Advances in Computational Mathematics (2026) 52:66
https://doi.org/10.1007/s10444-026-10327-2

Newtonian potentials of Legendre polynomials on rectangles have displacement structure

Sheehan Olver[1]

Received: 7 July 2025 / Accepted: 17 June 2026
© The Author(s) 2026

Abstract
Particular solutions of the Poisson equation can be constructed via Newtonian potentials, integrals involving the corresponding Green's function, which in two-dimensions has a logarithmic singularity. The singularity represents a significant challenge for computing the integrals, which is typically overcome via specially designed quadrature methods involving a large number of evaluations of the function and kernel. We present an attractive alternative: we show that Newtonian potentials (and their gradient) applied to (tensor products of) Legendre polynomials can be expressed in terms of complex integrals which satisfy simple and explicit recurrences that can be utilised to *exactly* compute singular integrals, i.e., singular integral quadrature is completely avoided. The inhomogeneous part of the recurrence has low rank structure (its rank is at most three for the Newtonian potential) and hence these recurrences have displacement structure. Using the recurrence directly is a fast approach for evaluation on or near the integration domain that remains accurate for low-degree polynomial approximations, while high-precision arithmetic allows accurate use of the approach for moderate-degree polynomials.

Keywords Newtonian potential · Singular integrals · Displacement structure

Mathematics Subject Classification (2010) 65D30 · 65R10

1 Introduction

We consider the *Newtonian potential* and its gradient over the unit square:

$$\iint_\Omega \log \|\boldsymbol{x} - \boldsymbol{t}\| f(\boldsymbol{t}) \mathrm{d}\boldsymbol{t}, \qquad \iint_\Omega \nabla \log \|\boldsymbol{x} - \boldsymbol{t}\| f(\boldsymbol{t}) \mathrm{d}\boldsymbol{t}$$

Communicated by: Zydrunas Gimbutas

✉ Sheehan Olver
s.olver@imperial.ac.uk

[1] Department of Mathematics, Imperial College, London, UK

Published online: 14 August 2026

where $\Omega := \{(x, y)^\top : -1 \le x, y \le 1\}$. We also consider Newtonian potentials over other rectangles and parallelograms. We allow $\boldsymbol{x}$ to be anywhere in $\mathbb{R}^2$, including $\boldsymbol{x} \in \Omega$. When it is far from Ω the kernel is smooth and traditional quadrature techniques are effective. However, as $\boldsymbol{x}$ approaches Ω, the kernels become increasingly singular, with weak singularities when $\boldsymbol{x} \in \Omega$.

Computing such integrals has a long history with many effective quadrature schemes, for example [3–5, 15, 26, 29], see also related methods for the computing layer potentials (boundary integrals) [1, 7, 16]. We refer the reader for [3, 4, 26] for recent overviews of existing methodology and a detailed description of their limitations for computing the integrals for $\mathbf{x}$ near or on Ω, due to the nearly singular or singular integrals.

In [26], an approach is advocated for computing the integrals via expansions in a monomial basis, whose conditioning can be controlled up to degree 20. In this paper, we investigate a closely related alternative: expand f in Legendre polynomials

$$f(x, y) \approx \sum_{k=0}^{m} \sum_{j=0}^{n} f_{kj} \underbrace{P_k(x) P_j(y)}_{P_{kj}(x,y)}$$

and compute the integrals

$$\iint_\Omega \log \|\boldsymbol{x} - \boldsymbol{t}\| P_{kj}(\boldsymbol{t}) \mathrm{d}\boldsymbol{t}, \qquad \iint_\Omega \nabla \log \|\boldsymbol{x} - \boldsymbol{t}\| P_{kj}(\boldsymbol{t}) \mathrm{d}\boldsymbol{t}$$

in closed form, which will be related to a simple recurrence relationship. This recurrence relationship can be solved extremely fast, albeit with some issues with ill-conditioning for large k or j. Expansion in Legendre polynomials enjoys numerous advantages over monomials, including the existence of fast and stable transforms for computing expansion coefficients [2, 21, 28].

To derive the recurrence relationship, we express the Newtonian potential as the real part of a complex logarithmic integral

$$\mathcal{L} f(z) := \iint_\Omega \log(z - (s + \mathrm{i}t)) f(s, t) \, \mathrm{d}s \, \mathrm{d}t$$

and its gradient in terms of the real and imaginary parts of the Stieltjes integral

$$\mathcal{S} f(z) := \iint_\Omega \frac{f(s, t)}{z - (s + \mathrm{i}t)} \mathrm{d}s \, \mathrm{d}t.$$

Thus, particular solutions to the Poisson equation with the (real-valued) right-hand side $f(x, y)$ are given by

$$\iint_\Omega \log \|\boldsymbol{x} - \boldsymbol{t}\| f(\boldsymbol{t}) \mathrm{d}\boldsymbol{t} = \mathrm{Re}\, \mathcal{L} f(x + \mathrm{i}y) \approx \sum_{k=0}^{m} \sum_{j=0}^{n} f_{kj} \mathrm{Re}\, L_{kj}(x + \mathrm{i}y).$$

where $L_{kj}(z) := \mathcal{L}P_{kj}(z)$ and we identify $\mathbf{x} = (x, y)^\top$ with $z = x + \mathrm{i}y$. The gradient can be related to the Stieltjes integral via:

$$\iint_\Omega \nabla \log \|\boldsymbol{x} - \boldsymbol{t}\| f(\boldsymbol{t}) \mathrm{d}\boldsymbol{t} = \begin{pmatrix} \mathrm{Re} \\ -\mathrm{Im} \end{pmatrix} \mathcal{S} f(x + \mathrm{i}y) \approx \sum_{k=0}^{m} \sum_{j=0}^{n} f_{kj} \begin{pmatrix} \mathrm{Re} \\ -\mathrm{Im} \end{pmatrix} S_{kj}(x + \mathrm{i}y)$$

where $S_{kj}(z) := \mathcal{S}P_{kj}(z)$. The key observation of this article is that $\{L_{kj}(z)\}$ and $\{S_{kj}(z)\}$ both solve simple recurrence relationships that can be used to compute them *exactly*.

In particular, we show that the complex logarithmic and Stieltjes integrals of Legendre polynomials have *displacement structure*. Define the (infinite) matrices

$$L(z) := \begin{pmatrix} L_{00}(z) & L_{10}(z) & \cdots \\ L_{10}(z) & L_{11}(z) & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix}, \qquad S(z) := \begin{pmatrix} S_{00}(z) & S_{10}(z) & \cdots \\ S_{10}(z) & S_{11}(z) & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix}.$$

We show in Theorem 4.4 that $L(z)$ satisfies a Sylvester equation of the form

$$CL(z) - \mathrm{i}L(z)C^\top = F(z)$$

where $F(z)$ has at most rank 3 and C is tridiagonal. We further show in Lemma 4.1 that $S(z)$ satisfies a Sylvester equation of the form

$$(zI - B)S(z) - \mathrm{i}S(z)B^\top = 4\mathbf{e}_0\mathbf{e}_0^\top,$$

that is the right-hand side has rank 1 and B is also tridiagonal. Displacement structure is a powerful property with substantial numerical implications, see [6, 8–10, 19].

The paper is structured as follows:

Section 2 The methodology we develop parallels singular integrals over (one-dimensional) intervals, in particular, it utilises a simple relationship between complex logarithmic and Stieltjes integrals. Numerical solution of Stieltjes integrals via recurrence relationships has a long history going back to Gautschi [13, 14], see also the review in [23]. In this section, we give a brief review of these results and extend them to a modified complex logarithmic integral over a complex interval needed in the derivation of the results on a square.

Section 3 We show that the Stieltjes integrals $S_{kj}(z)$ satisfy a simple "5-point stencil" recurrence relationship by adapting the technique from intervals. By mimicking the connection between complex logarithmic and Stieltjes integrals used on intervals, we show that $L_{kj}(z)$ simultaneously satisfies two "5-point stencil" recurrence relationships with an explicitly constructed inhomogeneous term. Moreover, we can construct the boundary cases $S_{k0}(z)$, $S_{0j}(z)$, $L_{k0}(z)$, and $L_{0j}(z)$ directly.

Section 4 The recurrences can be recast as Sylvester's equations, and the recurrence relationships for $L(z)$ can be combined in a way that exposes the displacement structure in $L(z)$ and $S(z)$.

Section 5 An implication of these results is that we can directly construct $L_{kj}(z)$ and $S_{kj}(z)$ by direct substitution in the recurrence relationship. The resulting algorithm gives an effective and fast approach to construction that is exact. However, in practice, there are errors due to round-off error, so it may only be effective for z near the square and for low-degree polynomials. High-precision arithmetic mitigates this issue, allowing for its effective usage for moderate-degree polynomials.

Section 6 The results can be extended from the unit square to other squares by an affine change-of-variables. We demonstrate this by computing a Newtonian potential over a square mesh, recovering a known solution to roughly machine precision. Other rectangles and parallelograms require modification of the recurrences by a simple scaling and low-rank perturbation.

Section 7 We conclude by discussing the implications of displacement structure to both computation of Newtonian potentials and their gradient, as well as the potential for generalising to other kernels, geometries, and higher dimensions.

Appendix We derive the formulæ for extension to non-square rectangles.

Remark 1.1 An experimental implementation of the results is available in the Julia package MultivariateSingularIntegrals.jl [22].

2 Log and Stieltjes integrals on intervals

A classic result (cf. [14]) is that Stieltjes integrals of weighted orthogonal polynomials satisfy the same three-term recurrence as the orthogonal polynomials themselves, apart from a simple inhomogeneity. Defining

$$\mathcal{S}_{[-1,1]}f(z) := \int_{-1}^{1} \frac{f(t)}{z-t}\mathrm{d}t$$

and $S_k(z) := \mathcal{S}_{[-1,1]}P_k(z)$, the three-term recurrence for Legendre polynomials (assuming $P_{-1}(x) = 0$)

$$x\,P_k(x) = \frac{k}{2k+1}P_{k-1}(x) + \frac{k+1}{2k+1}P_{k+1}(x) \tag{2.1}$$

gives us the three-term recurrence for their Stieltjes integrals:

$$\begin{aligned} z\,S_k(z) &= \int_{-1}^{1} \frac{z-t}{z-t}P_k(t)\mathrm{d}t + \int_{-1}^{1} \frac{t\,P_k(t)}{z-t}\mathrm{d}t \\ &= 2\delta_k + \frac{k}{2k+1}S_{k-1}(x) + \frac{k+1}{2k+1}S_{k+1}(x). \end{aligned} \tag{2.2}$$

Note, we also know the initial condition that can be used to kick off the recurrence,

$$S_0(z) = \int_{-1}^{1} \frac{\mathrm{d}t}{z-t} = \log(z+1) - \log(z-1),$$

thus we can deduce $S_k(z)$ by forward-substitution with the recurrence.

We now demonstrate that a similar process can be used to derive a simple recurrence relationship for $L_k(z) := \mathcal{L}_{[-1,1]} P_k(z)$ where

$$\mathcal{L}_{[-1,1]} f(z) := \int_{-1}^{1} \log(z-t) f(t) \mathrm{d}t,$$

using a construction motivated by [23, Section 7.3]. We first connect the complex logarithmic integral to the Stieltjes integral of a specific ultraspherical polynomial [11, Section 18.3]. For $\lambda \neq 0$, $\lambda > -1/2$, the ultraspherical polynomials $C_n^{(\lambda)}(x)$ are orthogonal with respect to the weight $(1-x^2)^{\lambda-1/2}$ on $[-1, 1]$ with the normalisation constant $2^n(\lambda)_n/n!$ where $(\lambda)_n = \Gamma(\lambda+n)/\Gamma(\lambda)$ is the Pochhammer symbol. The definition can be extended to $\lambda = -1/2$, where the weight is no longer integrable by an explicit expression in terms of Hypergeometric functions [11, 18.5.9], or more explicitly by the formula

$$C_k^{(-1/2)}(x) = \begin{cases} 1 & k = 0 \\ -x & k = 1 \\ \frac{(1-x^2)C_{k-2}^{(3/2)}(x)}{k(k-1)} & \text{otherwise} \end{cases}. \tag{2.3}$$

We will also use the standard extension to $\lambda = -3/2$ below:

$$C_k^{(-3/2)}(x) = \begin{cases} 1 & k = 0 \\ -3x & k = 1 \\ \frac{3(x^2+1)}{2} & k = 2 \\ \frac{x^3-3x}{2} & k = 3 \\ \frac{9(1-x^2)^2 C_{k-4}^{(5/2)}(x)}{k(k-1)(k-2)(k-3)} & \text{otherwise} \end{cases}. \tag{2.4}$$

We will denote the Stieltjes integral of ultraspherical polynomials as

$$S_k^{(\lambda)}(z) := \int_{-1}^{1} \frac{C_k^{(\lambda)}(t)}{z-t} \mathrm{d}t.$$

Since $C_k^{(1/2)}(x) = P_k(x)$, we have $S_k(z) = S_k^{(1/2)}(z)$, but here we are focussed on relating $L_k(z)$ to the case with $\lambda = -1/2$:

Proposition 2.1

$$L_k(z) = -S_{k+1}^{(-1/2)}(z) + \begin{cases} \log(z+1) + \log(z-1) & k = 0 \\ 0 & \text{otherwise} \end{cases}.$$

Proof Integrating by parts allows us to relate the complex logarithmic integral to the Stieltjes integral, which is more amenable to computation as:

$$\mathcal{L}_{[-1,1]}f(z) = \log(z+1)\int_{-1}^{1} f(t)\mathrm{d}t - \int_{-1}^{1}\frac{F(t)}{z-t}\mathrm{d}t$$

where $F(x) = \int_x^1 f(s)\,\mathrm{d}s$. In the case of Legendre polynomials, its indefinite integral is given in terms of certain ultraspherical polynomials [11, 18.9.19]:

$$\int_x^1 P_k(t)\mathrm{d}t = C_{k+1}^{(-1/2)}(x) + \delta_k = \frac{(1-x^2)C_{k-1}^{(3/2)}(x)}{k(k+1)} + \delta_k. \tag{2.5}$$

Thus, since $\int_{-1}^1 P_k(t)\mathrm{d}t = 0$ when $k \neq 0$, we have

$$L_k(z) = -\int_{-1}^{1}\frac{C_{k+1}^{(-1/2)}(t)}{z-t}\mathrm{d}t = -S_{k+1}^{(-1/2)}(z).$$

For $k = 0$, we have

$$L_0(z) = 2\log(z+1) - S_1^{(-1/2)}(z) - \int_{-1}^{1}\frac{1}{z-t}\mathrm{d}t,$$

which simplifies to the desired expression. □

This can be used to deduce a simple recurrence relationship:

Theorem 2.2 *The complex logarithmic integral of Legendre polynomials satisfies a three-term recurrence, using the convention* $L_{-1}(z) = 0$*:*

$$zL_k(z) = \frac{k-1}{2k+1}L_{k-1}(z) + \frac{k+2}{2k+1}L_{k+1}(z) + \lambda_k(z)$$

for

$$\lambda_k(z) := \begin{cases} (z-1)\log(z-1) + (z+1)\log(z+1) & k = 0 \\ -2/3 & k = 1 \\ 0 & \text{otherwise} \end{cases}.$$

Proof We shall deduce the recurrence relationship for $L_k(z)$ using a technique also advocated in [23, 27]. We employ the recurrence relationship for ultraspherical polynomials

$$tC_k^{(\lambda)}(t) = \frac{k+2\lambda-1}{2(k+\lambda)}C_{k-1}^{(\lambda)}(t) + \frac{k+1}{2(k+\lambda)}C_{k+1}^{(\lambda)}(t), \tag{2.6}$$

which remains valid for $\lambda = -1/2$ (and $\lambda = -3/2$). Note we have

$$\int_{-1}^{1} C_k^{(-1/2)}(x)\mathrm{d}x = \int_{-1}^{1} \begin{cases} 1 & k=0 \\ -x & k=1 \\ \frac{(1-x^2)C_{k-2}^{(3/2)}(x)}{(k-1)k} & \text{otherwise} \end{cases} \mathrm{d}x = \begin{cases} 2 & k=0 \\ 2/3 & k=2 \\ 0 & \text{otherwise} \end{cases}. \tag{2.7}$$

It follows from Proposition 2.1 that

$$\begin{aligned} zL_k(z) &= -\underbrace{\int_{-1}^{1} \frac{z-t}{z-t} C_{k+1}^{(-1/2)}(t)\mathrm{d}t}_{\text{use (2.7)}} - \int_{-1}^{1} \frac{1}{z-t} \underbrace{t C_{k+1}^{(-1/2)}(t)}_{\text{use (2.6)}} \mathrm{d}t \\ &\quad + \begin{cases} z\log(z+1) + z\log(z-1) & k=0 \\ 0 & \text{otherwise} \end{cases} \\ &= -\int_{-1}^{1} \frac{1}{z-t} \left(\frac{k-1}{2k+1} C_k^{(-1/2)}(t) + \frac{k+2}{2k+1} C_{k+2}^{(-1/2)}(t) \right) \mathrm{d}t \\ &\quad + \begin{cases} z\log(z+1) + z\log(z-1) & k=0 \\ -2/3 & k=1 \\ 0 & \text{otherwise} \end{cases} \\ &= \frac{k-1}{2k+1} L_{k-1}(z) + \frac{k+2}{2k+1} L_{k+1}(z). \\ &\quad + \begin{cases} \int_{-1}^{1} \frac{1}{z-t}\mathrm{d}t + z\log(z+1) + z\log(z-1) & k=0 \\ -2/3 & k=1 \\ 0 & \text{otherwise} \end{cases}, \end{aligned}$$

which simplifies to the desired result. □

To extend the results to 2D, we will need to work with the following variant, where we complexify the integration variable:

$$\mathcal{M}_{[-1,1]} f(z) := \int_{-1}^{1} \log(z - \mathrm{i}t) f(t)\mathrm{d}t$$

and $M_k(z) := \mathcal{M}_{[-1,1]} P_k(z)$. This is a subtle change: writing $z = x + \mathrm{i}y$, if $-1 \leq y \leq 1$ and $x < 0$ then the integral passes over the branch cut of the logarithm, which we take to be along the negative real axis. We nevertheless can relate this to $\mathcal{L}$:

Proposition 2.3 *For $z = x + \mathrm{i}y$, we have*

$$\mathcal{M}_{[-1,1]} f(z) = \mathcal{L}_{[-1,1]} f(-\mathrm{i}z) + \frac{\mathrm{i}\pi}{2} \begin{cases} \int_{-1}^{1} f(t)\mathrm{d}t & x > 0 \text{ or } y \geq 1 \\ -3\int_{-1}^{1} f(t)\mathrm{d}t & x \leq 0 \text{ and } y \leq -1 \\ \left(\int_{-1}^{y} - 3\int_{y}^{1} \right) f(t)\mathrm{d}t & x < 0 \text{ and } -1 < y < 1 \end{cases}$$

Proof We use the formula (using the principle value of the argument/phase, i.e., $-\pi < \arg z \le \pi$, see [11, Sections 1.9(i) & 4.2(i)])

$$\log(\mathrm{i}z) = \log|z| + \mathrm{i}\arg(\mathrm{i}z) = \log z + \frac{\mathrm{i}\pi}{2}\begin{cases}1 & -\pi < \arg z \le \pi/2\\ -3 & \text{otherwise}\end{cases}$$

to deduce

$$\begin{aligned}\int_{-1}^{1} f(t)\log(z-\mathrm{i}t)\mathrm{d}t &= \int_{-1}^{1} f(t)\log(-\mathrm{i}z-t)\mathrm{d}t\\ &\quad + \frac{\mathrm{i}\pi}{2}\int_{-1}^{1} f(t)\begin{cases}1 & -\pi < \arg(-\mathrm{i}z-t) \le \pi/2\\ -3 & \text{otherwise}\end{cases}\mathrm{d}t\end{aligned}$$

The result follows by considering the region where $-\mathrm{i}z - t = y - t - \mathrm{i}x$ is under the three different cases:

1. If $x > 0$ then $\mathrm{Im}\,(y-t-\mathrm{i}x) = -x < 0$ and hence $-\pi < \arg(-\mathrm{i}z-t) < 0$, whilst if $y \ge 1$ and $x \le 0$ we have $\mathrm{Re}\,(y-t-\mathrm{i}x) \ge y-1 \ge 0$ and $0 \le \arg(-\mathrm{i}z-t) \le \pi/2$.
2. If $x \le 0$ and $y < -1$ then $\mathrm{Im}\,(y-t-\mathrm{i}x) = -x \ge 0$ and $\mathrm{Re}\,(y-t-\mathrm{i}x) \le y+1 < 0$ and hence $\pi/2 < \arg(-\mathrm{i}z - t) \le \pi$. If $y = -1$, the integrand only differs at a single point, so its value is unchanged.
3. If $x \le 0$ and $-1 < y < 1$ then for $t < y$, we have $\mathrm{Re}\,(y - t - \mathrm{i}x) > 0$ whilst for $t > y$, we have $\mathrm{Re}\,(y - t - \mathrm{i}x) < 0$ and hence the integrand changes value at the point $y = t$. □

Specialising to Legendre polynomials and using (2.7) to simplify the expressions results in the following:

Theorem 2.4 *For $z = x + \mathrm{i}y$, we have*

$$\begin{aligned}M_k(z) &= L_k(-\mathrm{i}z) + \mathrm{i}\pi\delta_k\begin{cases}1 & x > 0 \text{ or } y \ge 1\\ -3 & x \le 0 \text{ and } y \le -1\\ -1 & x < 0 \text{ and } -1 < y < 1\end{cases}\\ &\quad - 2\mathrm{i}\pi\begin{cases}C_{k+1}^{(-1/2)}(y) & x < 0 \text{ and } -1 < y < 1\\ 0 & \text{otherwise}\end{cases}.\end{aligned}$$

We can deduce a three-term recurrence for $M_k(z)$ as well:

Corollary 2.5 *For $z = x + \mathrm{i}y$, we have*

$$zM_k(z) = \mathrm{i}\frac{k-1}{2k+1}M_{k-1}(z) + \mathrm{i}\frac{k+2}{2k+1}M_{k+1}(z) + \mu_k(z)$$

for

$$\mu_k(z) := \begin{cases} (z-\mathrm{i})\log(z-\mathrm{i}) + (z+\mathrm{i})\log_-(z+\mathrm{i}) & k=0 \\ -2\mathrm{i}/3 & k=1 \\ 0 & \textit{otherwise} \end{cases} - \begin{cases} 2\mathrm{i}\pi x C_{k+1}^{(-1/2)}(y) & x<0 \textit{ and } -1<y<1 \\ 0 & \textit{otherwise} \end{cases},$$

where we define the logarithm with the other limit along its branch cut:

$$\log_- z := \begin{cases} \log|z| - \mathrm{i}\pi & y=0 \textit{ and } x<0 \\ \log z & \textit{otherwise} \end{cases}.$$

Proof Considering the different cases, we find that:

$$(z-\mathrm{i})\log(z-\mathrm{i}) + (z+\mathrm{i})\log_-(z+\mathrm{i}) = (z-\mathrm{i})\log(-\mathrm{i}z-1) + (z+\mathrm{i})\log(-\mathrm{i}z+1) + \pi \begin{cases} \mathrm{i}z & x>0 \text{ or } y\geq 1 \\ -3\mathrm{i}z & x\leq 0 \text{ and } y\leq -1 \\ -\mathrm{i}z-2 & x<0 \text{ and } -1<y<1 \end{cases}.$$

For the $k=0$ case, we thus find, using Theorem 2.4,

$$\begin{aligned} zM_0(z) &= \mathrm{i}(-\mathrm{i}zL_0(-\mathrm{i}z)) + \mathrm{i}\pi z \begin{cases} 1 & x>0 \text{ or } y\geq 1 \\ -3 & x\leq 0 \text{ and } y\leq -1 \\ 2y-1 & x<0 \text{ and } -1<y<1 \end{cases} \\ &= 2\mathrm{i}L_1(-\mathrm{i}z) + (z-\mathrm{i})\log(-\mathrm{i}z-1) + (z+\mathrm{i})\log(-\mathrm{i}z+1) \\ &\quad + \mathrm{i}\pi z \begin{cases} 1 & x>0 \text{ or } y\geq 1 \\ -3 & x\leq 0 \text{ and } y\leq -1 \\ 2y-1 & x<0 \text{ and } -1<y<1 \end{cases} \\ &= 2\mathrm{i}M_1(z) + \pi \begin{cases} 2\mathrm{y}\mathrm{i}z + 2 - 4C_2^{(-1/2)}(y) & x<0 \text{ and } -1<y<1 \\ 0 & \text{otherwise} \end{cases} \\ &\quad + (z-\mathrm{i})\log(z-\mathrm{i}) + (z+\mathrm{i})\log(z+\mathrm{i}). \end{aligned}$$

This reduces to the result when we observe, using (2.3),

$$2\mathrm{y}\mathrm{i}z + 2 - 4C_2^{(-1/2)}(y) = 2x\mathrm{y}\mathrm{i} - 2y^2 + 2 - 2(1-y^2) = 2x\mathrm{y}\mathrm{i} = -2\mathrm{i}xC_1^{(-1/2)}(y).$$

For $k > 0$, we have, using the three-term recurrence (2.6) to expand $yC_{k+1}^{(-1/2)}(y)$,

$$\begin{aligned} zM_k(z) &= \mathrm{i}(-\mathrm{i}zL_k(-\mathrm{i}z)) - 2\mathrm{i}\pi z \begin{cases} C_{k+1}^{(-1/2)}(y) & x < 0 \text{ and } -1 < y < 1 \\ 0 & \text{otherwise} \end{cases} \\ &= \mathrm{i}\frac{k-1}{2k+1}L_{k-1}(-\mathrm{i}z) + \mathrm{i}\frac{k+2}{2k+1}L_{k+1}(-\mathrm{i}z) - 2\mathrm{i}/3\delta_{k1} \\ &\quad + 2\pi \begin{cases} \frac{k-1}{2k+1}C_k^{(-1/2)}(y) + \frac{k+2}{2k+1}C_{k+2}^{(-1/2)}(y) & x < 0 \text{ and } -1 < y < 1 \\ 0 & \text{otherwise} \end{cases} \\ &\quad - 2\mathrm{i}\pi x \begin{cases} C_{k+1}^{(-1/2)}(y) & x < 0 \text{ and } -1 < y < 1 \\ 0 & \text{otherwise} \end{cases} \end{aligned}$$

which reduces to the result via Theorem 2.4. □

Remark 2.6 The proceeding theorem may not apply for complex floating-point numbers due to the existence of a negative 0: since $-\mathrm{i}(0.0 + \mathrm{i}y) = y - 0.0\mathrm{i}$, some implementations of the logarithm may be evaluated as if it is below the branch cut. This can be avoided by specialising the implementation of the logarithm for complex floating-point numbers.

3 Complex logarithmic and Stieltjes integrals on squares

We now turn to deducing recurrence relationships for the two-dimensional integrals $S_{kj}(z)$ and $L_{kj}(z)$.

3.1 Stieltjes integrals

We begin with deriving a recurrence relationship for the 2D Stieltjes integral

$$S_{kj}(z) := \iint_\Omega \frac{P_k(s)P_j(t)}{z - (s + \mathrm{i}t)} \mathrm{d}s\, \mathrm{d}t.$$

Proposition 3.1 *The Stieltjes integral $S_{kj}(z)$ satisfies the "5-point stencil" recurrence relationships, using the convention $S_{-1,j}(z) = S_{k,-1} = 0$:*

$$\begin{aligned} zS_{kj}(z) &= \mathrm{i}\frac{j}{2j+1}S_{k,j-1}(z) + \mathrm{i}\frac{j+1}{2j+1}S_{k,j+1}(z) \\ &\quad + \frac{k}{2k+1}S_{k-1,j}(z) + \frac{k+1}{2k+1}S_{k+1,j}(z) + 4\delta_j\delta_k. \end{aligned}$$

Proof This follows from the three-term recurrence for $S_k(z)$. That is, first note that

$$S_{kj}(z) = \int_{-1}^1 P_k(s) \int_{-1}^1 \frac{P_j(t)}{z - (s + \mathrm{i}t)} \mathrm{d}t\, \mathrm{d}s = -\mathrm{i}\int_{-1}^1 P_k(s)S_j(-\mathrm{i}z + \mathrm{i}s)\mathrm{d}s. \quad (3.1)$$

We, therefore, deduce by writing $-\mathrm{i}z = -\mathrm{i}z + \mathrm{i}s - \mathrm{i}s$ and using the appropriate three-term recurrences:

$$\begin{aligned} zS_{kj}(z) &= \int_{-1}^{1} P_k(s) \underbrace{(-\mathrm{i}z+\mathrm{i}s)S_j(-\mathrm{i}z+\mathrm{i}s)}_{\text{use (2.2)}}\,\mathrm{d}s - \mathrm{i}\int_{-1}^{1} \underbrace{sP_k(s)}_{\text{use (2.1)}}\, S_j(-\mathrm{i}z+\mathrm{i}s)\mathrm{d}s \\ &= \frac{j}{2j+1}\int_{-1}^{1} P_k(s)S_{j-1}(-\mathrm{i}z+\mathrm{i}s)\mathrm{d}s + \frac{j+1}{2j+1}\int_{-1}^{1} P_k(s)S_{j+1}(-\mathrm{i}z+\mathrm{i}s)\mathrm{d}s \\ &\quad + 2\delta_{j0}\int_{-1}^{1} P_k(s)\mathrm{d}s - \mathrm{i}\frac{k}{2k+1}\int_{-1}^{1} P_{k-1}(s)S_j(-\mathrm{i}z+\mathrm{i}s)\mathrm{d}s \\ &\quad - \mathrm{i}\frac{k+1}{2k+1}\int_{-1}^{1} P_{k+1}(s)S_j(-\mathrm{i}z+\mathrm{i}s)\mathrm{d}s \\ &= \mathrm{i}\frac{j}{2j+1}S_{k,j-1}(z) + \mathrm{i}\frac{j+1}{2j+1}S_{k,j+1}(z) \\ &\quad + \frac{k}{2k+1}S_{k-1,j}(z) + \frac{k+1}{2k+1}S_{k+1,j}(z) + 4\delta_j\delta_k \end{aligned}$$

where we used (3.1) to simplify the expression in the last equality. □

To use this recurrence, in practice, we need to know the initial conditions. It turns out we can deduce these directly in terms of the one-dimensional singular integrals, which are deducible directly from recurrences:

Proposition 3.2

$$\begin{aligned} S_{k0}(z) &= (-1)^k\mathrm{i}(M_k(-1-\mathrm{i}z) - M_k(1-\mathrm{i}z)), \\ S_{0j}(z) &= M_j(z+1) - M_j(z-1) \end{aligned}$$

Proof We use the fact that we have an explicit form for $S_0(z)$:

$$\begin{aligned} S_{k0}(z) &= -\mathrm{i}\int_{-1}^{1} P_k(s)S_0(-\mathrm{i}z+\mathrm{i}s)\mathrm{d}s \\ &= -\mathrm{i}\int_{-1}^{1} P_k(s)\left[\log(1-\mathrm{i}z+\mathrm{i}s) - \log(-\mathrm{i}z+\mathrm{i}s-1)\right]\mathrm{d}s \\ &= \mathrm{i}(-1)^k\int_{-1}^{1} P_k(s)\left[\log(-1-\mathrm{i}z-\mathrm{i}s) - \log(1-\mathrm{i}z-\mathrm{i}s)\right]\mathrm{d}s \\ &= (-1)^k\mathrm{i}(M_k(-1-\mathrm{i}z) - M_k(1-\mathrm{i}z)). \end{aligned}$$

Similarly,

$$S_{0j}(z) = \int_{-1}^{1} P_j(t)S_0(z-\mathrm{i}t)\mathrm{d}t$$

$$
\begin{aligned}
&= \int_{-1}^{1} P_j(t)\left[\log(1+z-\mathrm{i}t) - \log(z-\mathrm{i}t-1)\right]\mathrm{d}s \\
&= M_j(z+1) - M_j(z-1).
\end{aligned}
$$

□

3.2 Complex logarithmic integral

The key idea in the one-dimensional complex logarithmic integrals was their reduction to a Stieltjes integral, which led to a recurrence relationship. Here, we observe that one can deploy this same technique on two-dimensional complex logarithmic integrals.

Theorem 3.3 *The complex logarithmic integral of Legendre polynomials satisfies two "5-point stencil"-like recurrence relations, using the convention* $L_{-1,j}(z) = L_{k,-1} = 0$*:*

$$
\begin{aligned}
zL_{kj} &= \frac{k-1}{2k+1}L_{k-1,j} + \frac{k+2}{2k+1}L_{k+1,j} + \mathrm{i}\left(\frac{j}{2j+1}L_{k,j-1} + \frac{j+1}{2j+1}L_{k,j+1}\right) + F^{(1)}_{kj}(z), \\
&= \frac{k}{2k+1}L_{k-1,j} + \frac{k+1}{2k+1}L_{k+1,j} + \mathrm{i}\left(\frac{j-1}{2j+1}L_{k,j-1} + \frac{j+2}{2j+1}L_{k,j+1}\right) + F^{(2)}_{kj}(z),
\end{aligned}
$$

where we take $L_{-1,j} = L_{k,-1} = 0$ *and have*

$$
\begin{aligned}
F^{(1)}_{0j}(z) &= \mathrm{i}\frac{M_{j+1}(z-1) + M_{j+1}(z+1) - M_{j-1}(z-1) - M_{j-1}(z+1)}{2j+1} \\
&\quad + \mu_j(z-1) + \mu_j(z+1), \\
F^{(1)}_{10}(z) &= -4/3, \\
F^{(1)}_{kj}(z) &= 0 \qquad \text{for } k \neq 0 \text{ and } (k,j) \neq (1,0), \\
F^{(2)}_{k0}(z) &= \frac{L_{k+1}(z-\mathrm{i}) + L_{k+1}(z+\mathrm{i}) - L_{k-1}(z-\mathrm{i}) - L_{k-1}(z+\mathrm{i})}{2k+1} \\
&\quad + \lambda_k(z-\mathrm{i}) + \lambda_k(z+\mathrm{i}) + \beta_{k0}(z) \\
F^{(2)}_{01}(z) &= -4\mathrm{i}/3 + \beta_{01}(z), \\
F^{(2)}_{kj}(z) &= \beta_{kj}(z) \qquad \text{for } j \neq 0 \text{ and } (k,j) \neq (0,1)
\end{aligned}
$$

where

$$
\begin{aligned}
\beta_{kj}(z) &= 2\mathrm{i}\pi C^{(-1/2)}_{j+1}(y) \\
&\quad \times \begin{cases} C^{(-3/2)}_{k+2}(x)/3 - x\delta_k + \delta_{k1}/3 & -1 < x < -1 \text{ and } -1 \le y < 1 \\ -2x\delta_k + 2\delta_{k1}/3 & x \le -1 \text{ and } -1 \le y < 1 \\ 0 & \text{otherwise} \end{cases}.
\end{aligned}
$$

Proof For the first recurrence relationship, we roughly follow the approach of the two-dimensional Stieltjes integral using the recurrence in Theorem 2.2. First note that we can write:

$$L_{kj}(z) = \int_{-1}^{1} P_j(t) \int_{-1}^{1} P_k(s) \log(z - (s + \mathrm{i}t))\mathrm{d}s\mathrm{d}t = \int_{-1}^{1} P_j(t) L_k(z - \mathrm{i}t)\mathrm{d}t.$$

Therefore, we have:

$$\begin{aligned} zL_{kj}(z) &= \int_{-1}^{1} P_j(t)(z - \mathrm{i}t)L_k(z - \mathrm{i}t)\mathrm{d}t + \mathrm{i}\int_{-1}^{1} tP_j(t)L_k(z - \mathrm{i}t)\mathrm{d}t \\ &= \frac{k-1}{2k+1}\underbrace{\int_{-1}^{1} P_j(t)L_{k-1}(z - \mathrm{i}t)\mathrm{d}t}_{L_{k-1,j}(z)} + \frac{k+2}{2k+1}\underbrace{\int_{-1}^{1} P_j(t)L_{k+1}(z - \mathrm{i}t)\mathrm{d}t}_{L_{k+1,j}(z)} \\ &\quad + \mathrm{i}\frac{j}{2j+1}\underbrace{\int_{-1}^{1} P_{j-1}(t)L_k(z - \mathrm{i}t)\mathrm{d}t}_{L_{k,j-1}(z)} + \mathrm{i}\frac{j+1}{2j+1}\underbrace{\int_{-1}^{1} P_{j+1}(t)L_k(z - \mathrm{i}t)\mathrm{d}t}_{L_{k,j+1}(z)} \\ &\quad + \int_{-1}^{1} P_j(t)\lambda_k(z - \mathrm{i}t)\mathrm{d}t. \end{aligned}$$

To compute $\int_{-1}^{1} P_j(t)\lambda_k(z - \mathrm{i}t)\mathrm{d}t$, we use Corollary 2.5 to see that

$$\begin{aligned} \int_{-1}^{1} P_j(t)(z - \mathrm{i}t)\log(z - \mathrm{i}t)\mathrm{d}t &= zM_j(z) - \mathrm{i}\int_{-1}^{1} tP_j(t)\log(z - \mathrm{i}t)\mathrm{d}s \\ &= \mathrm{i}\frac{j-1}{2j+1}M_{j-1}(z) + \mathrm{i}\frac{j+2}{2j+1}M_{j+1}(z) \\ &\quad - \mathrm{i}\frac{j}{2j+1}M_{j-1}(z) - \mathrm{i}\frac{j+1}{2j+1}M_{j+1}(z) + \mu_j(z) \\ &= -\frac{\mathrm{i}}{2j+1}M_{j-1}(z) + \frac{\mathrm{i}}{2j+1}M_{j+1}(z) + \mu_j(z). \end{aligned} \tag{3.2}$$

Hence, we can simplify:

$$\begin{aligned} &\int_{-1}^{1} P_j(t)\lambda_0(z - \mathrm{i}t)\mathrm{d}t \\ &\quad = \int_{-1}^{1} P_j(t)\left[(z - \mathrm{i}t - 1)\log(z - \mathrm{i}t - 1) + (z - \mathrm{i}t + 1)\log(z - \mathrm{i}t + 1)\right]\mathrm{d}t \\ &\quad = -\frac{\mathrm{i}}{2j+1}\left(M_{j-1}(z-1) + M_{j-1}(z+1)\right) \\ &\qquad + \frac{\mathrm{i}}{2j+1}(M_{j+1}(z-1) + M_{j+1}(z+1)) + (\mu_j(z-1) + \mu_j(z+1)), \end{aligned}$$

$$\int_{-1}^{1} P_j(t)\lambda_1(z-\mathrm{i}t)\mathrm{d}t = -2/3\int_{-1}^{1} P_j(t)\mathrm{d}t = -4/3,$$
$$\int_{-1}^{1} P_j(t)\lambda_k(z-\mathrm{i}t)\mathrm{d}t = 0, \qquad (k>1).$$

We have thus successfully derived the first recurrence.

For the second recurrence, we use Corollary 2.5 to reduce the integral. Now write

$$L_{kj}(z) = \int_{-1}^{1} P_k(s)\int_{-1}^{1} P_j(t)\log(z-(s+\mathrm{i}t))\mathrm{d}t\mathrm{d}s = \int_{-1}^{1} P_k(s)M_j(z-s)\mathrm{d}s$$

so that

$$\begin{aligned} zL_{kj}(z) &= \int_{-1}^{1} P_k(s)(z-s)M_j(z-s)\mathrm{d}s + \int_{-1}^{1} sP_k(s)M_j(z-s)\mathrm{d}s \\ &= \mathrm{i}\frac{j-1}{2j+1}\underbrace{\int_{-1}^{1} P_k(s)M_{j-1}(z-s)\mathrm{d}s}_{L_{k,j-1}(z)} + \mathrm{i}\frac{j+2}{2j+1}\underbrace{\int_{-1}^{1} P_k(s)M_{j+1}(z-s)\mathrm{d}s}_{L_{k,j+1}(z)} \\ &\quad + \frac{k}{2k+1}\underbrace{\int_{-1}^{1} P_{k-1}(s)M_j(z-s)\mathrm{d}s}_{L_{k-1,j}(z)} + \frac{k+1}{2k+1}\underbrace{\int_{-1}^{1} P_{k+1}(s)M_j(z-s)\mathrm{d}s}_{L_{k+1,j}(z)} \\ &\quad + \int_{-1}^{1} P_k(s)\mu_j(z-s)\mathrm{d}s. \end{aligned}$$

where μ_j is defined in Corollary 2.5. We thus need only simplify the last integral. Note that, using Theorem 2.2,

$$\begin{aligned} \int_{-1}^{1} P_k(s)(z-s)\log(z-s)\mathrm{d}s &= zL_k(z) - \int_{-1}^{1} sP_k(s)\log(z-s)\mathrm{d}s \\ &= \frac{k-1}{2k+1}L_{k-1}(z) + \frac{k+2}{2k+1}L_{k+1}(z) + \lambda_k(z) - \frac{k}{2k+1}L_{k-1}(z) - \frac{k+1}{2k+1}L_{k+1}(z) \\ &= \frac{L_{k+1}(z)-L_{k-1}(z)}{2k+1} + \lambda_k(z). \end{aligned} \tag{3.3}$$

If $-1<x<1$ we have, using (2.5) and (2.1),

$$\begin{aligned} \int_{-1}^{1} P_k(s)\begin{cases} x-s & x<s \\ 0 & \text{otherwise} \end{cases}\mathrm{d}s &= x\int_{x}^{1} P_k(s)\mathrm{d}s - \int_{x}^{1} sP_k(s)\mathrm{d}s \\ &= x(C_{k+1}^{(-1/2)}(x)+\delta_k) - \frac{k}{2k+1}\int_{x}^{1} P_{k-1}(s)\mathrm{d}s - \frac{k+1}{2k+1}\int_{x}^{1} P_{k+1}(s)\mathrm{d}s \\ &= \frac{C_{k+2}^{(-1/2)}(x)-C_k^{(-1/2)}(x)}{2k+1} + x\delta_k - \frac{\delta_{k1}}{3} = -\frac{C_{k+2}^{(-3/2)}(x)}{3} + x\delta_k - \frac{\delta_{k1}}{3} \end{aligned} \tag{3.4}$$

where the last equality comes from the relationship [11, 18.9.7]

$$(n+\lambda)C_n^{(\lambda)} = \lambda(C_n^{(\lambda+1)} - C_{n-2}^{(\lambda+1)})$$

with $\lambda = -3/2$. If $x \le -1$ then

$$\int_{-1}^{1} P_k(s) \begin{cases} x-s & x < s \\ 0 & \text{otherwise} \end{cases} \mathrm{d}s = \int_{-1}^{1} P_k(s)(x-s)\mathrm{d}s = \begin{cases} 2x & k=0 \\ -2/3 & k=1 \\ 0 & \text{otherwise} \end{cases}. \tag{3.5}$$

Furthermore,

$$\begin{aligned} \int_{-1}^{1} P_k(s)(z-s)\log_-(z-s)\mathrm{d}s &= \int_{-1}^{1} P_k(s)(z-s)\log(z-s)\mathrm{d}s \\ &\quad - 2\mathrm{i}\pi \underbrace{\int_{-1}^{1} P_k(s) \begin{cases} x-s & y=0 \text{ and } x<s \\ 0 & \text{otherwise} \end{cases} \mathrm{d}s}_{\text{use (3.4) and (3.5)}} \\ &= \frac{L_{k+1}(z) - L_{k-1}(z)}{2k+1} + \lambda_k(z) \\ &\quad + 2\mathrm{i}\pi \begin{cases} C_{k+2}^{(-3/2)}(x)/3 - x\delta_k + \delta_{k1}/3 & y=0 \text{ and } -1<x<1 \\ -2x\delta_k + 2\delta_{k1}/3 & y=0 \text{ and } x \le -1 \\ 0 & \text{otherwise} \end{cases}. \end{aligned} \tag{3.6}$$

These combine to give:

$$\begin{aligned} \int_{-1}^{1} P_k(s)\mu_0(z-s)\mathrm{d}s &= \underbrace{\int_{-1}^{1} P_k(s)(z-\mathrm{i}-s)\log(z-\mathrm{i}-s)\mathrm{d}s}_{\text{use (3.3)}} \\ &\quad + \underbrace{\int_{-1}^{1} P_k(s)(z+\mathrm{i}-s)\log_-(z+\mathrm{i}-s)\mathrm{d}s}_{\text{use (3.6)}} \\ &\quad + 2\mathrm{i}\pi\, y \underbrace{\int_{-1}^{1} P_k(s) \begin{cases} x-s & x<s \text{ and } -1<y<1 \\ 0 & \text{otherwise} \end{cases} \mathrm{d}s}_{\text{use (3.4) and (3.5)}} \\ &= \frac{L_{k+1}(z-\mathrm{i}) + L_{k+1}(z+\mathrm{i}) - L_{k-1}(z-\mathrm{i}) - L_{k-1}(z+\mathrm{i})}{2k+1} \\ &\quad + \lambda_k(z-\mathrm{i}) + \lambda_k(z+\mathrm{i}) \end{aligned}$$

$$+ 2\mathrm{i}\pi \begin{cases} \frac{C_{k+2}^{(-3/2)}(x)}{3} - x\delta_k + \delta_{k1}/3 & y = -1 \text{ and } -1 < x < 1 \\ -2x\delta_k + 2\delta_{k1}/3 & y = -1 \text{ and } x \le -1 \\ 0 & \text{otherwise} \end{cases}$$

$$- 2\mathrm{i}\pi\, y \begin{cases} C_{k+2}^{(-3/2)}(x)/3 - x\delta_k + \delta_{k1}/3 & -1 < x < 1 \text{ and } -1 < y < 1 \\ -2x\delta_k + 2\delta_{k1}/3 & -1 < y < 1 \text{ and } x \le -1 \\ 0 & \text{otherwise,} \end{cases}$$

where the last two terms can be combined to get the desired result. The computation for other choices of j is more straightforward: for $j \neq 0$, we have (using $C_{j+1}^{(-1/2)}(-1) = 0$ to ensure consistency when $y = -1$)

$$\begin{aligned} &\int_{-1}^{1} P_k(s)\mu_j(z-s)\mathrm{d}s \\ &\quad = -\frac{4\mathrm{i}}{3}\delta_k\delta_{j1} - 2\mathrm{i}\pi C_{j+1}^{(-1/2)}(y)\int_{-1}^{1} P_k(s) \begin{cases} x - s & x < s \text{ and } -1 < y < 1 \\ 0 & \text{otherwise} \end{cases} \mathrm{d}s \\ &\quad = -\frac{4\mathrm{i}}{3}\delta_k\delta_{j1} \\ &\qquad + 2\mathrm{i}\pi C_{j+1}^{(-1/2)}(y) \begin{cases} C_{k+2}^{(-3/2)}(x)/3 - x\delta_k + \delta_{k1}/3 & -1 < x < 1 \text{ and } -1 < y < 1 \\ -2x\delta_k + 2\delta_{k1}/3 & -1 < y < 1 \text{ and } x \le -1 \\ 0 & \text{otherwise.} \end{cases} \end{aligned}$$

□

To kick off the recurrence, we also need an explicit form for L_{00}:

Proposition 3.4

$$L_{00}(z) = (1-z)M_0(z-1) + \mathrm{i}M_1(z-1) + (1+z)M_0(z+1) - \mathrm{i}M_1(z+1) - 4.$$

Proof This follows from:

$$\begin{aligned} L_{00}(z) &= \int_{-1}^{1}\int_{-1}^{1} \log(z - (s + \mathrm{i}t))\,\mathrm{d}s\,\mathrm{d}t = \int_{-1}^{1} L_0(z - \mathrm{i}t)\,\mathrm{d}t \\ &= \int_{-1}^{1} \left[(1 - z + \mathrm{i}t)\log(z - \mathrm{i}t - 1) + (1 + z - \mathrm{i}t)\log(z - \mathrm{i}t + 1) - 2\right] \mathrm{d}t \\ &= (1-z)M_0(z-1) + \mathrm{i}M_1(z-1) + (1+z)M_0(z+1) - \mathrm{i}M_1(z+1) - 4. \end{aligned}$$

□

Having two different 5-point stencils allows us to eliminate degrees of freedom. In particular, we can deduce inhomogeneous three-term recurrence relationships that give the first row and column:

Corollary 3.5

$$\begin{aligned} zL_{k0}(z) &= \frac{k-2}{2k+1}L_{k-1,0}(z) + \frac{k+3}{2k+1}L_{k+1,0}(z) + 2F_{k0}^{(1)}(z) - F_{k0}^{(2)}(z) \\ zL_{0j}(z) &= \mathrm{i}\frac{j-2}{2j+1}L_{0,j-1}(z) + \mathrm{i}\frac{j+3}{2j+1}L_{0,j+1}(z) + 2F_{0j}^{(2)}(z) - F_{0j}^{(1)}(z) \end{aligned}$$

Proof The first part follows from multiplying the first formula in Theorem 3.3 by 2 and subtracting the second formula, and the second part follows the other way round. □

4 Recasting as a Sylvester equation

We can rewrite the recurrence relationhips as (infinite) Sylvester equations. The Stieltjes integral for Legendre polynomials has a displacement structure with rank 1:

Lemma 4.1 *The (infinite) matrix $S(z)$ satisfies the recurrence*

$$zS(z) = BS(z) + \mathrm{i}S(z)B^\top + 4\mathbf{e}_0\mathbf{e}_0^\top$$

for the matrix

$$B = \begin{pmatrix} 0 & 1 & & & \\ 1/3 & 0 & 2/3 & & \\ & 2/5 & 0 & 3/5 & \\ & & 3/7 & 0 & 4/7 \\ & & & 4/9 & 0 & \ddots \\ & & & & \ddots & \ddots \end{pmatrix}.$$

The logarithmic integral for Legendre polynomials satisfies two Sylvester equations for

$$F^{(\ell)}(z) = \begin{pmatrix} F_{00}^{(\ell)}(z) & F_{10}^{(\ell)}(z) & \cdots \\ F_{10}^{(\ell)}(z) & F_{11}^{(\ell)}(z) & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix}:$$

Lemma 4.2

$$\begin{aligned} zL(z) &= AL(z) + \mathrm{i}L(z)B^\top + F^{(1)}(z), \\ zL(z) &= BL(z) + \mathrm{i}L(z)A^\top + F^{(2)}(z) \end{aligned}$$

where B was defined in Lemma 4.1 and

$$A = \begin{pmatrix} 0 & 2 & & & \\ 0 & 0 & 1 & & \\ 1/5 & 0 & 4/5 & & \\ & 2/7 & 0 & 5/7 & \\ & & 3/9 & 0 & \ddots \\ & & & \ddots & \ddots \end{pmatrix}.$$

Remark 4.3 Note that B is the multication-by-x matrix for Legendre polynomials and A is the principle subsection of the multiplication-by-x matrix for the ultraspherical polynomials $C_k^{(-1/2)}(x)$.

We can combine these two recurrences to deduce a single recurrence where the entire z dependence is in the right-hand side:

Theorem 4.4

$$CL(z) - \mathrm{i}L(z)C^\top = F(z)$$

where $F(z) := F^{(2)}(z) - F^{(1)}(z)$ *has at most rank 3 and*

$$C := A - B = \begin{pmatrix} 0 & 1 & & & \\ -1/3 & 0 & 1/3 & & \\ & -1/5 & 0 & 1/5 & \\ & & -1/7 & 0 & \ddots \\ & & & \ddots & \ddots \end{pmatrix}.$$

Proof The form of the Sylvester equation follows from subtracting the Sylvester equations in the preceding lemma. The fact that $F(z)$ has rank 3 follows since dropping the first row and column, we find, for $z \in \Omega$,

$$\begin{aligned} F(z)[2:\infty, 2:\infty] &= \begin{pmatrix} \beta_{11}(z) & \beta_{12}(z) & \cdots \\ \beta_{21}(z) & \beta_{22}(z) & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix} \\ &= 2\mathrm{i}\pi \begin{pmatrix} C_3^{(-3/2)}(x) + 1/3 \\ C_4^{(-3/2)}(x) \\ C_5^{-3/2}(x) \\ \vdots \end{pmatrix} \begin{pmatrix} C_2^{(-1/2)}(y) & C_3^{(-1/2)}(y) & \cdots \end{pmatrix} \end{aligned}$$

which has rank 1 (and similarly for $z \notin \Omega$). □

5 Computation via recurrences

The derived recurrence relationships, combined with the fact that we know the first row and column of $L(z)$ and $S(z)$, lead to a direct routine for computing all of their entries. In particular, we can infer each row/column from entries in the preceding row/column:

Corollary 5.1 *The rows and columns of* $L(z)$ *satisfy a three-term recurrence relationship:*

$$\begin{aligned} \mathrm{i}L(z)\mathbf{e}_{j-1} + (2j+1)CL(z)\mathbf{e}_j - \mathrm{i}L(z)\mathbf{e}_{j+1} &= (2j+1)F(z)\mathbf{e}_j, \\ -L(z)^\top \mathbf{e}_{k-1} - \mathrm{i}(2k+1)CL(z)^\top \mathbf{e}_k + L(z)^\top \mathbf{e}_{k+1} &= (2k+1)F(z)^\top \mathbf{e}_k. \end{aligned}$$

We consider computing the truncation up to total degree p:

$$L^p(z) := \begin{pmatrix} L_{00}(z) & \cdots & L_{0,p-1}(z) & L_{0p}(z) \\ L_{10}(z) & \cdots & L_{1,p-1}(z) & \\ \vdots & \iddots & & \\ L_{p0}(z) & & & \end{pmatrix}$$

Note that Corollary 5.1 allows us to deduce the $(j+1)$-th column ($(k+1)$-th row) of $L^p(z)$ from the jth column (kth row). We use this in Algorithm 1 to alternatively filling in a row and column until we have computed all the entries of $L^p(z)$. For the right-hand side, we denote

$$F^{(\alpha),p}(z) := \begin{pmatrix} F_{00}^{(\alpha)}(z) & \cdots & F_{0,p-1}^{(\alpha)}(z) & F_{0p}^{(\alpha)}(z) \\ F_{10}^{(\alpha)}(z) & \cdots & F_{1,p-1}^{(\alpha)}(z) & \\ \vdots & \iddots & & \\ F_{p0}^{(\alpha)}(z) & & & \end{pmatrix}.$$

Algorithm 1 Computing complex logarithmic integrals for Legendre polynomials up to total degree p, $L^p(z) \in \mathbb{C}^{(p+1)\times(p+1)}$

Construct $F^{(1),p}(z)$ defined in Theorem 3.3. Note that $M_0(z \pm 1), \ldots, M_{p+1}(z \pm 1)$ can be computed via the recurrence Corollary 2.5 in $O(p)$ operations.
Construct $F^{(2),p}(z)$ also using Theorem 3.3. Note that $L_0(z \pm \mathrm{i}), \ldots, L_{p+1}(z \pm \mathrm{i})$ can be computed via the recurrence in Corollary 2.5, whilst $C_2^{(-3/2)}(x), \ldots, C_{p+2}^{(-3/2)}(x)$ and $C_1^{(-1/2)}(y), \ldots, C_{p+1}^{(-1/2)}(y)$ can be computed via (2.6).
Compute the first row/column of $L^p(z)$ using the three-term recurrences as defined in Proposition 3.4 and Corollary 3.5.
for $\ell = 0, \ldots, \lfloor \frac{p}{2} \rfloor - 1$ **do**
Compute $L_{\ell+1,\ell+1}(z), \ldots, L_{p-\ell-1,\ell+1}(z)$ from $L^p(z)\mathbf{e}_\ell$ and $L^p(z)\mathbf{e}_{\ell-1}$ (for $\ell \geq 1$) using the first equation in Corollary 5.1.
Compute $L_{\ell+1,\ell+2}(z), \ldots, L_{\ell+1,p-\ell-1}(z)$ from $\mathbf{e}_\ell^\top L^p(z)$ and $\mathbf{e}_{\ell-1}^\top L^p(z)$ (for $\ell \geq 1$) using the second equation in Corollary 5.1.
end for
return $L^p(z)$.

Algorithm 1 is effective for computing Newtonian potentials of Legendre polynomials, by taking the real part of $L^p(z)$. As this algorithm computes the values exactly, the only source of error is round-off error; there are no errors introduced by numerical approximation. For low order p, the round-off error remains small when z is on or near the square, see Fig. 1. Here, we compare the computation to a high-precision computation via Julia's in-built `BigFloat` type using 1000 bits of precision (which due to the exactness of the formulæ will give more accurate approximations to the true result). This implies these techniques can be effectively combined with a more traditional quadrature-based approach for z bounded away from the square, where h-refinement is used to ensure only low p is needed to resolve the desired function. The round-off error grows exponentially with p at an increasing rate, the further away z is from

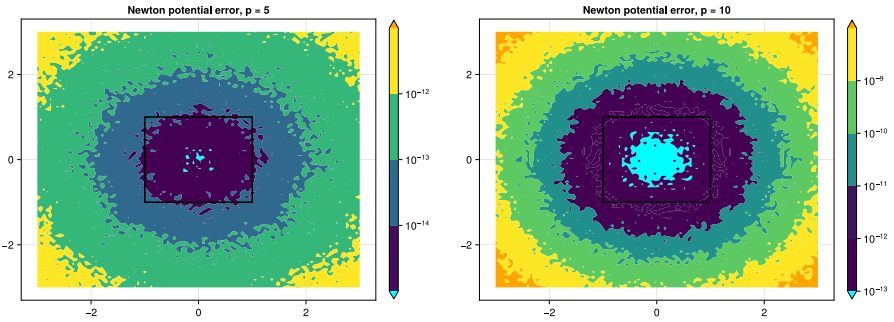


Fig. 1 The norm of the round-off absolute error in computing the Newtonian potential for $p = 5$ (left) and $p = 10$ (right) using Algorithm 1 to compute the $L^p(z)$ with double precision (`Float64`) arithmetic. We see that the approximation is accurate to about 12–15 digits within the square, remains accurate in a neighbourhood of the square but loses accuracy away from the square, where traditional quadrature becomes effective

the square, see Fig. 2. Using high-precision arithmetic is one approach to mitigating this issue, and we consider double-word arithmetic (cf. [20]) as implemented in the `Double64` type in [25]. Double-word arithmetic is based on using two traditional floating-point numbers to capture 106 bits of precision (compared with the 52 bits of standard double precision), and is very performant.

The recurrence is fast to solve as demonstrated in Fig. 3.[1] As a rough comparison, we compare the cost with that of calculating $\log(s+\mathrm{i}t)$ at $(p+1)^2$ points, which is a proxy for the cost of quadrature-based approaches which require kernel evaluation. Putting aside the round-off errors, the recurrence is faster than the cost of evaluating complex logarithms, as it has significantly fewer special function operations: most of the computation involves floating-point arithmetic operations. While with double-word precision as implemented in [25], the computational cost is greater than evaluating the complex logarithm, it is still in the same ballpark, and thus is potentially competitive for high-accuracy computations when compared to quadrature-based approaches.

We can similarly compute the Stieltjes integral $S(z)$ using the following recurrence on its rows and columns:

Corollary 5.2 *The rows and columns of $S(z)$ satisfy a three-term recurrence relationship: for $k, j > 0$, we have*

$$\begin{aligned} \mathrm{i}jS(z)\mathbf{e}_{j-1} + (2j+1)(B - zI)S(z)\mathbf{e}_j + \mathrm{i}(j+1)S(z)\mathbf{e}_{j+1} &= 0, \\ kS(z)^\top \mathbf{e}_{k-1} + (2k+1)(\mathrm{i}B - zI)S(z)^\top \mathbf{e}_k + (k+1)S(z)^\top \mathbf{e}_{k+1} &= 0. \end{aligned}$$

That is, we can compute

$$S^p(z) := \begin{pmatrix} S_{00}(z) & \cdots & S_{0,p-1}(z) & S_{0p}(z) \\ S_{10}(z) & \cdots & S_{1,p-1}(z) & \\ \vdots & \iddots & & \\ S_{p0}(z) & & & \end{pmatrix}$$

[1] The computations were performed on an M2 MacBook Air with a single thread.

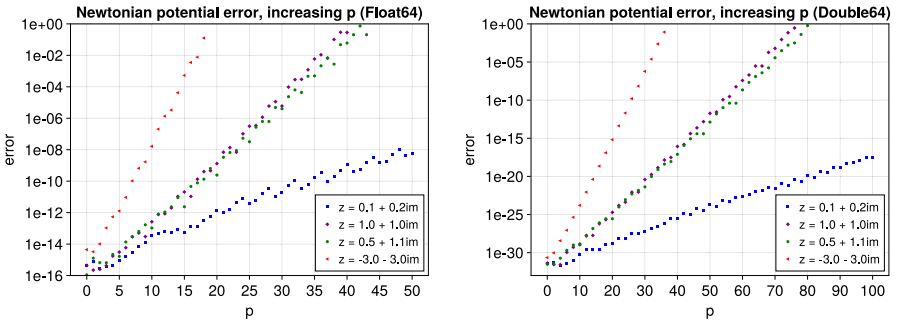


Fig. 2 The worst-case round-off absolute error in computing the Newtonian potential for increasing p using Algorithm 1 for four different points. The error grows significantly with p, but at a much slower rate for z near or within the square, which is the regime that traditional quadrature breaks down. Using double-word (`Double64`) arithmetic increases the accuracy and allows for roughly machine-precision accuracy inside the square up to around $p = 40$ and up to around $p = 20$ for an adjoining square sharing a corner

using Algorithm 2.

In Fig. 4, we see that the error for the Stieltjes integral (whose real and imaginary parts give the gradient of the Newtonian potential) behaves loosely similar to the Newtonian potential, though with larger errors, including within the square. Double-word arithmetic continues to be effective for evaluating the recurrence with high accuracy for moderately high p.

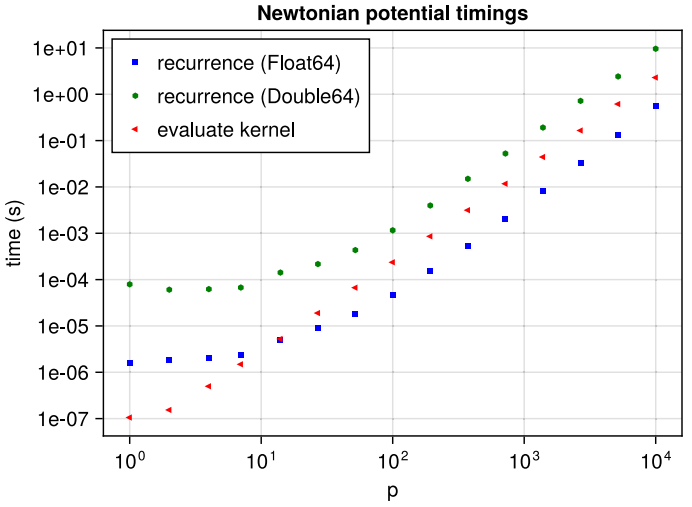


Fig. 3 Timing for computing the recurrence with double precision (`Float64`) and double-word precision (`Double64`), compared with the cost of evaluating $\log(s+it)$ at $(p+1)^2$ points (*evaluate kernel*), which we use as a proxy for quadrature-based approximations. The computational cost of Algorithm 1 is competitive, though the round-off error is currently a limiting factor

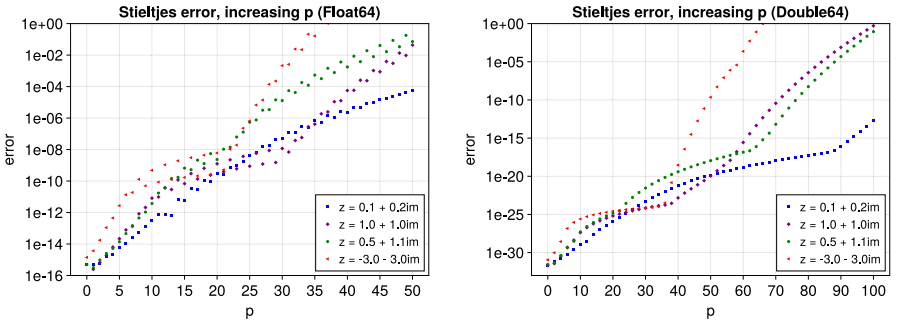


Fig. 4 The worst-case round-off absolute error in computing the Stieltjes integrals for increasing p using Algorithm 2 for four different points. The error behaviour is roughly similar to the Newtonian potential though with a kink phenomena dependent on the choice of z, likely due to the dependency on z in the Sylvester equation. Using double-word (`Double64`) arithmetic increases the accuracy and allows for roughly machine-precision accuracy inside the square up to around $p = 60$ and up to around $p = 40$ for an adjoining square sharing a corner

Algorithm 2 Computing the Stieltjes integrals for Legendre polynomials up to total degree p, $S^p(z) \in \mathbb{C}^{(p+1)\times(p+1)}$

Compute the first row/column of $S^p(z)$ using Proposition 3.2. Note that $M_0(z\pm 1), \dots, M_{p+1}(z\pm 1)$ and $M_0(-\mathrm{i}z\pm 1), \dots, M_{p+1}(-\mathrm{i}z\pm 1)$ can be computed via the recurrence Corollary 2.5 in $O(p)$ operations.
for $\ell = 0, \dots, \lfloor \frac{p}{2} \rfloor - 1$ **do**
Compute $S_{\ell+1,\ell+1}(z), \dots, S_{p-\ell-1,\ell+1}(z)$ from $S^p(z)\mathbf{e}_\ell$ and $S^p(z)\mathbf{e}_{\ell-1}$ (for $\ell \geq 1$) using the first equation in Corollary 5.2.
Compute $S_{\ell+1,\ell+2}(z), \dots, S_{\ell+1,p-\ell-1}(z)$ from $\mathbf{e}_\ell^\top S^p(z)$ and $\mathbf{e}_{\ell-1}^\top S^p(z)$ (for $\ell \geq 1$) using the second equation in Corollary 5.2.
end for
return $S^p(z)$.

6 Extensions to squares, square meshes, and rectangles

We now see how the results allow computation of Newtonian potentials on other squares and rectangles, which allow for computing on simple rectangular meshes. We tackle a general square of width and height h using an affine change of variables:

$$\begin{aligned}
&\int_a^{a+h}\int_b^{b+h} f(s,t)\log|z-(s+\mathrm{i}t)|\mathrm{d}s\mathrm{d}t \\
&\quad = \frac{h^2}{4}\int_{-1}^1\int_{-1}^1 f\left(a+\frac{(s+1)h}{2}, b+\frac{(t+1)h}{2}\right) \\
&\qquad\qquad \times \log\left|z-\left(a+\frac{(s+1)h}{2}+\mathrm{i}\left[b+\frac{(t+1)h}{2}\right]\right)\right|\mathrm{d}s\mathrm{d}t \\
&\quad = \frac{h^2}{4}\iint_\Omega f\left(a+\frac{(s+1)h}{2}, b+\frac{(t+1)h}{2}\right) \\
&\qquad\qquad \times \log\left|\frac{2}{h}(z-a-\mathrm{i}b)-1-\mathrm{i}-(s+\mathrm{i}t)\right|\mathrm{d}s\mathrm{d}t
\end{aligned}$$

$$-\frac{h^2\log(2/h)}{4}\iint_\Omega f\left(a+\frac{(s+1)h}{2},b+\frac{(t+1)h}{2}\right)\mathrm{d}s\mathrm{d}t \tag{6.1}$$

Thus, by expanding the mapped function $f\left(a+\frac{(s+1)h}{2},b+\frac{(t+1)h}{2}\right)$ in a Legendre expansion, we can compute its Newtonian potential from $L^p\left(\frac{2}{h}(z-a-\mathrm{i}b\right)-1-\mathrm{i})$, noting that the extra integral term only impacts the first entry by a simple modification.

Being able to compute Newtonian potentials over general squares allows us to compute Newtonian potentials over simple meshes, e.g., by dividing the unit square into smaller squares and summing over the Newtonian potential calculated over each of the smaller squares. As an example, consider recovering

$$u(x,y):=\mathrm{e}^{-60(x-1/5)^2-50(y-1/10)^2} \tag{6.2}$$

from its Laplacian, which equals

$$f(x,y):=8\mathrm{e}^{-60(x-1/5)^2-50(y-1/10)^2}(57-720x+1800x^2-250y+1250y^2). \tag{6.3}$$

Because u and f are extremely small outside the unit square, we can approximate

$$\begin{aligned}u(x,y)&=\frac{1}{2\pi}\iint_{\mathbb{R}^2}\log\|\boldsymbol{x}-\boldsymbol{t}\|f(\boldsymbol{t})\mathrm{d}\boldsymbol{t}\approx\frac{1}{2\pi}\iint_\Omega\log\|\boldsymbol{x}-\boldsymbol{t}\|f(\boldsymbol{t})\mathrm{d}\boldsymbol{t}\\&\approx\sum_{k,j=1}^{n}\int_{\frac{2(k-1)}{n}-1}^{\frac{2k}{n}-1}\int_{\frac{2(j-1)}{n}-1}^{\frac{2j}{n}-1}\log\|\boldsymbol{x}-\boldsymbol{t}\|f_p(\boldsymbol{t})\mathrm{d}\boldsymbol{t}\end{aligned}$$

where f_p is an approximation using mapped Legendre polynomials:

$$f_p\left(a+\frac{(s+1)h}{2},b+\frac{(t+1)h}{2}\right)=\sum_{k,j=0}^{p}c_{kj}^{(a,b,h,p)}P_k(s)P_j(t).$$

We compute $c_{kj}^{(a,b,h,p)}$ using the fast Legendre transform in [28], and the Newtonian potentials are then deducible in closed form from $L^p(z)$ using (6.1). In Fig. 5, we see how the approach converges for different choices of polynomial order p. Using purely `Float64` arithmetic, the approach is still fairly accurate despite the instability, and we can recover u to around 10 digits, using 100×100 sub-squares and $p=6$ on each square. If we use `DoubleF64` arithmetic to compute $L^p(z)$ (whilst still using `Float64` arithmetic for the Legendre transforms), we can recover u to roughly machine precision using 100×100 sub-squares and $p=9$ on each square.

Rectangles with differing height and weight, or more generally, parallelograms, require some modification of the techniques. When we rescale (or possibly rotate) the rectangle, we can reduce the computation to one over a canonical rectangle:

$$L_{kj}^{\gamma}(z):=\int_{-\gamma}^{\gamma}\int_{-1}^{1}P_k\left(\frac{s}{\gamma}\right))P_j(t)\log(z-(s+\mathrm{i}t))\mathrm{d}t\mathrm{d}s,$$

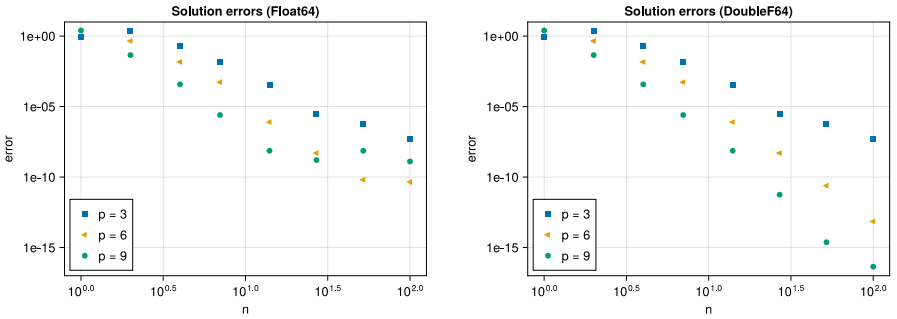


Fig. 5 The worst-case absolute error in computing the Newtonian potential (divided by 2π) of f defined in (6.3) compared with u defined in (6.2) measured at a 4×4 grid of points, by dividing the unit square into n^2 squares of size $2/(n+1) \times 2/(n+1)$. In each sub-square, the Newtonian potential is computed by finding the Legendre coefficients via [28] using `Float64` arithmetic, where the Newtonian potentials of the Legendre polynomials $L^p(z)$ are computed via Algorithm 1. In the left figure, we use `Float64` arithmetic to compute $L^p(z)$ and see that there is a limit to the accuracy, though it is accurate to up to 10 digits. In the right figure, we use `DoubleF64` to compute $L^p(z)$ and achieve roughly machine-precision accuracy

where $\gamma > 0$. The construction in Theorem 3.3 can be readily generalised to these integrals, see Appendix. One thus arrives at two Sylvester equations similar to Lemma 4.2, now of the form

$$\begin{aligned} zL^\gamma(z) &= \gamma A L^\gamma(z) + \mathrm{i} L^\gamma(z) B^\top + F^{(1),\gamma}(z), \\ zL^\gamma(z) &= \gamma B L^\gamma(z) + \mathrm{i} L^\gamma(z) A^\top + F^{(2),\gamma}(z), \end{aligned}$$

where $F^{(1),\gamma}/F^{(2),\gamma}$ are simple scalings and small rank perturbations of $F^{(1)}/F^{(2)}$. The techniques we developed for computing $L^p(z)$ will then apply to this modified setting.

7 Conclusion

We have shown that complex logarithmic and Stieltjes integrals of Legendre polynomials, whose real and imaginary parts contain the Newtonian potential and its gradient, have a displacement structure. This leads to an extremely fast approach for evaluating Newtonian potentials on and near the square, though there are issues with round-off error for large orders. There are a number of directions for which this result may be useful in real-world computations. One may combine the recurrence-based approach for evaluating Newtonian potentials on and near the square with quadrature-based techniques away from the square, though the precise switch-over point requires further investigation. When combined with h-refinement, this would result in an effective, fast, and parallelisable approach for computing Newtonian potentials. Initial experiments on a square mesh have shown this is an effective approach for computing Newtonian potentials, and with the use of high-precision arithmetic can lead to computations achieving roughly machine-precision accuracy.

An arguably more interesting direction would be utilisation of the displacement structure for faster and more reliable computations, including both near and far from the square, by overcoming the instability in the recurrence. The exact source of the instability is not yet clear, though in the one-dimensional case, the recurrences are known to have a maximal solution given by orthogonal polynomials which satisfy the same recurrence relationship; hence, the phenomena in the 2D case likely have a similar cause. In the 1D case, there are classical ways of finding the minimal solution of three-term recurrences, including Miller's algorithm and (F.W.J.) Olver's algorithm, cf. [11, Section 3.6], but it currently remains unclear how these techniques can be extended to 2D.

The structure of our system is very similar to that of the Poisson equation as discretised in [12], which was solved efficiently with the Alternative Direction Implicit method, which may provide a means for stable computation away from the square. There is also a recent algorithm which avoids inversion [6] which may also apply to our setting. A hiccup in utilising this result is that whilst our matrices have displacement structure, the right-hand side of the Sylvester equation does not decay rapidly for z in the square, hence truncation will result in large errors: we need to work with the recurrence in an infinite-dimensional way. Another alternative is to use mixed precision computation, which was introduced for computing an operator arising in fractional calculus which had displacement structure in [24]. These potential approaches for stable computation would be significantly slower than the unstable direct recurrence, however.

Displacement structure also has implications in terms of utilising the matrices in further computation. It is known that we can perform fast matrix–vector operations [19], and we further have precise bounds in the decay of the singular values [8, 9], and there has been recent progress adapting the results to complex matrices [10]. This latter result tells us that, in principle, we do not need to compute the full matrix, but rather can approximate it by a low-rank matrix, which may lead to much more efficient evaluation of Newtonian potentials.

In terms of generalisation of the result, we suspect that complex logarithmic and Stieltjes integrals applied to orthogonal polynomials on triangles and other simple two-dimensional geometries have simple recurrence relationships. It seems likely there is also displacement structure in higher dimensions; however, the derivation based on complex analysis and Stieltjes integrals is unlikely to generalise. Recent results on computing power law integrals of orthogonal polynomials on intervals [17] and balls [18] may facilitate generalisation of displacement structure results, in particular to power law integrals on squares and cubes. An extension to triangles would facilitate application of the results to more general geometries, as in the scheme introduced in [4].

Appendix. Modified complex logarithmic integrals for rectangles and parallelograms

Here, we consider the derivation of recurrence relationships for the integral

$$L_{kj}^{\gamma}(z) := \int_{-\gamma}^{\gamma} \int_{-1}^{1} P_k\left(\frac{s}{\gamma}\right) P_j(t) \log(z - (s + \mathrm{i}t)) \mathrm{d}s \mathrm{d}t,$$

following the proof of Theorem 3.3. Mimicking the first recurrence in Theorem 3.3, we first define

$$\begin{aligned} L_k^\gamma(z) &:= \int_{-\gamma}^{\gamma} P_k\left(\frac{s}{\gamma}\right)\log(z-s)\mathrm{d}s = \gamma\int_{-1}^{1} P_k(t)\log(\gamma[z/\gamma - t])\mathrm{d}t \\ &= \gamma L_k\left(\frac{z}{\gamma}\right) + 2\gamma\log\gamma\delta_k, \end{aligned}$$

which satisfies the three-term recurrence

$$\begin{aligned} zL_k^\gamma(z) &= \gamma^2\frac{z}{\gamma}L_k\left(\frac{z}{\gamma}\right) + 2z\gamma\log\gamma\delta_k \\ &= \gamma^2\frac{k-1}{2k+1}L_{k-1}\left(\frac{z}{\gamma}\right) + \gamma^2\frac{k+2}{2k+1}L_{k+1}\left(\frac{z}{\gamma}\right) + \gamma^2\lambda_k(z/\gamma) + 2z\gamma\log\gamma\delta_k \\ &= \gamma\frac{k-1}{2k+1}L_{k-1}^\gamma(z) + \gamma\frac{k+2}{2k+1}L_{k+1}^\gamma(z) + \gamma^2\lambda_k(z/\gamma) + 2z\gamma\log\gamma\delta_k \end{aligned}$$

We can then write $L_{kj}^\gamma(z) = \int_{-1}^{1} P_j(t)L_k^\gamma(z-\mathrm{i}t)\,\mathrm{d}t$ so that

$$\begin{aligned} zL_{kj}^\gamma(z) &= \gamma\int_{-1}^{1} P_j(t)(z-\mathrm{i}t)L_k^\gamma(z-\mathrm{i}t)\,\mathrm{d}t + \mathrm{i}\int_{-1}^{1} tP_j(t)L_k(z-\mathrm{i}t)\,\mathrm{d}t \\ &\quad + 4z\log\gamma\delta_k\delta_j \\ &= \gamma\frac{k-1}{2k+1}L_{k-1,j}^\gamma(z) + \gamma\frac{k+2}{2k+1}L_{k+1,j}^\gamma(z) \\ &\quad + \mathrm{i}\frac{j}{2j+1}L_{k,j-1}^\gamma(z) + \mathrm{i}\frac{j+1}{2j+1}L_{k,j+1}^\gamma(z) \\ &\quad + \gamma\int_{-1}^{1} P_j(t)\lambda_k\left(\frac{z-\mathrm{i}t}{\gamma}\right)\mathrm{d}t + 4\gamma\log\gamma\delta_k\left(z\delta_j - \frac{\mathrm{i}}{3}\delta_{j1}\right). \end{aligned}$$

We can simplify the remaining integral term using

$$\begin{aligned} &\int_{-1}^{1} P_j(t)\left(\frac{z-\mathrm{i}t}{\gamma}\pm 1\right)\log\left(\frac{z-\mathrm{i}t}{\gamma}\pm 1\right)\mathrm{d}t \\ &\quad = \frac{1}{\gamma}\int_{-1}^{1} P_j(t)(z\pm\gamma-\mathrm{i}t)\log(z\pm\gamma-\mathrm{i}t)\mathrm{d}t - \frac{\log\gamma}{\gamma}\int_{-1}^{1} P_j(t)(z\pm\gamma-\mathrm{i}t)\mathrm{d}t \\ &\quad = \frac{\mathrm{i}}{\gamma}\frac{M_{j+1}(z\pm\gamma) - M_{j-1}(z\pm\gamma)}{2j+1} \\ &\qquad + \frac{\mu_j(z\pm\gamma)}{\gamma} - \frac{2\log\gamma}{\gamma}\begin{cases} z\pm\gamma & j=0 \\ -\frac{\mathrm{i}}{3} & j=1 \\ 0 & \text{otherwise} \end{cases}, \end{aligned}$$

where we used (3.2). Thus, we get

$$\gamma\int_{-1}^{1} P_j(t)\lambda_0\left(\frac{z-\mathrm{i}t}{\gamma}\right)\mathrm{d}t$$
$$=\mathrm{i}\frac{M_{j+1}(z-\gamma)+M_{j+1}(z+\gamma)-M_{j-1}(z-\gamma)-M_{j-1}(z+\gamma)}{2j+1}$$
$$+\mu_j(z-\gamma)+\mu_j(z+\gamma)-2\log\gamma\begin{cases}2z & j=0\\ -\frac{2\mathrm{i}}{3} & j=1\\ 0 & \text{otherwise}\end{cases},$$
$$\gamma\int_{-1}^{1} P_j(t)\lambda_1\left(\frac{z-\mathrm{i}t}{\gamma}\right)\mathrm{d}t=-\frac{4\gamma\delta_j}{3},$$
$$\gamma\int_{-1}^{1} P_j(t)\lambda_k\left(\frac{z-\mathrm{i}t}{\gamma}\right)\mathrm{d}t=0,\qquad (k>1).$$

For the second recurrence, we first write: $L^{\gamma}_{kj}(z)=\int_{-\gamma}^{\gamma}P_k(s/\gamma)M_j(z-s)\mathrm{d}s$ so that

$$zL^{\gamma}_{kj}(z)=\int_{-\gamma}^{\gamma}P_k(s)(z-s)M_j(z-s)\mathrm{d}s+\gamma\int_{-\gamma}^{\gamma}\frac{s}{\gamma}P_k\left(\frac{s}{\gamma}\right)M_j(z-s)\mathrm{d}s$$
$$=\mathrm{i}\frac{j-1}{2j+1}L^{\gamma}_{k,j-1}(z)+\mathrm{i}\frac{j+2}{2j+1}L^{\gamma}_{k,j+1}(z)$$
$$+\gamma\frac{k}{2k+1}L^{\gamma}_{k-1,j}(z)+\gamma\frac{k+1}{2k+1}L^{\gamma}_{k+1,j}(z)$$
$$+\int_{-\gamma}^{\gamma}P_k\left(\frac{s}{\gamma}\right)\mu_j(z-s)\mathrm{d}s.$$

We have

$$\int_{-\gamma}^{\gamma}P_k(s/\gamma)(z-s)\log(z-s)\mathrm{d}s=\gamma\frac{L^{\gamma}_{k+1}(z)-L^{\gamma}_{k-1}(z)}{2k+1}+\gamma^2\lambda_k\left(\frac{z}{\gamma}\right)+2z\gamma\log\gamma\delta_k.$$

Noting that

$$\int_{x}^{\gamma}P_k\left(\frac{s}{\gamma}\right)\mathrm{d}s=\gamma C^{(-1/2)}_{k+1}\left(\frac{s}{\gamma}\right)+\gamma\delta_k,$$

if $-\gamma<x<\gamma$ we have

$$\int_{-\gamma}^{\gamma}P_k\left(\frac{s}{\gamma}\right)\begin{cases}x-s & x<s\\ 0 & \text{otherwise}\end{cases}\mathrm{d}s=\int_{x}^{\gamma}P_k(s/\gamma)(x-s)\mathrm{d}s$$
$$=\gamma^2\frac{x}{\gamma}C^{(-1/2)}_{k+1}\left(\frac{x}{\gamma}\right)-\frac{\gamma^2 k}{2k+1}C^{(-1/2)}_{k}\left(\frac{x}{\gamma}\right)-\gamma^2\frac{k+1}{2k+1}C^{(-1/2)}_{k+2}\left(\frac{x}{\gamma}\right)$$

$$
\begin{aligned}
&+\gamma x\delta_k - \gamma^2\frac{\delta_{k1}}{3}\\
&= -\gamma^2\frac{C_{k+2}^{(-3/2)}(x/\gamma)}{3} + \gamma x\delta_k - \gamma^2\frac{\delta_{k1}}{3}.
\end{aligned}
$$

Thus,

$$
\begin{aligned}
\int_{-\gamma}^{\gamma} P_k\left(\frac{s}{\gamma}\right)\mu_0(z-s)\mathrm{d}s &= \gamma\,\frac{L_{k+1}^{\gamma}(z-\mathrm{i}) + L_{k+1}^{\gamma}(z+\mathrm{i}) - L_{k-1}^{\gamma}(z-\mathrm{i}) - L_{k-1}^{\gamma}(z+\mathrm{i})}{2k+1}\\
&+4z\gamma\log\gamma\delta_k\\
&+2\mathrm{i}\pi y\begin{cases}
-\gamma^2\frac{C_{k+2}^{(-3/2)}\left(\frac{x}{\gamma}\right)}{3} + \gamma x\delta_k - \gamma^2\frac{\delta_{k1}}{3} & -\gamma < x < \gamma \text{ and } -1 \le y < 1\\
-2\gamma x\delta_k + 2\gamma^2\delta_{k1}/3 & x \le -\gamma \text{ and } -1 \le y < 1\\
0 & \text{otherwise.}
\end{cases}
\end{aligned}
$$

For $j \neq 0$, we have

$$
\begin{aligned}
&\int_{-\gamma}^{\gamma} P_k\left(\frac{s}{\gamma}\right)\mu_j(z-s)\mathrm{d}s\\
&\quad= -\frac{4\mathrm{i}}{3}\delta_k\delta_{j1} - 2\mathrm{i}\pi C_{j+1}^{(-1/2)}(y)\int_{-\gamma}^{\gamma} P_k\left(\frac{s}{\gamma}\right)\begin{cases} x-s & x < s \text{ and } -1 < y < 1\\ 0 & \text{otherwise}\end{cases}\mathrm{d}s\\
&\quad= -\frac{4\mathrm{i}}{3}\delta_k\delta_{j1}\\
&\qquad+2\mathrm{i}\pi C_{j+1}^{(-1/2)}(y)\begin{cases}
\frac{\gamma^2 C_{k+2}^{(-3/2)}\left(\frac{x}{\gamma}\right)}{3} - \gamma x\delta_k + \frac{\gamma^2\delta_{k1}}{3} & -\gamma < x < \gamma \text{ and } -1 < y < 1\\
-2x\delta_k + \frac{2\delta_{k1}}{3} & -1 < y < 1 \text{ and } x \le -\gamma\\
0 & \text{otherwise}
\end{cases}.
\end{aligned}
$$

Acknowledgements I would like to thank Thomas Anderson (Rice) who suggested this problem and discussed the work in depth. I also thank the very helpful comments from the anonymous referees.

Author contribution Not applicable.

Funding This work was supported by an Engineering and Physical Sciences Research Council (EPSRC) grant (EP/T022132/1).

Data availability Not applicable.

Materials availability Not applicable.

Code availability The implementation is available from [22].

Declarations

Ethical approval This article does not contain any studies involving animals. This article does not contain any studies involving human participants.

Consent for publication Not applicable.

Conflict of interest The author declares no competing interests.

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/.

References

1. Linteberg, L.a., Tornberg, A.-K.: Adaptive quadrature by expansion for layer potential evaluation in two dimensions. SIAM J. Sci. Comput. **40**, A1225–A1249 (2018)
2. Alpert, B.K., Rokhlin, V.: A fast algorithm for the evaluation of Legendre expansions. SIAM J. Sci. Stat. Comput. **12**, 158–179 (1991)
3. Anderson, T.G., Bonnet, M., Faria, L.M., Pérez-Arancibia, C.: Fast, high-order numerical evaluation of volume potentials via polynomial density interpolation. J. Comput. Phys. **511**, 113091 (2024)
4. Anderson, T.G., Zhu, H., Veerapaneni, S.: A fast, high-order scheme for evaluating volume potentials on complex 2D geometries via area-to-line integral conversion and domain mappings. J. Comput. Phys. **472**, 111688 (2023)
5. Atkinson, K.: The numerical evaluation of particular solutions for Poisson's equation. IMA J. Numer. Anal. **5**, 319–338 (1985)
6. Ballew, C., Trogdon, T., Wilber, H.: The Akhiezer iteration and an inverse-free solver for Sylvester matrix equations. arXiv:2503.17496 (2025)
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**, A427–A451 (2014)
8. Beckermann, B., Townsend, A.: On the singular values of matrices with displacement structure. SIAM J. Matrix Anal. Appl. **38**, 1227–1248 (2017)
9. Beckermann, B., Townsend, A.: Bounds on the singular values of matrices with displacement structure. SIAM Rev. **61**, 319–344 (2019)
10. Clouâtre, R., Klippenstein, B., Slevinsky, R.M.: Lifting Sylvester equations: singular value decay for non-normal coefficients. Integr. Eqn. Oper. Theory **97**, 7 (2025)
11. Olver, F.W.J., Olde Daalhuis, A.B., Lozier, D.W., Schneider, B.I., Boisvert, R.F., Clark, C.W., Miller, B.R., Saunders, B.V., Cohl, H.S., McClain, M.A., (eds.): NIST digital library of mathematical functions. https://dlmf.nist.gov/, Release 1.2.4 of 2025-03-15. https://dlmf.nist.gov/.
12. Fortunato, D., Townsend, A.: Fast Poisson solvers for spectral methods. IMA J. Numer. Anal. **40**, 1994–2018 (2020)
13. Gautschi, W.: Minimal solutions of three-term recurrence relations and orthogonal polynomials. Math. Comput. **36**, 547–554 (1981)
14. Gautschi, W., Wimp, J.: Computing the Hilbert transform of a Jacobi weight function. BIT Numer. Math. **27**, 203–215 (1987)
15. Greengard, L., Lee, J.-Y.: A direct adaptive Poisson solver of arbitrary order accuracy. J. Comput. Phys. **125**, 415–424 (1996)
16. Greengard, L., O'Neil, M., Rachh, M., Vico, F.: Fast multipole methods for the evaluation of layer potentials with locally-corrected quadratures. J. Comput. Phys. X **10**, 100092 (2021)
17. Gutleb, T., Carrillo, J., Olver, S.: Computing equilibrium measures with power law kernels. Math. Comput. **91**, 2247–2281 (2022)
18. Gutleb, T.S., Carrillo, J.A., Olver, S.: Computation of power law equilibrium measures on balls of arbitrary dimension. Constr. Approx. **58**, 75–120 (2023)
19. Heinig, G., Rost, K.: Algebraic Methods for Toeplitz-Like Matrices and Operators, vol. 13, Birkhäuser Basel (1984)

20. Joldes, M., Muller, J.-M., Popescu, V.: Tight and rigorous error bounds for basic building blocks of double-word arithmetic. ACM Trans. Math. Softw. (TOMS) **44**, 1–27 (2017)
21. Keiner, J.: Fast polynomial transforms. Logos Verlag Berlin GmbH (2011)
22. Olver, S.: MultivariateSingularIntegrals.jl. v0.1, GitHub (2025)
23. Olver, S., Slevinsky, R.M., Townsend, A.: Fast algorithms using orthogonal polynomials. Acta Numer **29**, 573–699 (2020)
24. Pu, T., Fasondini, M.: The numerical solution of fractional integral equations via orthogonal polynomials in fractional powers. Adv. Comput. Math. **49**, 7 (2023)
25. Sarnoff, J.: DoubleFloats.jl. v1.2, GitHub (2025)
26. Shen, Z., Serkh, K.: Rapid evaluation of Newtonian potentials on planar domains. SIAM J. Sci. Comput. **46**, A609–A628 (2024)
27. Slevinsky, R.M., Olver, S.: A fast and well-conditioned spectral method for singular integral equations. J. Comput. Phys. **332**, 290–315 (2017)
28. Townsend, A., Webb, M., Olver, S.: Fast polynomial transforms based on Toeplitz and Hankel matrices. Math. Comput. **87**, 1913–1934 (2018)
29. Vico, F., Greengard, L., Ferrando, M.: Fast convolution with free-space Green's functions. J. Comput. Phys. **323**, 191–203 (2016)

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