbf{y}) \, ds(\mathbf{y}). \quad (78)$$

Here \mathbf{f} is given force data, defined on the centerline, \mathcal{S} is the Stokeslet kernel, defined as

$$\mathcal{S}(\mathbf{R}) = \frac{\mathbf{I}}{|\mathbf{R}|} + \frac{\mathbf{R}\mathbf{R}^T}{|\mathbf{R}|^3}, \quad (79)$$

and \mathcal{D} is the doublet kernel, defined as

$$\mathcal{D}(\mathbf{R}) = \frac{1}{2} \Delta \mathcal{S}(\mathbf{R}) = \frac{\mathbf{I}}{|\mathbf{R}|^3} - 3 \frac{\mathbf{R}\mathbf{R}^T}{|\mathbf{R}|^5}. \quad (80)$$

Note that \mathcal{S} and \mathcal{D} are tensors, and \mathbf{I} is the 3×3 identity. To apply our quadrature, we first write (78) in kernel-split form,

$$\mathbf{u}(\mathbf{x}) = I_1 + I_3 + I_5, \quad (81)$$

where

$$I_1 = \int_{\Gamma} \frac{\mathbf{f}(\mathbf{y})}{|\mathbf{R}|} \, ds(\mathbf{y}), \quad (82)$$

$$I_3 = \int_{\Gamma} \frac{\left(\mathbf{R}\mathbf{R}^T + \varepsilon^2 \mathbf{I}/2 \right) \mathbf{f}(\mathbf{y})}{|\mathbf{R}|^3} \, ds(\mathbf{y}), \quad (83)$$

$$I_5 = -\frac{3\varepsilon^2}{2} \int_{\Gamma} \frac{\mathbf{R}\mathbf{R}^T \mathbf{f}(\mathbf{y})}{|\mathbf{R}|^5} \, ds(\mathbf{y}), \quad (84)$$

and $\mathbf{R} := \mathbf{x} - \mathbf{y}$. Once in this form, one can compute modified weights for I_1 , I_3 , and I_5 using the proposed SSQ quadrature.

For our tests, we use the force data $\mathbf{f}(\mathbf{y}) = \mathbf{y}$ and radius $\varepsilon = 10^{-3}$. We let Γ be the closed curve shown in Fig. 8, which is described by a Fourier series with decaying random coefficients,

$$\boldsymbol{y}(t) = \text{Re} \sum_{k=-20}^{20} \frac{\boldsymbol{c}(k)}{5 + |k|} \hat{\boldsymbol{y}}(k) e^{2\pi i k t}, \quad t \in [0, 1), \tag{85}$$

where the components of $\boldsymbol{c}(k)$ are complex and normally distributed random numbers.⁴ We subdivide Γ into panels by recursively bisecting the curve until each panel is resolved to a tolerance ϵ , using a criterion similar to that used in Sect. 5.1: a panel is deemed resolved if the expansion coefficients $\{\hat{s}_k\}$ of the speed function $s(t) = |\boldsymbol{y}'(t)|$ satisfy $\max(|\hat{s}_{n-1}|, |\hat{s}_n|) < \epsilon \|\hat{s}\|_\infty$. Each panel is then discretized using $n = 16$ Gauss–Legendre nodes.

5.2.1 Comparison to adaptive quadrature

As a comparison for our method, and also to compute reference values, we have implemented a scheme for nearly singular 3D line quadrature based on per-target adaptive refinement: For a given target point \boldsymbol{x} , nearby panels are recursively subdivided until each panel Γ_i satisfies $\min_{\boldsymbol{y} \in \Gamma_i} |\boldsymbol{x} - \boldsymbol{y}| < h_i$, where h_i is the arc length of Γ_i . Empirically, this is sufficient for accurate evaluation of the slender body kernel on a 16-point panel. Each newly formed panel is discretized using $n = 16$ Gauss–Legendre points, and the required data $(\boldsymbol{y}, |\boldsymbol{y}'|, \boldsymbol{f})$ is interpolated to these points from the parent panel using barycentric Lagrange interpolation [9]. After this new set of panels is formed, the field (78) can be accurately evaluated at \boldsymbol{x} using the Gauss–Legendre quadrature weights, since all source panels then are far away from \boldsymbol{x} , relative to their own length. This scheme is relatively simple to implement, but has the drawback of requiring an increasingly large amount of interpolations and kernel evaluations as \boldsymbol{x} approaches Γ .

In an attempt to obtain a fair comparison of runtimes between the adaptive quadrature and the proposed SSQ, we will only report on the time spent in the essential core tasks for each method, and ensure that they have been implemented with similar high efficiency. For the adaptive quadrature, we will therefore report the time spent on interpolations (reported as t_{interp}), since those are implemented using a combination of C and BLAS, and omit the time spent on the recursive subdivisions, since that is presently implemented in a relatively slow prototype code. For the SSQ, we will report on the total time spent on finding roots and computing the new quadrature weights (reported t_{weights}), since all the expensive steps of that algorithm have been implemented in C. For both methods we will also report on the time spent on near evaluations of the Stokeslet and doublet kernels (reported as t_{eval}), since those are computed using the same code. Far field evaluation times are omitted, since they are identical. Thus $t_{\text{eval}} + t_{\text{interp}}$ is somewhat less than the total time for an adaptive scheme, whereas $t_{\text{eval}} + t_{\text{weights}}$ is the total time for our proposed SSQ.

5.2.2 Results

As a first test, we evaluate the $\boldsymbol{u}(\boldsymbol{x})$ velocity field (78) on 200×200 points covering the xz -slice $[-1.4, 1.4] \times 0.25 \times [-1.4, 1.4]$. The curve is adaptively discretized using

⁴ In MATLAB: `rng(0); c = (randn(3,41)+1i*randn(3,41)).`

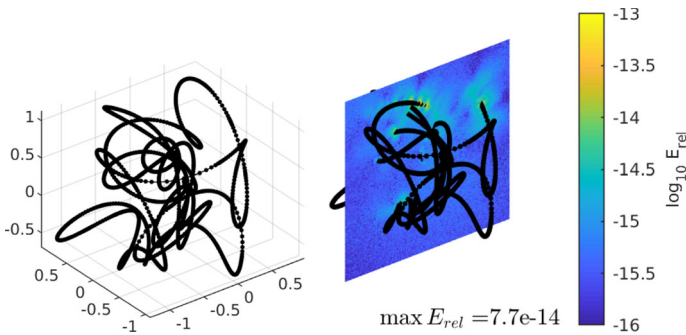


Fig. 8 Left: The curve in 3D described by Eq. (85), showing the discretization nodes (for $\epsilon = 10^{-10}$) as dots. Right: Relative error when evaluating the slender body field (78) on the shown slice using the proposed SSQ method

$\epsilon = 10^{-10}$, which results in 187 panels. For each point-panel interaction, we evaluate the SSQ using data upsampled to 32 Gauss–Legendre points, for all target points within the Bernstein radius $\rho_\epsilon = 3$ [corresponding via (41) to full accuracy]. Unlike in the 2D case, we find that this upsampling is necessary for achieving accurate results. The error is measured against a reference computed using the adaptive quadrature and a discretization with 18-point panels, created using $\epsilon = 5 \times 10^{-14}$ (this is to ensure that the grids are different, and that the reference grid is better resolved). The resulting error field, shown in Fig. 8, indicates that the SSQ can recover at least 13 digits of accuracy on all of this grid of target points, and that the largest errors are at the points closest to the curve.

To further study the behavior of SSQ close to the curve, and to compare it to the adaptive quadrature, we use both methods to compute \mathbf{u} at a set of random target points all located at the same distance d away from Γ . The test setup and results are listed in Table 1. As expected, the computational costs (kernel evaluations and runtime) of the proposed SSQ quadrature does not vary with the distance between the target points and the curve, as long as the target points are within the near evaluation threshold. The adaptive quadrature, on the other hand, has costs that grow slowly as the distance decreases, since more and more source points are required to evaluate the integral. The SSQ also appears to be a factor of several times cheaper, both in number of evaluations (4–7 times less) and in runtime (2.5–5 times faster), even though our timings are underestimates for adaptive quadrature. However, the error for the SSQ in this application starts to grow for close distances: for instance, at $d = 10^{-4}$ the table shows that it loses around 2 digits relative to the ϵ at which the panels were discretized.

Remark 11 This loss of accuracy in this application for targets at small distance d is not entirely surprising. The SSQ method gives weights λ at the given nodes for integrating the kernels $|\mathbf{R}|^{-1}$, $|\mathbf{R}|^{-3}$, and $|\mathbf{R}|^{-5}$. The resulting integrals diverge, respectively as $\log d$, d^{-2} , and d^{-4} . We have checked that, for these kernels with generic densities, the relative errors of SSQ are close to machine precision (around 13 digits), even though as $d \rightarrow 0$ the λ_j are growing and oscillatory. However, the Stokes kernels (82)–(84) involve near-vanishing numerator factors (such as $\mathbf{R}\mathbf{R}^T$) which partially cancel

Table 1 Adaptive versus proposed SSQ quadratures for 3D Stokes slender body

Parameters	Adaptive quad.				Singularity swap quad. (SSQ)			
	<i>epsilon</i>	N_{eval}	t_{eval}	t_{interp}	N_{eval}	t_{eval}	$t_{weights}$	$\max E_{rel}$
d								
1.0e-02	1.0e-10	2.8e+06	0.17 s	0.45 s	6.3e+05	0.06 s	0.19 s	1.7e-13
1.0e-02	1.0e-06	4.7e+06	0.30 s	0.78 s	1.3e+06	0.12 s	0.14 s	4.8e-09
1.0e-04	1.0e-10	4.4e+06	0.26 s	0.55 s	6.3e+05	0.06 s	0.18 s	2.0e-08
1.0e-04	1.0e-06	6.3e+06	0.38 s	0.86 s	1.3e+06	0.12 s	0.13 s	7.7e-05

There are 5000 random target points located at distance d from the curve Γ , which is discretized using tolerance ϵ and 16-point panels. SSQ uses upsampling to 32-point panels, while the adaptive quadrature uses subdivision into new 16-point panels. For either method, the number of near-field kernel evaluations is N_{eval} and the time spent on them t_{eval} . t_{interp} is the time interpolating to temporary grids in the adaptive quadrature. $t_{weights}$ is the total time finding the interpolatory quadrature weights in the proposed SSQ (i.e., rootfinding, recursions and Vandermonde solve). Error is measured against a reference solution computed with adaptive quadrature, $\epsilon = 5 \times 10^{-14}$, and $n = 18$

the singularities, leading to much smaller values for the integrals. In our method such factors are incorporated into a modified density \tilde{f} in (44), thus catastrophic cancellation as $d \rightarrow 0$ is inevitable for this application (this is also true for the straight fiber method of [45]). This could only be avoided by a more specialized set of recursions for the full Stokes kernels (83) and (84).

6 Conclusions

We have presented an improved version of the monomial approximation method of Helsing and Ojala [19,23,36] for nearby evaluation of 2D layer potentials discretized with panels. At the cost of a single Newton search for the target preimage, our method uses singularity cancellation to “swap” the singular problem on the general curved (complex) panel for one on the standard (real) panel $[-1, 1]$, much improving the error. Hence we call the method “singularity swap quadrature” (SSQ). We emphasize that it is not the *conditioning* of the monomial interpolation problem that improves—the Vandermonde matrix is exponentially ill-conditioned in either case—rather, the exponential *convergence rate* improves markedly in the case of curved panels. In the case of a Nyström discretization adapted to a requested tolerance, this allows SSQ to achieve this tolerance when HO would demand further subdivision. We gave a quantitative explanation for both the HO and SSQ convergence rates using classical polynomial approximation theory: the HO rate is limited by electrostatic “shielding” of a nearby Schwarz singularity by the panel itself.

We then showed that the singularity swap idea gives a close-evaluation quadrature for line integrals on general curves in 3D, which has so far been missing from the literature. The (to these authors) delightful fact that complex analysis can help in a 3D application relies on analytically continuing the squared-distance function (45). Note that QBX [7,29] *cannot* apply to line integrals in 3D, since the potential becomes singular (in the transverse directions) when approaching the curve. We demonstrated 3D SSQ in a slender body theory Stokes application. We found that, at least when targets are only moderately close, comparable accuracies to a standard adaptive quadrature are achieved with several times fewer kernel evaluations, and, in a simple C implementation, speeds several times faster.

As Remark 11 discusses, in the Stokes application, in order to retain full relative accuracy at arbitrarily small target distances a new set of recurrences for these particular kernels would be essential; we leave this for future work. Of course, the slender body approximation breaks down for distances comparable to or smaller than the body radius, so that the need for full accuracy at such small distances is debatable.

Acknowledgements Open access funding provided by Royal Institute of Technology. The authors would like to thank Johan Helsing, Charlie Epstein, Ondrej Maxian, and Aleks Donev for fruitful discussions, and the anonymous referees for useful comments. L.a.K. would like to thank the Knut and Alice Wallenberg Foundation for their support under Grant No. 2016.0410, and the Flatiron Institute for hosting him during part of this work. The Flatiron Institute is a division of the Simons Foundation.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence,

and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

Appendix A 2D kernels in complex variable form

In order to use either the Helsing–Ojala quadrature or the two-dimensional singularity swap quadrature proposed in this paper, the complex variable form of the kernel is required. In [41, App. A]) the single and double layer kernels for several common 2D elliptic PDEs are listed. If analytically separated into a kernel-split form [20, Sec. 6], these kernels can be summarized as having singularities of the following types. (Note that this is a subset of the forms derived in [21, Sec. 6], which include denominators up to $|\mathbf{y} - \mathbf{x}|^6$.)

$$\int \rho \frac{(\mathbf{y} - \mathbf{x}) \cdot \mathbf{f}}{|\mathbf{y} - \mathbf{x}|^2} ds(\mathbf{y}), \quad \int \rho \log|\mathbf{y} - \mathbf{x}| ds(\mathbf{y}), \tag{86}$$

$$\int \rho \frac{(\mathbf{y} - \mathbf{x})}{|\mathbf{y} - \mathbf{x}|^2} ds(\mathbf{y}), \quad \int \frac{\mathbf{f} \cdot (\mathbf{y} - \mathbf{x})(\mathbf{y} - \mathbf{x})(\mathbf{y} - \mathbf{x}) \cdot \hat{\mathbf{n}}}{|\mathbf{y} - \mathbf{x}|^4} ds(\mathbf{y}). \tag{87}$$

Let us use the following identifications between vectors in \mathbb{R}^2 and points in \mathbb{C} : $\mathbf{x} = z$, $\mathbf{y} = \tau$, $\hat{\mathbf{n}} = \nu$, $\gamma(t) = \mathbf{g}(t)$, and $\mathbf{f} = \mathbf{f}$. Our rewrites use the following definitions and basic results:

$$\mathbf{x} \cdot \mathbf{y} = \text{Re} [z\bar{\tau}], \tag{88}$$

$$\nu = i\gamma'/|\gamma'|, \tag{89}$$

$$ds(\mathbf{y}) = i\bar{\nu} d\tau = -i\nu d\bar{\tau}. \tag{90}$$

Throughout, ρ and \mathbf{f} are assumed to be arbitrary functions that are smooth and real-valued. The above integrals can then in complex form be written as

$$\int \rho \frac{(\mathbf{y} - \mathbf{x}) \cdot \mathbf{f}}{|\mathbf{y} - \mathbf{x}|^2} ds(\mathbf{y}) = -\text{Im} \int \frac{\rho \bar{\nu} d\tau}{\tau - z}, \tag{91}$$

$$\int \rho \frac{(\mathbf{y} - \mathbf{x})}{|\mathbf{y} - \mathbf{x}|^2} ds(\mathbf{y}) = \int \frac{\rho \bar{\nu}}{(\tau - z)} i d\tau, \tag{92}$$

$$\int \rho \log|\mathbf{y} - \mathbf{x}| ds(\mathbf{y}) = -\text{Im} \int \rho \bar{\nu} \log(\tau - z) d\tau, \tag{93}$$

$$\int \frac{\mathbf{f} \cdot (\mathbf{y} - \mathbf{x})(\mathbf{y} - \mathbf{x})(\mathbf{y} - \mathbf{x}) \cdot \hat{\mathbf{n}}}{|\mathbf{y} - \mathbf{x}|^4} ds(\mathbf{y}) = \frac{1}{4i} \left[\int \frac{(\bar{\tau} - \bar{z})\bar{f} d\tau}{(\tau - z)^2} + \int \frac{(\bar{f} + f\bar{v}^2) d\tau}{\tau - z} - \int \frac{f d\tau}{\tau - z} \right]. \quad (94)$$

As an example of how these can be used, if we let $\mathbf{f} = \hat{\mathbf{n}}$ in (91) and use that $v\bar{v} = 1$, we directly get the complex form of the double layer kernel of the Laplace equation,

$$\int \rho \frac{(\mathbf{y} - \mathbf{x}) \cdot \mathbf{n}}{|\mathbf{y} - \mathbf{x}|^2} ds(\mathbf{y}) = -\operatorname{Im} \int \frac{\rho d\tau}{\tau - z}. \quad (95)$$

References

1. af Klinteberg, L.: Line quadrature library (linequad) (2019). URL <http://github.com/ludvigak/linequad>
2. 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>
3. 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>
4. Atkinson, K.: *The Numerical Solution of Integral Equations of the Second Kind*. Cambridge University Press, Cambridge (1997)
5. Aurentz, J.L., Vandebril, R., Watkins, D.S.: Fast computation of eigenvalues of companion, comrade, and related matrices. *BIT Numer. Math.* **54**(1), 7–30 (2014). <https://doi.org/10.1007/s10543-013-0449-x>
6. Barnett, A., Wu, B., Veerapaneni, S.: Spectrally accurate quadratures for evaluation of layer potentials close to the boundary for the 2D Stokes and Laplace equations. *SIAM J. Sci. Comput.* **37**(4), B519–B542 (2015). <https://doi.org/10.1137/140990826>
7. Barnett, A.H.: Evaluation of layer potentials close to the boundary for Laplace and Helmholtz problems on analytic planar domains. *SIAM J. Sci. Comput.* **36**(2), A427–A451 (2014). <https://doi.org/10.1137/120900253>
8. Barnett, S.: A companion matrix analogue for orthogonal polynomials. *Linear Algebra Appl.* **12**(3), 197–202 (1975). [https://doi.org/10.1016/0024-3795\(75\)90041-5](https://doi.org/10.1016/0024-3795(75)90041-5)
9. Berrut, J.-P., Trefethen, L.N.: Barycentric Lagrange interpolation. *SIAM Rev.* **46**(3), 501–517 (2004). <https://doi.org/10.1137/S0036144502417715>
10. Björck, A., Pereyra, V.: Solution of Vandermonde systems of equations. *Math. Comput.* **24**(112), 893 (1970). <https://doi.org/10.2307/2004623>
11. Bruno, O.P., Haslam, M.C.: Regularity theory and superalgebraic solvers for wire antenna problems. *SIAM J. Sci. Comput.* **29**(4), 1375–1402 (2007)
12. Carvalho, C., Khatri, S., Kim, A.D.: Asymptotic analysis for close evaluation of layer potentials. *J. Comput. Phys.* **355**, 327–341 (2018)
13. Cortez, R.: Regularized Stokeslet segments. *J. Comput. Phys.* **375**, 783–796 (2018). <https://doi.org/10.1016/j.jcp.2018.08.055>
14. Davis, P.J.: *The Schwarz Function and Its Applications*. The Carus Mathematical Monographs, No. 17. The Mathematical Association of America, Buffalo (1974)
15. Donaldson, J.D., Elliott, D.: A unified approach to quadrature rules with asymptotic estimates of their remainders. *SIAM J. Numer. Anal.* **9**(4), 573–602 (1972). <https://doi.org/10.1137/0709051>
16. Gautschi, W., Inglese, G.: Lower bounds for condition number of Vandermonde matrices. *Numer. Math.* **52**, 241–250 (1988)
17. Götz, T.: *Interactions of fibers and flow: asymptotics, theory and numerics*. Ph.D. Thesis, University of Kaiserslautern, Germany (2000)

18. Hao, S., Barnett, A.H., Martinsson, P.G., Young, P.: High-order accurate methods for Nyström discretization of integral equations on smooth curves in the plane. *Adv. Comput. Math.* **40**(1), 245–272 (2014). <https://doi.org/10.1007/s10444-013-9306-3>
19. 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>
20. 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>
21. 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)
22. Helsing, J., Karlsson, A.: An explicit kernel-split panel-based Nyström scheme for integral equations on axially symmetric surfaces. *J. Comput. Phys.* **272**, 686–703 (2014). <https://doi.org/10.1016/j.jcp.2014.04.053>
23. Helsing, J., Ojala, R.: On the evaluation of layer potentials close to their sources. *J. Comput. Phys.* **227**(5), 2899–2921 (2008a). <https://doi.org/10.1016/j.jcp.2007.11.024>
24. Helsing, J., Ojala, R.: Corner singularities for elliptic problems: integral equations, graded meshes, quadrature, and compressed inverse preconditioning. *J. Comput. Phys.* **227**(20), 8820–8840 (2008b). <https://doi.org/10.1016/j.jcp.2008.06.022>
25. Ho, N., Leiderman, K., Olson, S.: A three-dimensional model of flagellar swimming in a Brinkman fluid. *J. Fluid Mech.* **864**, 1088–1124 (2019). <https://doi.org/10.1017/jfm.2019.36>
26. Johnson, R.E.: An improved slender-body theory for Stokes flow. *J. Fluid Mech.* **99**(2), 411–431 (1980)
27. Johnson, S.G.: Notes on adjoint methods (2012). <https://math.mit.edu/~stevenj/18.336/adjoint.pdf>
28. Keller, J.B., Rubinow, S.I.: Slender-body theory for slow viscous flow. *J. Fluid Mech.* **75**(4), 705–714 (1976)
29. Klöckner, A., Barnett, A., Greengard, L., O’Neil, M.: Quadrature by expansion: a new method for the evaluation of layer potentials. *J. Comput. Phys.* **252**, 332–349 (2013). <https://doi.org/10.1016/j.jcp.2013.06.027>
30. Koens, L., Lauga, E.: The boundary integral formulation of Stokes flows includes slender-body theory. *J. Fluid Mech.* **850**, R1 (2018). <https://doi.org/10.1017/jfm.2018.483>
31. Kress, R.: *Linear Integral Equations*, 3rd edn. Springer, New York (2014)
32. Millar, R.F.: Singularities and the Rayleigh hypothesis for solutions to the Helmholtz equation. *IMA J. Appl. Math.* **37**(2), 155–171 (1986)
33. Mori, Y., Ohm, L., Spirn, D.: Theoretical justification and error analysis for slender body theory. *Commun. Pure Appl. Math.* (2020a). <https://doi.org/10.1002/cpa.21872>. [arXiv:1807.00178](https://arxiv.org/abs/1807.00178)
34. Mori, Y., Ohm, L., Spirn, D.: Theoretical justification and error analysis for slender body theory with free ends. *Arch. Ration. Mech. Anal.* **235**, 1905–1978 (2020b)
35. Nazockdast, E., Rahimian, A., Zorin, D., Shelley, M.J.: A fast platform for simulating semi-flexible fiber suspensions applied to cell mechanics. *J. Comput. Phys.* **329**, 173–209 (2017)
36. Ojala, R.: A robust and accurate solver of Laplace’s equation with general boundary conditions on general domains in the plane. *J. Comput. Math.* **30**(4), 433–448 (2012)
37. 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>
38. Pan, V.Y.: How bad are Vandermonde matrices? *SIAM J. Matrix Anal. Appl.* **37**(2), 676–694 (2016)
39. Pérez-Arancibia, C., Faria, L.M., Turc, C.: Harmonic density interpolation methods for high-order evaluation of Laplace layer potentials in 2D and 3D. *J. Comput. Phys.* **376**, 411–434 (2019). <https://doi.org/10.1016/j.jcp.2018.10.002>
40. Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P.: *Numerical Recipes: The Art of Scientific Computing*, 3rd edn. Cambridge University Press, New York (2007). ISBN 9780521880688
41. Rahimian, A., Barnett, A., Zorin, D.: Ubiquitous evaluation of layer potentials using quadrature by kernel-independent expansion. *BIT Numer. Math.* **58**(2), 423–456 (2018). <https://doi.org/10.1007/s10543-017-0689-2>
42. Scharstein, R.W., Wilson, H.B.: Electrostatic excitation of a conducting toroid: exact solution and thin-wire approximation. *Electromagnetics* **25**(1), 1–19 (2005)
43. Shapiro, H.S.: *The Schwarz Function and Its Generalization to Higher Dimensions*. University of Arkansas Lecture Notes in the Mathematical Sciences, vol. 9. Wiley, Hoboken (1992)
44. Sloan, I.H.: Error analysis of boundary integral methods. *Acta Numer.* **1**, 287–339 (1992)

45. Tornberg, A.-K., Gustavsson, K.: A numerical method for simulations of rigid fiber suspensions. *J. Comput. Phys.* **215**(1), 172–196 (2006). <https://doi.org/10.1016/j.jcp.2005.10.028>
46. Tornberg, A.-K., Shelley, M.J.: Simulating the dynamics and interactions of flexible fibers in Stokes flows. *J. Comput. Phys.* **196**(1), 8–40 (2004). <https://doi.org/10.1016/j.jcp.2003.10.017>
47. Trefethen, L.N.: *Approximation Theory and Approximation Practice*. SIAM, Philadelphia (2012). ISBN 9781611972399
48. Wala, M., Klöckner, A.: A fast algorithm with error bounds for quadrature by expansion. *J. Comput. Phys.* **374**, 135–162 (2018). <https://doi.org/10.1016/j.jcp.2018.05.006>
49. Walsh, J.L.: *Interpolation and Approximation by Rational Functions in the Complex Domain*, vol. 20. AMS, Philadelphia (1935)
50. Wu, B., Zhu, H., Barnett, A.H., Veerapaneni, S.V.: Solution of Stokes flow in complex nonsmooth 2D geometries via a linear-scaling high-order adaptive integral equation scheme. *J. Comput. Phys.* (2020). [arxiv:1909.00049](https://arxiv.org/abs/1909.00049)

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