Implementation and study of boundary integral operators related to PDE:s in the plane

Erik Andersson

Master's thesis
2023:E27

Lund University

Faculty of Engineering
Centre for Mathematical Sciences
Numerical Analysis

Centrum Scientiarum Mathematicarum

Abstract

The method of solving boundary value problems of partial differential equations numerically by first reformulating the problem as a boundary integral equation has many advantages over other methods, but also some unique difficulties. Some of these difficulties stem from problems in evaluating singular or nearly singular integral operators, and solving these difficulties is an active field of research. Known results are summarized, and an accessible program package is developed, using underlying Gauss–Legendre quadrature and product integration, which can be applied to boundary value problems with smooth boundaries. The program package is available on Github at the link `https://github.com/erikandersson98/BIE-CELib`. Different methods of implementing integral operators related to the Laplace and Helmholtz equations are compared with regards to accuracy and convergence rate, both when used in boundary value problems, and when applied to theoretical identities. The methods are based on product integration, as well as on global and local regularization. To conclude, recommended implementations based on the results are given, as well as possible directions to expand the package.

1 Introduction

When solving boundary value problems (BVPs) of linear partial differential equations (PDEs) with piecewise constant coefficients, one can often represent the solution in the form of a convolution of a layer density against the fundamental solution of the PDE over a boundary, yielding a boundary integral equation (BIE). Discretizing this BIE is a method of solving boundary value problems, which has some advantages over other methods, like the finite element method. First, since only the boundary needs to be discretized the dimension of the problem is reduced to the dimension of the boundary. Second, boundary integral equation methods can yield very high accuracy results and high rates of convergence compared to other methods. But there are also problems associated with methods of discretizing BIEs, like difficulties stemming from a singular integral kernel, and difficulties evaluating the field close to, or on, the boundary. Solving these difficulties is very much an active field of research. State-of-the-art methods include, for example, the hedgehog method [19], related to quadrature by expansion [18], and modified versions of the trapezoidal rule and Gauss–Legendre quadrature [10]. Also worth mentioning is the method from the recent article [21], which utilises Stokes theorem on manifolds to solve problems for 3-dimensional geometries. In this paper, the method of product integration with an underlying Gauss–Legendre quadrature [14] is used to solve the problem of evaluation close to the boundary.

The paper is about implementation of integral operators that come up when solving the Laplace equation and the Helmholtz equation in the plane using BIE methods. The main purpose of the paper is to summarize known results and to develop an easily accessible program package for MATLAB, to be used by interested parties. Secondly, the purpose is also to test different implementations of integral operators arising in the solution of the Laplace and Helmholtz equations, specifically different implementations on the boundary. The package can be found on GitHub at `https://github.com/erikandersson98/BIE-CELib`.

The paper is organized as follows. The current section, Section 1, serves as

an introduction, with Section 1.1 being an introductory example to exemplify the general structure of the types of numerical solution this paper is concerned with, as well as to give context to the theory in Sections 2-6. In Section 2 some integral operators which arise in the solution of the Laplace equation in the plane are introduced and defined. Section 3 describes the method of product integration, and Section 4 introduces the methods global and local regularization, which can serve as an alternative to product integration for target points on a smooth boundary. Two different methods of local regularization are considered depending on which variable interpolation is carried out in. In Section 5 methods of handling the branch cut of the logarithm are described with one of the methods seemingly being new for this paper. Section 6 introduces operators which arise in the solution of the Helmholtz equation, as well as describes a kernel splitting method of handling these operators using the theory of product integration and regularization from Sections 3 and 4. In Section 7 the tests which were conducted to compare the implementations are described, and Section 8 shows the results of these tests. Lastly, Section 9 ends with discussion of the results as well as conclusions.

1.1 Introductory example

We start with an example showing how one might arrive at an integral equation from a BVP, and outlining the general procedure of solving the BVP. First we apply Nyström discretization in Section 1.1.1, and then deal with a singular integrand in Section 1.1.2, which will turn out particularly easy in this first example. Lastly, field evaluation is mentioned in Section 1.1.3.

Consider the Interior Dirichlet problem for the Laplace equation in the plane, that is

$$\Delta U(\boldsymbol{x}) = 0\,, \qquad \boldsymbol{x} \in D\,, \tag{1}$$

with D being some interior domain with boundary Γ. On Γ we have Dirichlet boundary conditions

$$\lim_{D \ni \boldsymbol{x}' \to \boldsymbol{x}} U(\boldsymbol{x}') = g(\boldsymbol{x}), \qquad \boldsymbol{x} \in \Gamma\,. \tag{2}$$

It is often possible to represent the solution $U(\boldsymbol{x})$ in the form

$$U(\boldsymbol{x}) = \frac{1}{\pi} \int_\Gamma \frac{\mu(\boldsymbol{y}(t))(\boldsymbol{n}(t) \cdot (\boldsymbol{y}(t) - \boldsymbol{x}))\,\mathrm{d}t}{|\boldsymbol{y}(t) - \boldsymbol{x}|^2}, \qquad \boldsymbol{x} \in D\,, \tag{3}$$

where $\mathrm{d}t$ is an element of arc length, $\boldsymbol{n}$ is the outward unit normal on Γ, and $\boldsymbol{y}(t)$ is a point on Γ. We make an ansatz that the solution U can, in fact, be represented as in (3). The function $\mu(\boldsymbol{x})$ is called the layer density, which is what we want to solve for as it then determines $U(\boldsymbol{x})$. By way of identification of $\mathbb{R}^2$ with the complex plane, $\mathbb{C}$, $z = \boldsymbol{x}$, $n = \boldsymbol{n}$ as well as $\mathrm{d}\tau = \mathrm{i}n\mathrm{d}t$ with the complex integration variable $\tau = \boldsymbol{y}(t)$, we get

$$\begin{aligned} U(\boldsymbol{x}) &= \frac{1}{\pi} \int_\Gamma \frac{\mu(\boldsymbol{y}(t))(\boldsymbol{n}(t) \cdot (\boldsymbol{y}(t) - \boldsymbol{x}))\,\mathrm{d}t}{|\boldsymbol{y}(t) - \boldsymbol{x}|^2} = \frac{1}{\pi} \int_\Gamma \frac{\mu(\tau)\Re(\bar{n}(\tau - z))\,\mathrm{d}t}{|\tau - z|^2} \\ &= \frac{1}{\pi} \int_\Gamma \mu(\tau)\Re\left(\frac{\mathrm{d}\tau}{\mathrm{i}(\tau - z)}\right) = \frac{1}{\pi} \int_\Gamma \mu(\tau)\Im\left(\frac{\mathrm{d}\tau}{\tau - z}\right) = U(z), \end{aligned} \tag{4}$$

where the notation $\Re$ is used for the real part and $\Im$ for the imaginary part. From (2), (4) and jump relations [9, Eqs. (2.5) and (2.6)] we get the Fredholm integral of the second kind

$$\mu(z) + \frac{1}{\pi} \int_\Gamma \mu(\tau) \Im \left(\frac{\mathrm{d}\tau}{\tau - z} \right) = g(z) \,, \qquad z \in \Gamma \,. \tag{5}$$

The jump relations describe how the values of the layer density on the boundary relate to the values off the boundary.

To move further we specify the boundary Γ by a parametrization $z(p)$, $-\pi \leq p < \pi$. In this example the following parametrization holds

$$z(p) = (1 + a \cos 5p) e^{\mathrm{i}p} \,, \tag{6}$$

with $a = 0.3$, see Figure 1. Since $\mathrm{d}\tau = \tau'(p)\mathrm{d}p$, we can write (5) as

$$\mu(z) + \frac{1}{\pi} \int_{-\pi}^{\pi} \mu(p) \Im \left(\frac{\tau'(p)}{\tau(p) - z} \right) \mathrm{d}p = g(z) \,, \qquad z \in \Gamma \,. \tag{7}$$

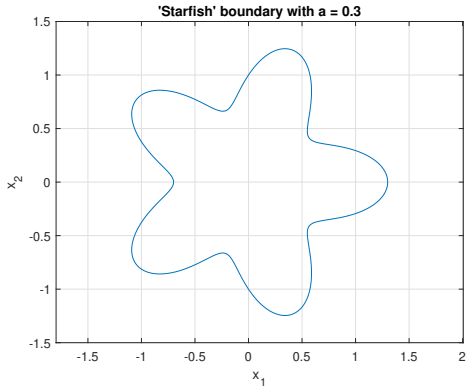


Figure 1: Starfish boundary (6), with $a = 0.3$.

1.1.1 Nyström discretization

Having arrived at the BIE (5), Nyström discretization leads to a system of equations which can be solved numerically. First, composite Gauss–Legendre quadrature is applied to the integral in (5). Divide the boundary into N panels, Λ_i, $i = 1, 2, \ldots N$, each spanning a parameter interval $[a_i, b_i)$ of length Δp_i in the parameter p which will each be integrated by 16-point Gauss–Legendre quadrature.

$$\mu(z) + \frac{1}{\pi} \sum_{i=1}^{N} \int_{a_i}^{b_i} \mu(p) \Im \left(\frac{\tau'(p)}{\tau(p) - z} \right) \mathrm{d}p = g(z) \,. \tag{8}$$

We introduce the canonical Gauss–Legendre nodes s_j and weights w_j, $j = 1, 2, \ldots, 16$. They can be used to integrate the interval $[-1, 1]$, but here we have intervals of length Δp_i. With a change of variables

$$s(p) = \frac{p}{\Delta p_i} + a_i \,, \tag{9}$$

we get,

$$\int_{a_i}^{b_i} \mu(p)\Im\left(\frac{\tau'(p)}{\tau(p)-z}\right)\,\mathrm{d}p = \int_{-1}^{1} \mu(s)\Im\left(\frac{\tau'(s)}{\tau(s)-z}\right)\Delta p_i\,\mathrm{d}s$$
$$\approx \Delta p_i \sum_{j=1}^{16} w_j\mu(s_j)\Im\left(\frac{\tau'(s_j)}{\tau(s_j)-z}\right) = \sum_{j=1}^{16} w_{i,j}\mu(p_{i,j})\Im\left(\frac{\tau'(p_{i,j})}{\tau(p_{i,j})-z}\right), \qquad (10)$$

where $w_{i,j} = \Delta p_i w_j$ are the rescaled Gauss–Legendre weights, and $p_{i,j} = \Delta p_i(s_j - a_i)$ are the rescaled Gauss–Legendre nodes. Throughout the paper, functions $f(x)$ of some variable x will be written $f(y)$ for some integration variable y. This is shorthand which should be interpreted as $f(y) = f(x(y))$. See (8) where $\mu(p)$ is shorthand for $\mu(\tau(p))$, or (10) where $\tau(s)$ is shorthand for $\tau(p(s))$. In short, we switch integration variables, but don't want to introduce intermediary functions or write out $f(x(y))$ every time, so we let the relation be implicit and only write the dependence on the integration variable, $f(y)$. Introducing the shorthand $f(p_{i,j}) = f_{i,j}$, as well, we get

$$\mu(z) + \frac{1}{\pi}\sum_{i=1}^{N}\sum_{j=1}^{16} w_{i,j}\mu_{i,j}\Im\left(\frac{\tau'_{i,j}}{\tau_{i,j}-z}\right) \approx g(z)\,. \qquad (11)$$

We arrive at Equation (11), where we have discretized the integral on the left-hand side of (5). As the next step we set this equation to hold at every quadrature node and get a large system of $16N$ equations. Relabelling p_i such that the index $i = 1, 2, \ldots 16N$ goes through every node of every panel in order, and using the notation $f(p_i) = f_i$, we get

$$\mu_i + \frac{1}{\pi}\sum_{j=1}^{16N} w_j\mu_j\Im\left(\frac{\tau'_j}{\tau_j-\tau_i}\right) = g_i\,, \qquad i = 1, 2, \ldots 16N\,. \qquad (12)$$

Or written in matrix form,

$$(\boldsymbol{I} - \boldsymbol{K})\boldsymbol{\mu} = \boldsymbol{g}\,, \qquad (13)$$

which we can solve for $\boldsymbol{\mu}$, a vector of $\mu(z)$ evaluated at the nodes. The reasoning for the $-\boldsymbol{K}$ will be clear later, as this is the negative Neumann–Poincaré operator. The matrix $-\boldsymbol{K}$ has elements $\frac{1}{\pi}w_j\Im\left(\frac{\tau'_j}{\tau_j-\tau_i}\right)$. To make sense of the diagonal elements we need to consider limit values, which we do in Section 1.1.2. Equations (12) and (13) are the Nyström discretization of the original integral equation (5).

1.1.2 Limit values

Some trouble occurs when $i = j$ in (12) since there is a singularity in $\frac{\tau'_j}{\tau_j-\tau_i}$, but the imaginary part is actually finite. Considering the real parameter p variable:

$$\frac{\tau'(p_j)}{\tau(p_j)-\tau(p)} = -\frac{|\tau'_j|^2}{(p-p_j)|\tau'_j + O(p-p_j)|^2} - \frac{\tau'_j\overline{\tau''_j}}{2|\tau'_j + O(p-p_j)|^2} + O(p-p_j)\,. \qquad (14)$$

The first term on the right hand side of (14) has no imaginary part, and the third term goes to zero as p goes to p_j, so we only need to consider the second term.

$$\lim_{p\to p_j} \Im\left(\frac{\tau_j'}{\tau_j - \tau(p)}\right) = \lim_{p\to p_j} \Im\left(-\frac{\tau_j'\overline{\tau_j''}}{2|\tau_j' + O(p-p_j)|^2}\right) = \frac{\tau_j''}{2\tau_j'}\,, \tag{15}$$

and this limit value is used in the diagonal elements in the matrix $\boldsymbol{K}$.

1.1.3 Field evaluation

In the field evaluation we evaluate the function $U(\boldsymbol{x}) = U(z)$ from (3), for $z \in D$. After solving (13) for the density $\boldsymbol{\mu}$, field evaluation is in theory easy. We only need to integrate

$$U(z) = \frac{1}{\pi}\int_\Gamma \mu(\tau)\Im\left(\frac{\mathrm{d}\tau}{\tau - z}\right) \approx -\boldsymbol{K}\boldsymbol{\mu}\,, \qquad z \in D\,. \tag{16}$$

But in practice there are difficulties in the evaluation for z close to the boundary Γ, because of the pole in the integrand. We did not have difficulties with the Nyström discretization stemming from close evaluation on the boundary, since the integrand in (8) is smooth, at least on a smooth boundary, and its singularity is removable, replaced with a limit value in the diagonal of $\boldsymbol{K}$. It might not be clear at first glance that the integrand is smooth on a smooth boundary, but the fact that numerical results are good even without close evaluation considerations indicates this, as well as, for example, [13, Sec. 4.3]. We will put a pin in this and move on to some more general concepts and return later to field evaluation.

2 Operators and potentials, Laplace equation

In this section we define integral operators which arise in the solution of the Laplace equation. Since this is not the first paper written on the subject of boundary integral equations, most of the more common integral operators have already been named, and it is good if naming conventions are respected. Though in this case, it is difficult to find consistent naming conventions, and there can be differences in sign or a multiplicative constant between different authors. Integral operators from potential theory are also called layer potentials in the literature, and we will use the terms layer potential and integral operator synonymously for those integral operators.

Starting with the fundamental solution to Laplace's equation in two dimensions

$$E(\boldsymbol{x},\boldsymbol{y}) = \frac{1}{2\pi}\log\frac{1}{|\boldsymbol{x}-\boldsymbol{y}|}\,, \tag{17}$$

we define the single-layer potential,

$$Sf(\boldsymbol{x}) = \int_\Gamma E(\boldsymbol{x},\boldsymbol{y})f(\boldsymbol{y})\,\mathrm{d}|\boldsymbol{y}|\,, \tag{18}$$

as well as the double-layer potential [3, Eq. (2.2)]

$$Df(\boldsymbol{x}) = \int_\Gamma \frac{\partial}{\partial \boldsymbol{n}_y}E(\boldsymbol{x},\boldsymbol{y})f(\boldsymbol{y})\,\mathrm{d}|\boldsymbol{y}|\,, \tag{19}$$

and the adjoint of the double-layer potential

$$D^{\mathrm{A}} f(\boldsymbol{x}) = \int_\Gamma \frac{\partial}{\partial \boldsymbol{n}_x} E(\boldsymbol{x}, \boldsymbol{y}) f(\boldsymbol{y}) \,\mathrm{d}|\boldsymbol{y}|\,, \tag{20}$$

where the directional derivatives

$$\frac{\partial}{\partial \boldsymbol{n}_y} = \boldsymbol{n}_y \cdot \nabla_y\,, \qquad \frac{\partial}{\partial \boldsymbol{n}_x} = \boldsymbol{n}_x \cdot \nabla_x\,, \tag{21}$$

have been introduced. We have

$$\nabla_y E(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{2\pi} \frac{\boldsymbol{x} - \boldsymbol{y}}{|\boldsymbol{x} - \boldsymbol{y}|^2}\,, \tag{22}$$

so the double-layer potential and it's adjoint are

$$Df(\boldsymbol{x}) = \int_\Gamma D(\boldsymbol{x}, \boldsymbol{y}) f(\boldsymbol{y}) \,\mathrm{d}|\boldsymbol{y}| = \frac{1}{2\pi} \int_\Gamma \frac{\boldsymbol{n}_y \cdot (\boldsymbol{x} - \boldsymbol{y})}{|\boldsymbol{x} - \boldsymbol{y}|^2} f(\boldsymbol{y}) \,\mathrm{d}|\boldsymbol{y}|\,, \tag{23}$$

$$D^{\mathrm{A}} f(\boldsymbol{x}) = \int_\Gamma D^{\mathrm{A}}(\boldsymbol{x}, \boldsymbol{y}) f(\boldsymbol{y}) \,\mathrm{d}|\boldsymbol{y}| = -\frac{1}{2\pi} \int_\Gamma \frac{\boldsymbol{n}_x \cdot (\boldsymbol{x} - \boldsymbol{y})}{|\boldsymbol{x} - \boldsymbol{y}|^2} f(\boldsymbol{y}) \,\mathrm{d}|\boldsymbol{y}|\,, \tag{24}$$

where $D(\boldsymbol{x}, \boldsymbol{y})$ is the kernel of the double-layer potential, and $D^{\mathrm{A}}(\boldsymbol{x}, \boldsymbol{y})$ is the kernel of its adjoint.

There is also a similar integral operator to the double-layer potential, namely, the Neumann–Poincaré operator [3, Eq. (2.5)],

$$Kf(\boldsymbol{x}) = 2 \int_\Gamma D(\boldsymbol{x}, \boldsymbol{y}) f(\boldsymbol{y}) \,\mathrm{d}|\boldsymbol{y}| = \frac{1}{\pi} \int_\Gamma \frac{\boldsymbol{n}_y \cdot (\boldsymbol{x} - \boldsymbol{y})}{|\boldsymbol{x} - \boldsymbol{y}|^2} f(\boldsymbol{y}) \,\mathrm{d}|\boldsymbol{y}|\,. \tag{25}$$

The adjoint to the Neumann–Poincaré operator is then obtained by replacing $D(\boldsymbol{x}, \boldsymbol{y})$ with $D^{\mathrm{A}}(\boldsymbol{x}, \boldsymbol{y})$ in (25).

To give some examples of works which use different definitions, in [9], the single-layer and double-layer potentials are of opposite sign, and in [6] the kernel of the Neumann–Poincaré operator seems to be $-D^{\mathrm{A}}(\boldsymbol{x}, \boldsymbol{y})$. Though the definitions given above seem most common in the literature and are used in this paper.

Of use when discretizing layer potentials are the following three integral operators: The Cauchy singular operator

$$M_{\mathrm{C}} f(z) = \frac{1}{\pi \mathrm{i}} \int_\Gamma \frac{f(\tau) \,\mathrm{d}\tau}{\tau - z}\,, \tag{26}$$

the logarithmic operator

$$M_{\mathrm{L}} f(z) = \int_\Gamma f(\tau) \log|\tau - z| \,\mathrm{d}|\tau|\,, \tag{27}$$

and the hypersingular operator

$$M_{\mathrm{H}} f(z) = \frac{1}{\pi \mathrm{i}} \int_\Gamma \frac{f(\tau) \,\mathrm{d}\tau}{(\tau - z)^2}\,. \tag{28}$$

Note that these integral operators are defined on functions of a complex variable.

In order to make sense of the singularities in the integrands of the integral operators (26) and (28), in the context of solving BIEs arising from BVPs for all problems considered in this paper, M_{C} should be taken in the Cauchy principal value sense and M_{H} should be taken as the Hadamard finite part. There exist obvious connections between M_{C} and the double-layer potential and Neuman–Poincaré operators, as well as between M_{L} and the single-layer potential. For real f, the real part of M_{C} is the negative Neumann–Poincaré operator. Only a multiplicative constant separates the double-layer potential from the Nemann–Poincaré operator, and only a multiplicative constant separates the single-layer potential from M_{L}.

As in Section 1, by identification of $\mathbb{R}^2$ with $\mathbb{C}$, $\tau = \boldsymbol{y}$ and $z = \boldsymbol{x}$, the single-layer and double-layer potentials as well as the Neumann–Poincaré operator can be written as integrals in a complex variable. Using $\mathrm{d}|\boldsymbol{y}| = \mathrm{d}|\tau| = \frac{1}{\mathrm{i}\boldsymbol{n}_y}\mathrm{d}\tau$,

$$Sf(z) = -\frac{1}{2\pi}\int_\Gamma f(\tau)\log|z-\tau|\,\mathrm{d}|\tau|\,, \tag{29}$$

$$Df(z) = \frac{1}{2\pi}\int_\Gamma f(\tau)\Re\left(\frac{n_\tau(\bar{z}-\bar{\tau})}{|z-\tau|^2}\,\mathrm{d}|\tau|\right) = \frac{1}{2\pi}\int_\Gamma f(\tau)\Re\left(\frac{\mathrm{d}\tau}{\mathrm{i}(z-\tau)}\right)\,, \tag{30}$$

$$D^{\mathrm{A}}f(z) = \frac{1}{2\pi}\int_\Gamma f(\tau)\Re\left(\frac{n_z(\bar{z}-\bar{\tau})}{|z-\tau|^2}\,\mathrm{d}|\tau|\right) = \frac{1}{2\pi}\int_\Gamma f(\tau)\Re\left(\frac{n_z\bar{n}_\tau\,\mathrm{d}\tau}{\mathrm{i}(z-\tau)}\right)\,, \tag{31}$$

$$Kf(z) = \frac{1}{\pi}\int_\Gamma f(\tau)\Re\left(\frac{\mathrm{d}\tau}{\mathrm{i}(z-\tau)}\right)\,. \tag{32}$$

3 Product integration

3.1 General

In this section we summarize the product integration method introduced in [14] and expanded in [20, 11]. Starting in Section 3.1 with the concept and general notation. In Section 3.2 selected product integration weights are introduced and calculated, and in Section 3.3 the operators M_{C}, M_{L}, and M_{H} are rewritten to a form in which the product integration weights are easily applied. Lastly, in Section 3.4 we return to field evaluation of the introductory example mentioned in Section 1.1.3.

In the evaluation of a layer potential close to a kernel singularity, there could be problems with discretization, as mentioned in Section 1.1.3. The problem was not present for $z \in \Gamma$ in (13) since the singularity was removable in this case and the integrand was smooth if the boundary was smooth. Sometimes we will have the same problem in the discretization on the boundary, say, with any of the three operators M_{C}, M_{L} and M_{H}. One way to solve this problem of close evaluation is to split the integration over the panels and use product integration on the panels close to the singularity, a method popularized in [14]. For an integral operator with kernel $G(z,\tau)$,

$$\int_\Gamma G(z,\tau)f(\tau)\,\mathrm{d}\tau = \int_{\Gamma_{\mathrm{f}}} G(z,\tau)f(\tau)\,\mathrm{d}\tau + \sum_{i\in I_{\mathrm{c}}}\int_{\Lambda_i} G(z,\tau)f(\tau)\,\mathrm{d}\tau\,, \tag{33}$$

where I_{c} is an index set of panels close to the target point, z, and Γ_{f} is the part of the boundary far from the target point. This split of the boundary can be seen for the starfish boundary and an example point in Figure 2.

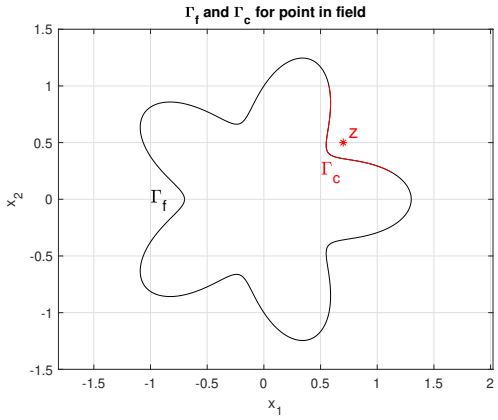


Figure 2: Split of boundary into one part close to target point, Γ_{c}, and one far from target point, Γ_{f}. Target point is $z = 0.7 + 0.5\mathrm{i}$, seen in red. Starfish boundary used with 30 panels.

Two schemes for determining which panels are considered far from the target point and close to the target point are explained in Section 7.4. After a split according to (33), for each of these integral terms close to the singularity we transform the panel by scaling, rotation and translation, so that it starts in -1 and ends in 1. Calling this transformed panel Λ_i^{t}, we end up with a sum of terms like

$$\int_{\Lambda_i^{\mathrm{t}}} G(z,\tau) f(\tau)\,\mathrm{d}\tau\,. \tag{34}$$

And here we interpolate $f(\tau)$ as a polynomial on the transformed panel. If we interpolate to a polynomial of order m, $f(\tau) \approx \sum_{i=0}^{m} \alpha_i \tau^i$, and integrate each term exactly, we end up with

$$\int_{\Lambda_i^{\mathrm{t}}} G(z,\tau) f(\tau)\,\mathrm{d}\tau \approx \sum_{k=0}^{m} \alpha_k \int_{\Lambda_i^{\mathrm{t}}} G(z,\tau)\tau^k\,\mathrm{d}\tau = \sum_{k=0}^{m} \alpha_k G_k(z)\,, \tag{35}$$

where $G_k(z)$ denotes the exact value of the integral of the kernel $G(z,\tau)$ multiplied with a monomial τ^k.

For the part of the boundary far from the singularity, Γ_{f}, regular Gauss–Legendre quadrature can be used. If we use 16-point Gauss–Legendre quadrature, we could interpolate f as a 15-degree polynomial using the quadrature nodes, τ_i, $i = 1, 2, \ldots 16$.

We interpolate by solving the system of equations

$$\begin{aligned} &\sum_{k=0}^{15} \tau_i^k \alpha_k = f(\tau_i)\,,\ i = 1, 2, \ldots, 16\,, \\ &\iff \boldsymbol{V}\boldsymbol{\alpha} = \boldsymbol{f}\,, \end{aligned} \tag{36}$$

where $\boldsymbol{V}$ is a Vandermonde matrix. If f is a layer density, then we might not know the value of f at the nodes, in fact, that is exactly what we are trying to approximate in the first step of solving the BIE, the Nyström discretization.

But in the product integration

$$\int_{\Lambda_i^{\mathrm{t}}} G(z,\tau)f(\tau)\,\mathrm{d}\tau \approx \sum_{k=0}^{m} \alpha_k G_k(z) = \boldsymbol{G}(z)^T\boldsymbol{\alpha} = \boldsymbol{G}(z)^T\boldsymbol{V}^{-1}\boldsymbol{f}\,. \tag{37}$$

We can factor out $\boldsymbol{f}$ and include the remaining factors in the matrix resulting from Nyström discretization. In other words, if product integration is used for the Nyström discretization, the result is a modification to the matrix so that it includes submatrices of the form $\boldsymbol{G}(z)\boldsymbol{V}^{-1}$, corresponding to the close evaluation. If the close evaluation is used in field evaluation, then f is known at the quadrature nodes and (37) can be used immediately for a target point z.

3.2 Product integration weights on transformed panels

If we want to use product integration in the close evaluation for the operators M_{C}, M_{L} and M_{H}, we need the following quantities, which we call product integration weights,

$$p_k(z) = \int_\Lambda \frac{\tau^k\,\mathrm{d}\tau}{\tau - z}\,, \tag{38}$$

$$q_k(z) = \int_\Lambda \tau^k \log(\tau - z)\,\mathrm{d}\tau\,, \tag{39}$$

$$\tilde{p}_k(x) = \int_{-1}^{1} \frac{s^k\,\mathrm{d}s}{s - x}\,, \tag{40}$$

$$\tilde{q}_k(x) = \int_{-1}^{1} s^k \log|s - x|\,\mathrm{d}s\,, \tag{41}$$

$$r_k(z) = \int_\Lambda \frac{\tau^k\,\mathrm{d}\tau}{(\tau - z)^2}\,, \tag{42}$$

where $k = 0, 1 \ldots, 15$ and Λ is a transformed panel, starting in -1 and ending in 1. The variants (40) and (41) are taken with the parameter integration variable s, with x being the parameter value corresponding to the target point z, that is, $\tau(x) = z$ where $\tau(s)$ is a parametrization of the whole boundary Γ, with $\tau(s)$ for $s \in [-1, 1]$ being points on the panel Λ.

If $z \in \Lambda$ or $\tau(x) \in \Lambda$ then (38) and (40) are to be taken in the Cauchy principal value sense, and (42) is to be taken as the Hadamard finite part, see Section 2. If the boundary is smooth we could imagine pushing the boundary to move in a circle segment, C_ϵ, of radius ϵ around the target point z, see Figure 3. With Λ_ϵ being the panel Λ with the segment inside the circle of radius ϵ around z removed, and assuming that the branch cut of $\log(\tau - z)$ isn't crossed as τ traverses the curve $\Lambda_\epsilon + C_\epsilon$, z being fixed,

$$\int_{\Lambda_\epsilon + C_\epsilon} \frac{\mathrm{d}\tau}{\tau - z} = \log(1 - z) - \log(-1 - z)\,, \tag{43}$$

$$\int_{\Lambda_\epsilon} \frac{\mathrm{d}\tau}{\tau - z} = \log(1 - z) - \log(-1 - z) - \int_{C_\epsilon} \frac{\mathrm{d}\tau}{\tau - z}\,. \tag{44}$$

The principal value of (38) for $z \in \Lambda$ is the limit of the integral along Λ_ϵ in (44) as $\epsilon \to 0$, so on a smooth boundary and assuming the branch cut of $\log(\tau - z)$

isn't crossed as τ traverses $\Lambda_\epsilon + C_\epsilon$, we get

$$p_0(z) = \begin{cases} \log(1-z) - \log(-1-z)\,, & \text{if } z \notin \Gamma\,, \\ \log(1-z) - \log(-1-z) \mp \pi\mathrm{i}\,, & \text{if } z \in \Gamma\,, \end{cases} \tag{45}$$

where the sign of $\pi\mathrm{i}$ is dependent on the orientation of the circle segment C_ϵ. Note that in one of these cases of circle orientation, the assumption that the branch cut of $\log(\tau - z)$ isn't crossed will not hold, so we need to compensate this case by adding or subtracting $2\pi\mathrm{i}$. What is seemingly two answers depending on the sign of $\pi\mathrm{i}$ in (45), is actually the same answer because of this compensation term.

Once we have $p_0(z)$, we can get all other $p_k(z)$ by recursion,

$$p_{k+1}(z) = -zp_k(z) + \frac{1-(-1)^{k+1}}{k+1}\,, \qquad k = 1, 2, \ldots, m-1\,. \tag{46}$$

We also get $q_k(z)$ by integration by parts,

$$q_k(z) = \frac{-p_{k+1}(z) + \log(1-z) - (-1)^{k+1}\log(-1-z)}{k+1}\,, \tag{47}$$

So far we haven't committed to a branch of the logarithm, keeping the general notation log, but if k is even then we now get a $q_k(z)$ which varies depending on which branch of the logarithm is chosen, because the log terms are added instead of subtracted. So do we now need to commit to a branch, and if so which one? For now we can state that the integral (39) comes from the logarithmic operator (27) and the rewrite

$$\begin{aligned} &\int_\Lambda f(\tau)\log|\tau - z|\,\mathrm{d}|\tau| = \int_\Lambda f(\tau)\Re\left(\log(\tau - z)\frac{\mathrm{d}\tau}{\mathrm{i}n_\tau}\right) \\ &= \Re\left(\int_\Lambda \frac{\Re(f(\tau))}{\mathrm{i}n_\tau}\log(\tau - z)\,\mathrm{d}\tau\right) + \mathrm{i}\Re\left(\int_\Lambda \frac{\Im(f(\tau))}{\mathrm{i}n_\tau}\log(\tau - z)\,\mathrm{d}\tau\right). \end{aligned} \tag{48}$$

In (48) the functions to be interpolated are $\frac{\Re(f(\tau))}{\mathrm{i}n_\tau}$ and $\frac{\Im(f(\tau))}{\mathrm{i}n_\tau}$. The logarithmic operator has $\log|\tau - z|$ in the integrand and so obviously shouldn't depend on the choice of branch, but the problem arises as a result of the inexact interpolation. Let us assume f is real, and we have

$$\begin{aligned} &\int_\Lambda f(\tau)\log|\tau - z|\,\mathrm{d}|\tau| = \Re\left(\int_\Lambda \frac{f(\tau)}{\mathrm{i}n_\tau}\log(\tau - z)\,\mathrm{d}\tau\right) \\ &\approx \Re\left(\int_\Lambda (\sum_{k=0}^{m}\alpha_k\tau^k)\log(\tau - z)\,\mathrm{d}\tau\right). \end{aligned} \tag{49}$$

For (49) we get

$$\begin{aligned} &\Re\left(\int_\Lambda (\sum_{k=0}^{m}\alpha_k\tau^k)\log(\tau - z)\,\mathrm{d}\tau\right) \\ &= \int_\Lambda \log|\tau - z|\Re\left((\sum_{k=0}^{m}\alpha_k\tau^k)\,\mathrm{d}\tau\right) - \int_\Lambda \arg(\tau - z)\Im\left((\sum_{k=0}^{m}\alpha_k\tau^k)\,\mathrm{d}\tau\right), \end{aligned} \tag{50}$$

where the function $\arg(z)$ gives the argument of z in some interval of length 2π, but the interval should be thought of as not yet fixed. The approximation sign in (49) is one which we hope holds, we want to think about if it actually does. In other words, we want to think about if a small interpolation error leads to a small error in the real part of the integral. If the interpolation $f(\tau)/(in_\tau) \approx \sum_{k=0}^m \alpha_k \tau^k$ is good then

$$\int_\Lambda \log|\tau - z| \Re\left((\sum_{k=0}^m \alpha_k \tau^k)\, \mathrm{d}\tau \right) \approx \int_\Lambda f(\tau) \log|\tau - z|\, \mathrm{d}|\tau| \,, \tag{51}$$

since $\sum_{k=0}^m \alpha_k \tau^k$ approximates $f(\tau)/(in_\tau)$, and we have $\mathrm{d}\tau/(in_\tau) = \mathrm{d}|\tau|$ which is real. The right-hand side of (51) is the logarithmic operator we wanted to approximate, so the approximation (49) is good if the second term in (50) is small. We can find an upper bound

$$\left| \int_\Lambda \arg(\tau - z) \Im\left((\sum_{k=0}^m \alpha_k \tau^k)\, \mathrm{d}\tau \right) \right| \leq M \int_\Lambda \left| \Im\left((\sum_{k=0}^m \alpha_k \tau^k)\, \mathrm{d}\tau \right) \right| = M\epsilon_{\text{int}} \,, \tag{52}$$

where $M = \max_{\tau\in\Lambda}(|\arg(\tau - z)|)$ and ϵ_{int} is the value of the integral. If the interpolation is good then ϵ_{int} should be small, since, as mentioned after (51), we have $\mathrm{d}\tau/(in_\tau) = \mathrm{d}|\tau|$ which is real. If the branch is chosen to give small values of the argument the second term in (50) should be ≈ 0, since M would be relatively small. But say that we choose a very non-standard branch of the logarithm that gives argument values in $[10^{10}, 10^{10} + 2\pi)$, then the second term in (50) might be large even for a good interpolation.

The fact that the interpolation used in the application of the product integration method to the logarithmic operator leads to an approximation which changes depending on the choice of branch is at the very least an issue worth mentioning, though not mentioned in, for example, [20], where no branch is given. Though, as was just shown, the choice of branch shouldn't matter much as long as it's chosen to give small argument values.

The issue of interpolation leading to a product integration weight with a value depending on the branch of the logarithm is not present in (41), for which we get

$$\tilde{q}_k(x) = \frac{-\tilde{p}_{k+1}(x) + \log|1 - x| - (-1)^{k+1} \log|-1 - x|}{k+1} \,, \tag{53}$$

$$\tilde{p}_0(x) = \log|1 - x| - \log|-1 - x| \,, \tag{54}$$

and the recursion for $\tilde{p}_k(x)$ is the same as for $p_k(z)$, given by (46). The Cauchy principal value for $\tilde{p}_k(x)$ when $z \in \Lambda$ is the same as the value when $z \notin \Lambda$, where z is the target point, so we don't separate those two cases here. Note that the weights $\tilde{p}_k(x)$ and $\tilde{q}_k(x)$ in (40) and (41) only apply if the target point z is on the boundary, since we need an x which has $\tau(x) = z$ for $\tau(s)$ being a parametrization of the boundary Γ.

Lastly, we have by integration by parts

$$r_0(z) = -\frac{1}{1 - z} + \frac{(-1)^{k-1}}{-1 - z} \,, \tag{55}$$

$$r_k(z) = k p_{k-1}(z) - \frac{1}{1 - z} + \frac{(-1)^{k-1}}{-1 - z} \,, \qquad \text{if } k \geq 1 \,. \tag{56}$$

3.3 M_{C}, M_{L}, and M_{H}

We have the product integration weights on transformed panels Λ^{t}. Now we want to apply the results from the previous section to the Cauchy singular operator M_{C}, the logarithmic operator M_{L} and the hypersingular operator M_{H}.

The operators are first split according to (33), keeping the notation general as to allow use in both field evaluation and Nyström discretization.

$$M_{\mathrm{C}} f(z) = \frac{1}{\pi \mathrm{i}} \int_{\Gamma_{\mathrm{f}}} \frac{f(\tau)\,\mathrm{d}\tau}{\tau - z} + \sum_{i \in I_{\mathrm{c}}} \frac{1}{\pi \mathrm{i}} \int_{\Lambda_i} \frac{f(\tau)\,\mathrm{d}\tau}{\tau - z}\,, \tag{57}$$

$$M_{\mathrm{L}} f(z) = \int_{\Gamma_{\mathrm{f}}} f(\tau) \log|\tau - z|\,\mathrm{d}|\tau| + \sum_{i \in I_{\mathrm{c}}} \int_{\Lambda_i} f(\tau) \log|\tau - z|\,\mathrm{d}|\tau|\,, \tag{58}$$

$$M_{\mathrm{H}} f(z) = \frac{1}{\pi \mathrm{i}} \int_{\Gamma_{\mathrm{f}}} \frac{f(\tau)\,\mathrm{d}\tau}{(\tau - z)^2} + \sum_{i \in I_{\mathrm{c}}} \frac{1}{\pi \mathrm{i}} \int_{\Lambda_i} \frac{f(\tau)\,\mathrm{d}\tau}{(\tau - z)^2}\,. \tag{59}$$

If Λ_i is a panel spanning $[a_i, b_i)$ in parameter and $\tau(p)$ is a parametrization of the boundary, the endpoints of the panel are $z_a = \tau(a_i)$ and $z_b = \tau(b_i)$. The transformation

$$y = \frac{\tau - (z_b + z_a)/2}{(z_b - z_a)/2} = T(\tau) \Longleftrightarrow \tau(y) = \frac{(z_b - z_a)y}{2} + \frac{z_b + z_a}{2}, \tag{60}$$

brings Λ_i to the transformed panel Λ_i^{t} which has start point -1 and endpoint 1. We are now going to change integration variables from τ to $y = T(\tau)$ in the right-most terms in (57), (58) and (59). For (57), (58) we will also change the right-most terms to a form in which the weights (40) and (41) can be applied.

Starting with M_{C} we have

$$\int_{\Lambda_i} \frac{f(\tau)\,\mathrm{d}\tau}{\tau - z} = \int_{\Lambda_i^{\mathrm{t}}} \frac{f(y)\,\mathrm{d}y}{y - \hat{z}}\,, \tag{61}$$

where $\hat{z} = T(z)$. Alternatively, if $z \in \Gamma$ and $z = \tau(p_z)$,

$$\int_{\Lambda_i} \frac{f(\tau)\,\mathrm{d}\tau}{\tau - z} = \int_{\Lambda_i} \frac{f(\tau)\,\mathrm{d}\tau}{\tau - z} - \int_{a_i}^{b_i} \frac{f(p)\,\mathrm{d}p}{p - p_z} + \int_{-1}^{1} \frac{f(s)\,\mathrm{d}s}{s - s_z}, \tag{62}$$

with s as in (9) and $s_z = s(p_z)$. The first two terms of the right-hand side in (62) together make a smooth expression if the boundary is smooth, which can be integrated by Gauss–Legendre quadrature.

Now for M_{L}. Assume for simplicity that f is real, then

$$\int_{\Lambda_i} f(\tau) \log|\tau - z|\,\mathrm{d}|\tau| = \int_{\Lambda_i} \log\left|\frac{z_b - z_a}{2}\right| \rho(\tau)\,\mathrm{d}|\tau| + \Re\left(\frac{z_b - z_a}{2} \int_{\Lambda_i^{\mathrm{t}}} \frac{f(y)}{\mathrm{i} n_y} \log(y - \hat{z})\,\mathrm{d}y\right), \tag{63}$$

where $n_y = n_\tau|_{\tau = \tau(y)}$. If f is not real then it can be split into a real and imaginary part, to which the results can be applied. Alternatively, if $z \in \Gamma$, and

$z = \tau(p_z)$,

$$\int_{\Lambda_i} f(\tau) \log|\tau - z| \,\mathrm{d}|\tau| = \int_{a_i}^{b_i} f(p) \log\left|\frac{b_i - a_i}{2} \frac{\tau(p) - z}{p - p_z}\right| |\tau'(p)| \,\mathrm{d}p + \int_{-1}^{1} f(s) \log|(s - s_z)||\tau'(s)| \,\mathrm{d}s\,, \tag{64}$$

with s as in (9) and $s_z = s(p_z)$.

Lastly, for M_{H} we have

$$\int_{\Lambda_i} \frac{f(\tau)\,\mathrm{d}\tau}{(\tau - z)^2} = \frac{2}{b_i - a_i} \int_{\Lambda_i^{\mathrm{t}}} \frac{f(y)\,\mathrm{d}y}{(y - \hat{z})^2}\,. \tag{65}$$

Then we interpolate according to (37), replacing $G_k(z)$ with one of the product integration weights from Section 3.2.

3.4 Field evaluation of introductory example

Now that we have described the process of product integration we can solve the problem of evaluating (16). We will do so by splitting the integral according to (33)

$$U(z) = \frac{1}{\pi} \int_{\Gamma_{\mathrm{f}}} \mu(\tau) \Im\left(\frac{\mathrm{d}\tau}{\tau - z}\right) + \frac{1}{\pi} \sum_{i \in I_{\mathrm{c}}} \int_{\Lambda_i} \mu(\tau) \Im\left(\frac{\mathrm{d}\tau}{\tau - z}\right). \tag{66}$$

And the integrals over panels close to the target point are calculated by product integration. In practice we can first evaluate $U(z)$ at every point by using standard Gauss–Legendre quadrature, and then go through and add compensation terms for every panel which the target point is close to, which makes the algorithm of evaluating many field points simpler:

$$U(z) = \frac{1}{\pi} \sum_{j=1}^{16N} w_j \mu_j \Im\left(\frac{\tau_j'}{\tau_j - z}\right) + \frac{1}{\pi} \sum_{i \in I_{\mathrm{c}}} \sum_{j=1}^{16} \Im(w_{i,j}^{\mathrm{cmpC}}(z)) \mu_{i,j}\,, \tag{67}$$

where $w_{i,j}^{\mathrm{cmpC}}(z)$ are compensation weights for the Cauchy singular operator multiplied by $\pi\mathrm{i}$. The compensation weights consist of one part related to the product integration, and one part to correct for the inappropriate parts of the first sum in (67). Looking back at (26), there is a coefficient $1/(\pi\mathrm{i})$ in the kernel of the M_{C} operator. We let the compensation weights compensate for the operator multiplied by $\pi\mathrm{i}$ and divide by $\pi\mathrm{i}$ when using the weights. The π is outside the sums in the right-most term in (67), and the i has been implicitly divided, as we take the imaginary part of the weight, while the real part of the Cauchy operator is the negative Neumann–Poincaré operator. This same method of first using regular Gauss–Legendre quadrature and then adding compensation terms related to the product integration can also be used in Nyström discretization for the operators like M_{L}, where close evaluation is a problem on the boundary as well.

4 Global and Local Regularization

If the target point is on the boundary, then for the operators M_{C} and M_{H} there is another option of using global or local regularization [11]. In global regularization, we replace the integrand with one smooth term and one term which we can integrate exactly. In local regularization we only do this rewrite for the part of the boundary close to the target point. For M_{C}, $z \in \Gamma$ where Γ is a smooth boundary, and assuming the branch cut of $\log(\tau - z)$ isn't crossed as τ traverses Γ_{c}, deformed to go around z in a circle-segment of radius ϵ,

$$M_{\mathrm{C}} f(z) = \frac{1}{\pi \mathrm{i}} \int_{\Gamma} \frac{f(\tau) - f(z)}{\tau - z} \, \mathrm{d}\tau + f(z) \,, \tag{68}$$

$$\begin{aligned} M_{\mathrm{C}} f(z) =& \frac{1}{\pi \mathrm{i}} \int_{\Gamma_{\mathrm{f}}} \frac{f(\tau) \, \mathrm{d}\tau}{\tau - z} + \frac{f(z)}{\pi \mathrm{i}} (\log(z_b - z) - \log(z_a - z) \mp \pi \mathrm{i}) \\ &+ \frac{1}{\pi \mathrm{i}} \int_{\Gamma_{\mathrm{c}}} \frac{f(\tau) - f(z)}{\tau - z} \, \mathrm{d}\tau \,, \end{aligned} \tag{69}$$

where Γ_{c} is the part of the boundary close to the target point z, starting in z_a and ending in z_b. Again, the sign of $\pi \mathrm{i}$ depends on which way around z we imagine pushing the boundary. When discretizing, the diagonal terms will contain $f'(z)$. We will get this derivative by interpolating f as a polynomial of degree m, and then differentiating the polynomial is easily done by matrix multiplication with a matrix $\boldsymbol{D}$ which is zero except on the first superdiagonal which is $(1, 2 \ldots, m-1)$. One could imagine two different ways of interpolating f, one by interpolating $f(z)$ directly in the complex plane, and another by rewriting $f'(z) = f'(p)/\tau'(p)$, with a parametrization $\tau(p)$ and interpolating f in the parameter variable.

Considering both of these possible interpolation methods for the regularization is an idea which seems new, though logical since a similar separation has been considered for the product integration. Compare (39) and (41) for example.

For M_{H}, $z \in \Gamma$, where Γ is a smooth boundary we will just consider local regularization,

$$\begin{aligned} M_{\mathrm{H}} f(z) =& \frac{1}{\pi \mathrm{i}} \int_{\Gamma_{\mathrm{f}}} \frac{f(\tau) \, \mathrm{d}\tau}{(\tau - z)^2} - \frac{f(z)}{\pi \mathrm{i}} \left(\frac{1}{z_b - z} - \frac{1}{z_a - z} \right) \\ &+ f'(z) (\log(z_b - z) - \log(z_a - z) \mp \pi \mathrm{i}) \\ &+ \frac{1}{\pi \mathrm{i}} \int_{\Gamma_{\mathrm{c}}} \frac{f(\tau) - f(z) - f'(z)(\tau - z)}{(\tau - z)^2} \, \mathrm{d}\tau \,. \end{aligned} \tag{70}$$

Here we have also the second derivative in the diagonal terms of the discretization, which we will get by interpolating and twice differentiating the resulting polynomial. Once again one could imagine interpolating $f(z)$ and differentiating or using $f'(z) = f'(p)/\tau'(p)$ meaning $f''(z) = f''(p)/\tau'(p) - f'(p)/\tau'(p)^2$ and interpolating f in the parameter variable. See again Section 3.1 for more on the interpolation.

5 Handling the branch cut

This section treats the question of handling the problems which might arise from the branch cut of the complex logarithm, and we introduce a simple method for target points on the boundary, new for this paper, as well as a variation of a method seen in [20].

In the product integration and regularization we get expressions like

$$\log(1-z) - \log(-1-z)\,, \tag{71}$$

$$\log(1-z) - \log(-1-z) \mp \pi\mathrm{i}\,. \tag{72}$$

when integrating along a panel Λ starting in -1 and ending in 1, where we assume the branch cut of $\log(\tau - z)$ isn't crossed as τ traverses Λ. We could figure out whether the branch cut is crossed, which way it is crossed, and then compensate for it. But ideally we don't want to figure this out for every target point. If we now specify the principal branch of the logarithm, since this is what Matlab uses, we could write

$$\int_\Lambda \frac{\mathrm{d}\tau}{\tau - z} = \operatorname{Log}\left(\frac{1-z}{-1-z}\right), \tag{73}$$

which holds as long as the argument of $\tau - z$ varies less than π between the end points as τ traverses Λ. For a target point $z \in \Lambda$, we get to decide whether to imagine the panel being pushed in the positive or negative direction around z, as described in Section 3.2, and we could choose in such a way that (73) does hold for any reasonably shaped panel. We consider Λ oriented from -1 to 1, and if the target point $z \in \Lambda$ lies above the real axis, we push the panel to go in the positive direction around z, and if $z \in \Lambda$ lies below the real axis, we push the panel to go in the negative direction around z. Then we will always get the argument of $\tau - z$ varying less than π between the end points of the panel. If z is exactly on the real axis between -1 and 1, the principal branch gives $\arg((1-z)/(-1-z)) = \pi$ so we push the panel in the positive direction around z to match the principal branch. In Figure 3, a simple example panel is shown. The target point z has $\Im(z) \geq 0$, which means we have considered the circle segment oriented positively in order to make the conditions for (73) hold, the argument of $\tau - z$ varies less than π along the panel. To summarize, a method which appears to be new for this paper, although very simple, for $z \in \Lambda$,

$$\int_\Lambda \frac{\mathrm{d}\tau}{\tau - z} = \begin{cases} \operatorname{Log}\left(\frac{1-z}{-1-z}\right) - \pi\mathrm{i}\,, & \text{if } \Im(z) \geq 0\,, \\ \operatorname{Log}\left(\frac{1-z}{-1-z}\right) + \pi\mathrm{i}\,, & \text{if } \Im(z) < 0\,. \end{cases} \tag{74}$$

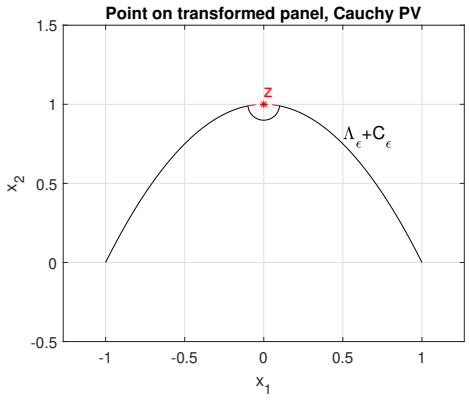


Figure 3: Example for Cauchy principal value and handling branch cut of the logarithm. Target point, z, in red.

If we instead have a point $z \notin \Lambda$, then it is more difficult to find some simple method of handling the branch cut by using the same kind of argument as the one leading to (74). Since the point is no longer on the boundary, we have to figure out if the point lies to the left or right of the boundary. One could do something similar to (74) and think that the panel wraps around z if $\Im(z) \geq 0$ and z lies to the right of the boundary, or if $\Im(z) < 0$ and z lies to the left of the boundary. But it's not so simple, as this fails for certain curves, and the shape of, not only the transformed panel, but also the boundary outside of the panel, plays a role. Another way to handle the problems arising from the branch cut of the logarithm is to assume that the panel is somewhat flat and implicitly deform the panel around z by letting $\phi = \pm\frac{\pi}{2}$, positive if z is to the left of the boundary, and negative if it is to the right and then split the logarithmic term,

$$\int_\Lambda \frac{\mathrm{d}\tau}{\tau - z} = \mathrm{Log}\left(\frac{1-z}{e^{\mathrm{i}\phi}(-1-z)}\right) + \mathrm{i}\phi\,. \tag{75}$$

This holds if the argument of $\tau - z$ along the deformed panel section between $z + e^{\mathrm{i}\phi}(-1-z)$ and 1 varies less than π. The basic idea is that one can deform the panel freely while keeping the value of the integral on the left hand side of (75) the same, as long as the start- and end points are the same and the argument of $\tau - z$ varies the same amount between the start- and end points as τ traverses Λ. So we deform the panel into one quarter circle around the target point z, which is integrated exactly, and what remains to integrate is a part of the panel which has been slightly 'unwrapped' around the target point, see Figure 4 and note the quarter circle part of the deformed panel called Λ' in the figure. The method in (75) is seen in [20], but with the angle $\phi = \pi/4$.

Below some code that handles the branch cut in the package is included with added variable descriptions, see Listing 1 and 2. Additionally, the code for constructing the rest of the p_k, q_k and r_k is included in Listing 3. For context, the matrices `P`, `Q` and `R` are of size $n \times m$, with n being the number of target points and m being the order of interpolation. The matrix `P` has elements $p_j(z_i)$, with z_i from the vector `ztgtrc`, with `Q` and `R` being defined equivalently for q_k and r_k. For more details on the specific implementation, the full code is available at `https://github.com/erikandersson98/BIE-CELib`.

Listing 1: Target point on the boundary Γ.

```
%Points on boundary.
sgn = ones(Nc,1); %Nc-number of close target points
sgn(imag(ztgtrc) < 0) = -1; %ztgtrc-close transformed
                                    %target points
argAdd = -sgn*pi*1i;
if j = 2 %true if target points are on the current
   panel, Lambda
    P(:,1)=argAdd+log((1-ztgtrc)./(-1-ztgtrc));
else
    P(:,1)=log((1-ztgtrc)./(-1-ztgtrc));
end
```

Listing 2: Target point not on the boundary Γ.

```
%Points not on boundary
gam=-1i*ones(Nt,1); %Nt-number of target points
loggam=-0.5i*pi*ones(Nt,1);
gam(ifleft)=1i; %ifleft-true if point left of boundary
loggam(ifleft)=0.5i*pi;
P=zeros(Nt,Ng); %Ng-order of Gauss quadrature
P(:,1)=loggam+log((1-ztgtr)./(gam.*(-1-ztgtr)));
                %ztgtr-transformed target points
```

Listing 3: Recursion for the rest of P, as well as construction of Q and R

```
R(:,1) = -1./(1-ztgtrc)+1./(-1-ztgtrc);
for k=1:ngl
    P(:,k+1)=ztgtrc.*P(:,k)+c(k);
    R(:,k+1) = -1./(1-ztgtrc)+(-1)^k./(-1-ztgtrc) + ...
            k*P(:,k);
    if mod(k,2) == 0
        Q(:,k) = (-P(:,k+1) + P(:,1))/k;
    elseif mod(k,2) == 1
        Q(:,k) = (-P(:,k+1) + ...
                log((1-ztgtrc).*(-1-ztgtrc)))/k;
    end
end
```

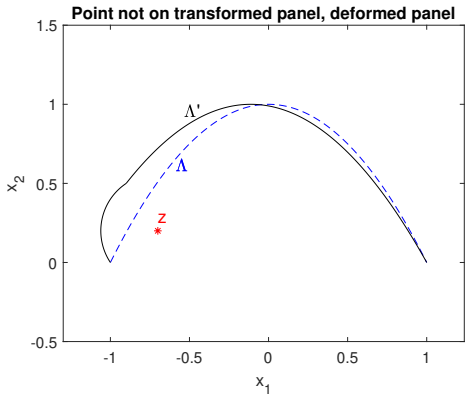


Figure 4: Deforming the panel as in the method (75). The dashed blue line is the original simple panel Λ, and the black curve is the deformed panel Λ'. The target point is in red.

6 Acoustic operators

6.1 Definitions

So far we have talked about the single-layer and double-layer potentials, the Neumann–Poincaré operator and M_{C}, M_{L} and M_{H}. In this section we move on to slightly more complicated operators that arise when solving the Helmholtz equation. In Section 6.1 we define the operators, and in Section 6.2 we describe a method involving splitting the kernel into parts with singularities we can handle with the product integration method or global/local regularization as described in Sections 3 and 4, seen in [13, 12].

We start with the fundamental solution to Helmholtz equation with wavenumber k,

$$\Phi_k(\boldsymbol{x},\boldsymbol{y}) = \frac{\mathrm{i}}{4}H_0^{(1)}(k|\boldsymbol{x}-\boldsymbol{y}|)\,, \tag{76}$$

where $H_0^{(1)}(x)$ is the zeroth order Hankel function of the first kind, $H_n^{(1)}(x) = J_n(x) + \mathrm{i}Y_n(x)$, with $J_n(x) and Y_n(x)$ being the Bessel functions of the first and second kind. We introduce the acoustic single-layer potential S_k,

$$S_k f(\boldsymbol{x}) = \int_\Gamma S_k(\boldsymbol{x},\boldsymbol{y}) f(\boldsymbol{y})\,\mathrm{d}|\boldsymbol{y}| = 2\int_\Gamma \Phi_k(\boldsymbol{x},\boldsymbol{y}) f(\boldsymbol{y})\,\mathrm{d}|\boldsymbol{y}|\,, \tag{77}$$

the acoustic double-layer potential K_k and its adjoint,

$$K_k f(\boldsymbol{x}) = \int_\Gamma K_k(\boldsymbol{x},\boldsymbol{y}) f(\boldsymbol{y})\,\mathrm{d}|\boldsymbol{y}| = 2\int_\Gamma \frac{\partial}{\partial \boldsymbol{n}_y}\Phi_k(\boldsymbol{x},\boldsymbol{y}) f(\boldsymbol{y})\,\mathrm{d}|\boldsymbol{y}|\,, \tag{78}$$

$$K_k^{\mathrm{A}} f(\boldsymbol{x}) = \int_\Gamma K_k^{\mathrm{A}}(\boldsymbol{x},\boldsymbol{y}) f(\boldsymbol{y})\,\mathrm{d}|\boldsymbol{y}| = 2\int_\Gamma \frac{\partial}{\partial \boldsymbol{n}_x}\Phi_k(\boldsymbol{x},\boldsymbol{y}) f(\boldsymbol{y})\,\mathrm{d}|\boldsymbol{y}|\,, \tag{79}$$

as well as the acoustic hypersingular operator T_k [7, Eqs. (3.8)-(3.11)],

$$T_k f(\boldsymbol{x}) = \int_\Gamma T_k(\boldsymbol{x},\boldsymbol{y}) f(\boldsymbol{y})\,\mathrm{d}|\boldsymbol{y}| = 2\int_\Gamma \frac{\partial^2}{\partial \boldsymbol{n}_x \partial \boldsymbol{n}_y}\Phi_k(\boldsymbol{x},\boldsymbol{y}) f(\boldsymbol{y})\,\mathrm{d}|\boldsymbol{y}|\,. \tag{80}$$

Performing the differentiation using

$$\frac{\mathrm{d}}{\mathrm{d}x}H_0^{(1)}(x) = -H_1^{(1)}(x)\,, \tag{81}$$

$$\frac{\mathrm{d}}{\mathrm{d}x}H_1^{1}(x) = \frac{1}{x}H_1^{(1)}(x) - H_2^{(1)}(x)\,, \tag{82}$$

together with the chain rule yields the kernels

$$S_k(\boldsymbol{x},\boldsymbol{y}) = \frac{\mathrm{i}}{2}H_0^{(1)}(k|\boldsymbol{x}-\boldsymbol{y}|)\,, \tag{83}$$

$$K_k(\boldsymbol{x},\boldsymbol{y}) = \frac{\mathrm{i}}{2}k|\boldsymbol{x}-\boldsymbol{y}|H_1^{(1)}(k|\boldsymbol{x}-\boldsymbol{y}|)\frac{\boldsymbol{n}_y\cdot(\boldsymbol{x}-\boldsymbol{y})}{|\boldsymbol{x}-\boldsymbol{y}|^2}\,, \tag{84}$$

$$K_k^{\mathrm{A}}(\boldsymbol{x},\boldsymbol{y}) = -\frac{\mathrm{i}}{2}k|\boldsymbol{x}-\boldsymbol{y}|H_1^{(1)}(k|\boldsymbol{x}-\boldsymbol{y}|)\frac{\boldsymbol{n}_x\cdot(\boldsymbol{x}-\boldsymbol{y})}{|\boldsymbol{x}-\boldsymbol{y}|^2}\,, \tag{85}$$

$$\begin{aligned} T_k(\boldsymbol{x},\boldsymbol{y}) = {}& \frac{\mathrm{i}}{2}k|\boldsymbol{x}-\boldsymbol{y}|H_1^{(1)}(k|\boldsymbol{x}-\boldsymbol{y}|)\frac{\boldsymbol{n}_x\cdot\boldsymbol{n}_y}{|\boldsymbol{x}-\boldsymbol{y}|^2} \\ & - \frac{\mathrm{i}}{2}(k|\boldsymbol{x}-\boldsymbol{y}|)^2H_2^{(1)}(k|\boldsymbol{x}-\boldsymbol{y}|)\frac{\boldsymbol{n}_y\cdot(\boldsymbol{x}-\boldsymbol{y})}{|\boldsymbol{x}-\boldsymbol{y}|^2}\frac{\boldsymbol{n}_x\cdot(\boldsymbol{x}-\boldsymbol{y})}{|\boldsymbol{x}-\boldsymbol{y}|^2}\,. \end{aligned} \tag{86}$$

The reason for writing the kernels on this form, without cancelling out certain $|\boldsymbol{x}-\boldsymbol{y}|$ terms, is because one can recognize parts which look much like the kernels of the Laplace double-layer potential and its adjoint. Like before, by identification between $\mathbb{R}^2$ and $\mathbb{C}$, $z=\boldsymbol{x}$, $\tau=\boldsymbol{y}$, and replacing the dot products by $\boldsymbol{a}\cdot\boldsymbol{b} = \Re(a\bar{b})$, these kernels can be considered functions of two complex variables. We introduce the notation

$$D_x(\boldsymbol{x},\boldsymbol{y}) = -\frac{\boldsymbol{n}_x\cdot(\boldsymbol{x}-\boldsymbol{y})}{|\boldsymbol{x}-\boldsymbol{y}|^2} \Longleftrightarrow D_z(z,\tau) = -\frac{\Re(n_z(\bar{z}-\bar{\tau}))}{|z-\tau|^2}\,, \tag{87}$$

$$D_y(\boldsymbol{x},\boldsymbol{y}) = \frac{\boldsymbol{n}_y\cdot(\boldsymbol{x}-\boldsymbol{y})}{|\boldsymbol{x}-\boldsymbol{y}|^2} \Longleftrightarrow D_\tau(z,\tau) = \frac{\Re(n_\tau(\bar{z}-\bar{\tau}))}{|z-\tau|^2}\,. \tag{88}$$

6.2 Kernel split

The way we intend to handle these more complicated operators is to split a general kernel $G(z,\tau)$, standing in for one of the kernels introduced in Section 6.1, into parts which we know how to handle [12, Sec. 6]. The split is formulated similar to [13, Eq. (43)], with an added coefficient c to correct for a typographical error. See the remark at the end of this section.

$$\begin{aligned} G(z,\tau)\,\mathrm{d}|\tau| = {}& G_0(z,\tau)\,\mathrm{d}|\tau| + G_{\mathrm{L}}(z,\tau)\log|z-\tau|\,\mathrm{d}|\tau| \\ & + c\Im\left(\frac{G_{\mathrm{C}}(z,\tau)\,\mathrm{d}\tau}{z-\tau}\right) + \Im\left(\frac{G_{\mathrm{H}}(z,\tau)\,\mathrm{d}\tau}{(z-\tau)^2}\right), \end{aligned} \tag{89}$$

where $G_0(z,\tau)$, $G_{\mathrm{L}}(z,\tau)$, $G_{\mathrm{C}}(z,\tau)$ and $G_{\mathrm{H}}(z,\tau)$ are smooth functions, and c is a complex coefficient. Then we can handle the logarithmic-, Cauchy-, and hypersingular singularities by a previously described method. In practice, the idea described in Section 3.4, where, for field evaluation, first regular 16-point Gauss–Legendre quadrature was used everywhere and then compensation terms were added for target points close to some panel, can be used here. Then we

never need to consider G_0, unless the target point is exactly a quadrature node, in which case there is some need to consider the limit $G_0(z,\tau)$ as $\tau \to z$.

If z is not a quadrature node then, and f is some function,

$$\int_\Gamma G(z,\tau)f(\tau)\,\mathrm{d}|\tau| \approx \sum_{j=1}^{16N} w_j G(z,\tau_j) f_j |\tau_j'| + \sum_{i\in I_c}\sum_{j=1}^{16} \left(w_{i,j}^{\mathrm{cmpL}}(z) G_{\mathrm{L},i,j} + c\Im\left(w_{i,j}^{\mathrm{cmpC}}(z) G_{\mathrm{C},i,j} \right) + \Im\left(w_{i,j}^{\mathrm{cmpH}}(z) G_{\mathrm{H},i,j} \right) \right) f_{i,j}, \tag{90}$$

where $w_{i,j}^{\mathrm{cmpL}}$ are compensation weights related to the logarithmic operator, $w_{i,j}^{\mathrm{cmpC}}$ are compensation weights related to the Cauchy singular operator multiplied by $\pi\mathrm{i}$ and $w_{i,j}^{\mathrm{cmpH}}$ are compensation weights related to the hypersingular operator multiplied by $\pi\mathrm{i}$. Note that the weights w_j used in the standard quadrature are 'rescaled weights' as introduced in Section 1.1.1, they include a factor which accounts for the panel not spanning the canonical interval. Also, the shorthand $G(z,\tau_{i,j}) = G_{i,j}$ has been used.

The compensation weights are not explicitly described here as their exact construction differs depending on which method is used in the close evaluation, as well as depending on if the target point is on or off the boundary. As mentioned previously in Section 3.4, they contain one part which corrects for the inappropriate terms of the standard Gauss–Legendre quadrature, and one part which is related to either product integration or regularization, including, potentially, some terms arising from the change in variables as described in Section 3.3.

If z is exactly a quadrature node, $z = \tau_i$, as in the Nyström discretization, then

$$\int_\Gamma G(\tau_i,\tau)f(\tau)\,\mathrm{d}|\tau| \approx \sum_{\tau_j \neq \tau_i}^{16N} w_j G(\tau_i,\tau_j) f_j |\tau_j'| + \lim_{\tau\to\tau_i} G_0(\tau_i,\tau) f_i w_i |\tau_i'| + \sum_{i\in I_c}\sum_{j=1}^{16} \left(w_{i,j}^{\mathrm{cmpL}}(z) G_{\mathrm{L},i,j} + c\Im\left(w_{i,j}^{\mathrm{cmpC}}(z) G_{\mathrm{C},i,j} \right) + \Im\left(w_{i,j}^{\mathrm{cmpH}}(z) G_{\mathrm{H},i,j} \right) \right) f_{i,j}. \tag{91}$$

Now, if nothing else is stated, $c = 1$ and any smooth term in (89) is 0. Using the representation of the Bessel function of the second kind from [1], we split the kernels.

For the acoustic operators we have, for S_k,

$$S_{k,\mathrm{L}}(z,\tau) = -\frac{1}{\pi} J_0(k|z-\tau|)\,, \tag{92}$$

$$\lim_{\tau\to z} S_{k,0}(z,\tau) = \frac{\mathrm{i}}{2} - \frac{1}{\pi}\left(\log\frac{k}{2} - \psi(1) \right), \tag{93}$$

where ψ is the digamma function.

For K_k,

$$K_{k,\mathrm{L}}(z,\tau) = -\frac{1}{\pi}k|z-\tau|J_1(k|z-\tau|)D_y(z,\tau)\,, \tag{94}$$

$$K_{k,\mathrm{C}}(z,\tau) = -\frac{1}{\pi}\,, \tag{95}$$

$$\lim_{\tau\to z} K_{k,0}(z,\tau) = 0\,, \tag{96}$$

but for smooth boundaries, the Cauchy term in (89) is smooth, so for a smooth boundary Γ and target point $z = \tau(p)$ on the boundary, we will have

$$K_{k,\mathrm{C}}(z,\tau) = 0\,, \tag{97}$$

$$\lim_{\tau\to z} K_{k,0}(z,\tau) = \frac{n_\tau \tau''(p)}{2\pi|\tau'(p)|^2}\,. \tag{98}$$

For K_k^{A},

$$K^{\mathrm{A}}_{k,\mathrm{L}}(z,\tau) = -\frac{1}{\pi}k|z-\tau|J_1(k|z-\tau|)D_x(z,\tau)\,, \tag{99}$$

$$K^{\mathrm{A}}_{k,\mathrm{C}}(z,\tau) = \frac{n_z\bar{n}_\tau}{\pi}\,, \tag{100}$$

$$\lim_{\tau\to z} K^{\mathrm{A}}_{k,0}(z,\tau) = 0\,. \tag{101}$$

Again, for smooth boundaries the Cauchy term is smooth, so for a smooth boundary Γ and target point $z = \tau(p)$ on the boundary, we will have

$$K^{\mathrm{A}}_{k,\mathrm{C}}(z,\tau) = 0\,, \tag{102}$$

$$\lim_{\tau\to z} K^{\mathrm{A}}_{k,0}(z,\tau) = \frac{n_\tau \tau''(p)}{2\pi|\tau'(p)|^2}\,. \tag{103}$$

Lastly, for T_k,

$$\begin{aligned} T_{k,\mathrm{L}}(z,\tau) = &-\frac{k}{\pi}J_1(k|z-\tau|)\frac{\Re(n_z\bar{n}_\tau)}{|z-\tau|^2} \\ &-\frac{1}{k}(k|z-\tau|^2)J_2(k|z-\tau|)D_z(z,\tau)D_\tau(z,\tau)\,, \end{aligned} \tag{104}$$

$$T_{k,\mathrm{C}}(z,\tau) = \frac{1}{2\pi}\Re\left(n_z(\bar{\tau}-\bar{z})\right)\,, \tag{105}$$

$$c = k^2\,, \tag{106}$$

$$T_{k,\mathrm{H}}(z,\tau) = -\frac{n_z}{\pi}\,, \tag{107}$$

$$\lim_{\tau\to z} T_{k,0}(z,\tau) = \frac{k^2\mathrm{i}}{4} - \frac{k^2}{4\pi}\left(2\log\frac{k}{2} - 2\psi(1) - 1\right). \tag{108}$$

Remark 1: The term $T_{k,\mathrm{C}}$ contains a typographical error in [13, Eq. (64)], since the k^2 part should be moved outside the imaginary part, leading to the c coefficient. This will matter if a complex wavenumber k is used on a non-smooth boundary. But for smooth boundaries we have as before

$$T_{k,\mathrm{C}}(z,\tau) = 0\,, \tag{109}$$

with no change in $\lim_{\tau\to z} T_{k,0}(z,\tau)$.

7 Method

Three different categories of tests were performed for assessing the success of implementation of the integral operators described in Sections 2 and 6.

First, integral operators were tested directly with matrix-vector multiplication or solution of a linear system of equations, where the result of the operation was compared to a known analytic solution, or an approximated solution. Here some operators identities were also tested. Second, the Laplace equation and related operators in Section 2 were tested by a solving the Interior Laplace problem with Dirichlet boundary conditions (ILD) and the Laplace transmission problem (LT). Third, the Helmholtz equation and related operators in Section 6 were tested by solving the Exterior Helmholtz problem with Dirichlet boundary conditions (EHD), and with Neumann boundary conditions (EHN).

The different ways of implementing the operators for close evaluation on the boundary are described in Sections 3 and 4. The following shorthand is used. A method using product integration is called PI-z or PI-p depending on if the product integration is used for a complex integration variable (PI-z) or the parameter integration variable s of the boundary parametrization $\tau(s)$ (PI-p). The global regularization method for M_{C} is called GR. The local regularization method is called LR-z or LR-p, again depending on if the layer density is interpolated as a polynomial of the complex variable z (LR-z), or the real parameter s, related to the parametrization (LR-p).

In all tests the boundary used was the starfish (6) with $a = 0.3$. In field evaluation, a 300×300 equidistant grid was used in the square with corners in $(\pm 1.5, \pm 1.5)$. All tests were carried out on a unit with the processor AMD Ryzen 5 5500U and 8GB of RAM using the 2022a release of MATLAB. The linear system of equations resulting from Nyström discretization was solved by a GMRES method with stopping criterion threshhold of machine epsilon in the relative residual, roughly $2.22 \cdot 10^{-16}$, or a maximum iteration of 100. The specific GMRES method is described in [14, Sec. 8], and was supplied by the project supervisor Johan Helsing.

In the program package, a function is included for calculation of the Gauss–Legendre nodes and weights when the number of nodes is not 16. It is a slightly altered version of [2].

7.1 Operator implementation tests (Matrix-vector multiplication / Operator Identities)

Matrix-vector multiplication in the expression

$$Mf(z) \approx \boldsymbol{M}\boldsymbol{f}\,, \tag{110}$$

was tested for the operators M_{C}, M_{L} and M_{H}, for some test function f with $\boldsymbol{M}$ being the discretization using Gauss–Legendre quadrature and one of the described methods, product integration or global/local regularization. For M_{C} and M_{H}, the test function was

$$f(z) = z^6 + z^{-6}\,. \tag{111}$$

For comparison, this has analytic solution

$$M_{\mathrm{C}} f = z^6 - z^{-6}\,, \tag{112}$$

$$M_{\mathrm{H}} f = 6z^5 + 6z^{-7}\,. \tag{113}$$

For M_{L} the test function was

$$f(z) = |z^6 + z^{-6}|\,. \tag{114}$$

This has no simple analytic solution, so comparison was done with a fine discretization $\boldsymbol{M}_{\mathrm{L}}\boldsymbol{f}$, which approximates the exact solution.

For M_{C} the Poincaré–Bertrand identity

$$M_{\mathrm{C}} M_{\mathrm{C}} = I\,, \tag{115}$$

holds. This identity was tested by discretizing the operator and solving a system of equations

$$\boldsymbol{M}_{\mathrm{C}}\boldsymbol{M}_{\mathrm{C}}\boldsymbol{g} = \boldsymbol{f}\,. \tag{116}$$

If the implementation is good then $\boldsymbol{g} \approx \boldsymbol{f}$ should hold.

For the acoustic operators the Calderón identities hold [7, Eqs. (3.12) and (3.13)]

$$K_k K_k - S_k T_k = I\,, \tag{117}$$

$$K_k^{\mathrm{A}} K_k^{\mathrm{A}} - T_k S_k = I\,. \tag{118}$$

These were tested by discretization of the operators using different methods for handling close evaluation. Then matrix-vector multiplication was performed with the test function (111). Four different pairs of integral operator implementation were considered according to Table 1.

	Pair 1	Pair 2	Pair 3	Pair 4
M_{L}	PI-p	PI-z	PI-p	PI-p
M_{H}	PI-z	PI-z	LR-p	LR-z

Table 1: Method pairs for testing the Calderón identities

7.2 The Laplace Equation

The formulation of the Laplace equation for an interior domain is repeated, for a boundary Γ with Dirichlet conditions

$$\Delta U(\boldsymbol{x}) = 0\,, \qquad \boldsymbol{x} \in D\,, \tag{119}$$

$$\lim_{D \ni \boldsymbol{x}' \to \boldsymbol{x}} U(\boldsymbol{x}') = g(\boldsymbol{x})\,, \qquad \boldsymbol{x} \in \Gamma\,. \tag{120}$$

The solution $U(\boldsymbol{x})$ can be expressed as a boundary integral equation with a layer density $\mu(\boldsymbol{x})$ assumed real,

$$U(\boldsymbol{x}) = -K\mu(\boldsymbol{x})\,, \qquad \boldsymbol{x} \in D\,, \tag{121}$$

with K being the Neumann–Poincaré operator. From jump relations,

$$(I - K)\mu(\boldsymbol{x}) = g(\boldsymbol{x}), \qquad x \in \Gamma. \tag{122}$$

The formulation of the Laplace transmission problem, from [15], for a boundary Γ is

$$\Delta U(\boldsymbol{x}) = 0, \qquad \boldsymbol{x} \in \mathbb{R}^2 \setminus \Gamma, \tag{123}$$

$$\epsilon_1 \lim_{\boldsymbol{x}' \to \boldsymbol{x}} \frac{\partial}{\partial \boldsymbol{n}_x} U^{\text{ext}}(\boldsymbol{x}') = \epsilon_2 \lim_{\boldsymbol{x}' \to \boldsymbol{x}} \frac{\partial}{\partial \boldsymbol{n}_x} U^{\text{int}}(\boldsymbol{x}'), \qquad \boldsymbol{x} \in \Gamma, \tag{124}$$

$$\lim_{|\boldsymbol{x}| \to \infty} \nabla U(\boldsymbol{x}) = \boldsymbol{e}, \tag{125}$$

where $\boldsymbol{e}$ is an applied field of unit magnitude, ϵ_1 and ϵ_2 are two constants, and the superscripts ext and int differ between the limit approaching the boundary from the exterior domain and the interior domain. The solution U is expressed as

$$U(\boldsymbol{x}) = \boldsymbol{x} \cdot \boldsymbol{e} + S\mu(\boldsymbol{x}), \qquad \boldsymbol{x} \in \mathbb{R}^2 \setminus \Gamma, \tag{126}$$

with S being the single-layer potential (29). From jump relations we get

$$(I + \lambda K^{\text{A}})\mu(\boldsymbol{x}) = -2\lambda(\boldsymbol{e} \cdot \boldsymbol{n}_x), \tag{127}$$

with K^{A} being the adjoint of the Neumann–Poincaré operator, and the parameter λ,

$$\lambda = \frac{\epsilon_2 - \epsilon_1}{\epsilon_2 + \epsilon_1}. \tag{128}$$

These two problems were implemented and solved for $g(z) = \Re(z)$ in the interior domain problem and $\boldsymbol{e} = (1, 0)$ in the transmission problem.

7.3 The Helmholtz Equation

The Helmholtz equation on an exterior domain E with Dirichlet conditions on a boundary Γ is

$$\Delta U(\boldsymbol{x}) + k^2 U(\boldsymbol{x}) = 0, \qquad \boldsymbol{x} \in E, \tag{129}$$

$$\lim_{E \ni \boldsymbol{x}' \to \boldsymbol{x}} U(\boldsymbol{x}') = g(\boldsymbol{x}), \qquad \boldsymbol{x} \in \Gamma, \tag{130}$$

$$\lim_{|\boldsymbol{x}| \to \infty} \sqrt{|\boldsymbol{x}|}\left(\frac{\partial}{\partial |\boldsymbol{x}|} - ik\right) U(\boldsymbol{x}) = 0, \tag{131}$$

where (131) is the added Sommerfeld radiation condition. The solution can be expressed as a boundary integral equation with a layer density μ [7, Eq. (3.26)]

$$U(\boldsymbol{x}) = \left(\frac{1}{2} K_k - \frac{\mathrm{i}k}{4} S_k\right) \mu(\boldsymbol{x}), \qquad \boldsymbol{x} \in E. \tag{132}$$

From jump relations we end up with

$$\left(I + K_k - \frac{\mathrm{i}k}{2} S_k\right) \mu(\boldsymbol{x}) = 2g(\boldsymbol{x}), \qquad \boldsymbol{x} \in \Gamma. \tag{133}$$

For Neumann conditions on the boundary, (130) is switched for

$$\lim_{E \ni \boldsymbol{x}' \to \boldsymbol{x}} \frac{\partial}{\partial \boldsymbol{n}_x} U(\boldsymbol{x}') = g(\boldsymbol{x}) \,, \qquad \boldsymbol{x} \in \Gamma \,, \tag{134}$$

and we express U as a boundary integral equation, similar to [7, Eq. (3.29)] but using the regularizing operator suggested in [5],

$$U(\boldsymbol{x}) = \left(\frac{1}{2} S_k + \frac{\mathrm{i}}{4} K_k S_{\mathrm{i}k} \right) \mu(\boldsymbol{x}) \,, \qquad \boldsymbol{x} \in E \,. \tag{135}$$

From (130) and jump relations [7, Thm. 3.1],

$$(I - K_k^{\mathrm{A}} - \mathrm{i} T_k S_{\mathrm{i}k}) \mu(\boldsymbol{x}) = -2 g(\boldsymbol{x}) \,, \qquad \boldsymbol{x} \in \Gamma \,. \tag{136}$$

These two problems were implemented using different methods of discretizing the operators on the boundary. The function g was chosen to be from a point source in the point $\boldsymbol{x}_0 = (0.3, 0.5)$. For Dirichlet boundary conditions,

$$g(\boldsymbol{x}) = H_0^{(1)}(k|\boldsymbol{x}_0 - \boldsymbol{x}|) \,, \qquad \boldsymbol{x} \in \Gamma \,, \tag{137}$$

and for Neumann boundary conditions,

$$g(\boldsymbol{x}) = k H_1^{(1)}(k|\boldsymbol{x}_0 - \boldsymbol{x}|) \frac{\boldsymbol{n}_x \cdot (\boldsymbol{x}_0 - \boldsymbol{x})}{|\boldsymbol{x}_0 - \boldsymbol{x}|} \,, \qquad \boldsymbol{x} \in \Gamma. \tag{138}$$

We know the analytic solution U for these boundary conditions to be the g used in (137), but for $\boldsymbol{x} \in E$. The wavenumber was chosen to be $k = 10$.

7.4 Screening functions

In addition to the operators and tests mentioned in Sections 7.1-7.3, two screening functions were implemented as a part of the package. One that determines which panels a point is close to by checking the minimum distance to a quadrature node divided by the length of the panel on which is it situated, as determined by 16-point Gauss–Legendre quadrature. If this quantity is less than a constant d_{lim} the point is considered close to the panel. In the implementation $d_{\mathrm{lim}} = 1.1$ was chosen.

A second version of the screening function that determines if a point is close to the boundary is suggested in [20, eq. (27)]. It is based on the fact that there are known convergence results for the Gauss-Legendre quadrature if the integrand is analytic within a Bernstein ellipse in the parameter variable. For flat panels, the Bernstein ellipse in the parameter variable is well-approximated by an ellipse with foci at the end points of the panel in the complex plane, leading to the rule that a point is close if it lies within such an ellipse,

$$|z - \tau(a)| + |z - \tau(b)| < CS, \tag{139}$$

where the panel has end points $\tau(a)$ and $\tau(b)$, S is the length of the panel, and C is a constant. In [20], $C = 2.5$ is suggested, and this value of C is tested to see if there is any difference in the numerical results compared to the first version of the screening function.

The second screening function determines if a point z is to the left of the right of the boundary by checking the dot product between the outward unit normal n_{τ_c} and the vector $z - \tau_c$, where τ_c is the closest quadrature node on the boundary. Note that we use complex notation here, but consider the dot product and angles between the corresponding vectors in $\mathbb{R}^2$. If the dot product is positive and the angle between n_{τ_c} and $z - \tau_c$ is less than some constant angle ϕ_{lim} then the point is considered to be on the left, if the dot product is negative with the same condition for the angle, the point is considered to be on the right. If the condition on the angle doesn't hold, the boundary is iteratively refined close to τ_c, until a closest point is found for which the condition on the angle holds. The function assumes that the boundary is smooth. In the implementation $\phi_{\text{lim}} = \pi/8$ was chosen.

These screening functions are only utilized for target points off the boundary. When the target point is on the boundary, we clearly don't need the second screening function. For determining which target points a panel lies close to, we only look at points on the adjacent panels, and have a distance condition for a target point in the coordinates of the transformed panel:

$$|z| < 2. \tag{140}$$

If the condition holds then the target point z is close to the transformed panel.

8 Results

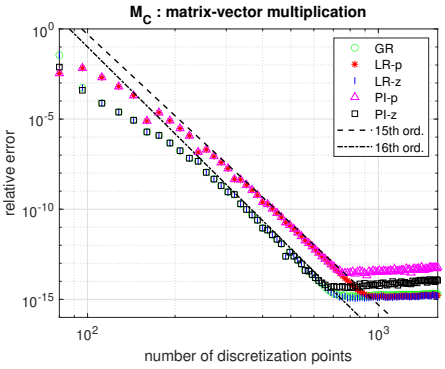
8.1 Operator identities/matrix-vector multiplication

The results of the two tests of the M_C operator can be seen in Figure 5. Note in Figure 5(a) how the methods PI-z, GR and LR-z seem to follow the added line which shows 16th order convergence, while the methods LR-p and PI-p seem to follow 15th order convergence. In Figure 5(b), the convergence is not as uniform, except for the method PI-p, which seems to follow a 15th order convergence. At 1600 discretization points, or 100 panels using 16-point quadrature, the methods stop converging, and LR-p is by far the most accurate method at that point.

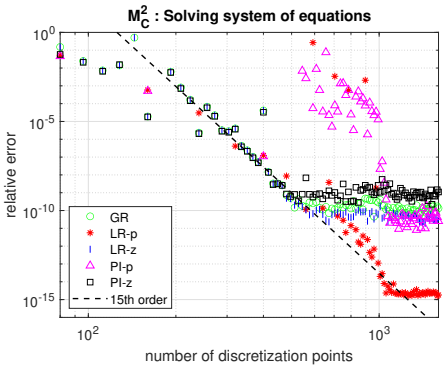
Figure 6 shows the convergence of matrix-vector multiplication for the M_L operator. Both tested methods, PI-p and PI-z, seem close in accuracy, following 14th order convergence and reaching a similar final accuracy at 2080 discretization points, or 130 panels using 16-point quadrature.

Figure 7 shows the convergence of matrix-vector multiplication for the M_H operator. The methods PI-z and LR-z seem to follow 16th order convergence, while LR-p has 15th order convergence. The method PI-z has lower final accuracy at around 10^{-12} while LR-p and LR-z reach final accuracy of around 10^{-13}.

Figure 8 shows the resulting error of matrix-vector multiplication using the two Calderón identities, (117) and (118). Pair 3 is by far the best of the methods, following 14th order convergence in both tests. The other pairs follow 14th order convergence in the test of the first identity, while in the second, they seemingly follow 15th order convergence at higher number of discretization points, though still behind Pair 3 in terms of accuracy.



(a) Relative error as a function of the number of discretization points for different methods of close evaluation for the operator M_{C}. Matrix-vector multiplication with test function (111).



(b) Relative error as a function of the number of discretization points for different methods of close evaluation for the operator M_{C}. Solving linear of system equations with matrix $M_{\mathrm{C}}M_{\mathrm{C}}$ for test function (111).

Figure 5: Operator tests for M_{C}.

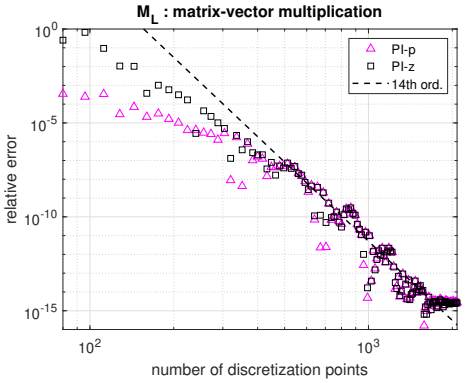


Figure 6: Relative error as a function of the number of discretization points for different methods of close evaluation for the operator M_{L}. Matrix-vector multiplication with test function (114).

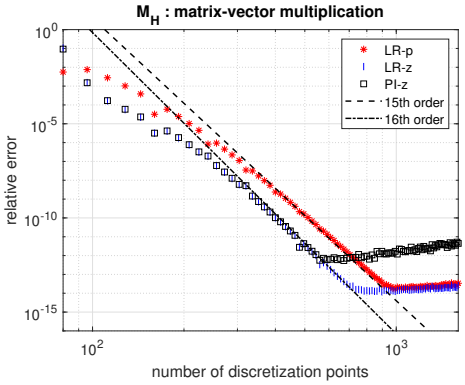
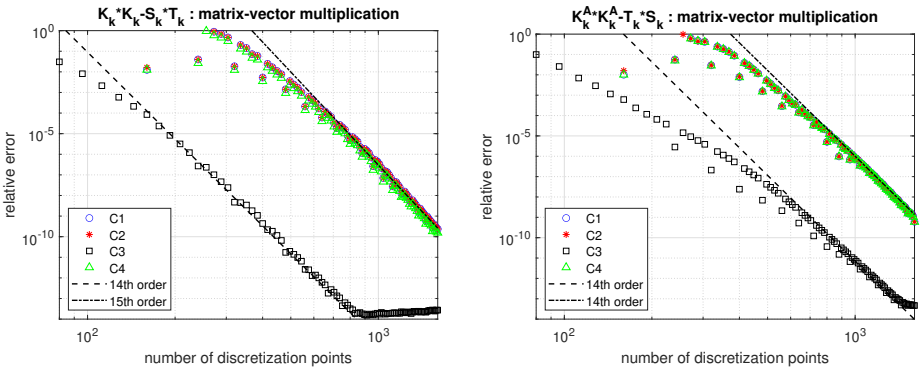


Figure 7: Relative error as a function of the number of discretization points for different methods of close evaluation for the operator M_{H}. Matrix-vector multiplication with test function (111).



(a) Relative error as a function of the number of discretization points for the 4 pairs described in Table 1. First Calderón identity, (117).

(b) Relative error as a function of the number of discretization points for the 4 pairs described in Table 1. Second Calderón identity, (118).

Figure 8: Calderón identities tests.

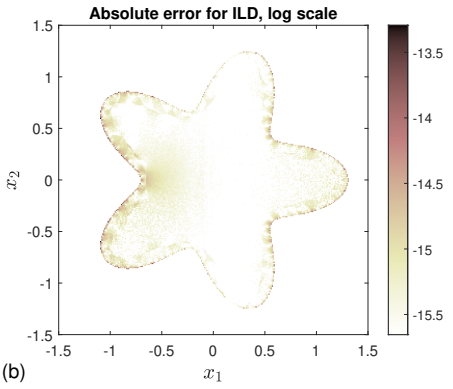
8.2 Helmholtz and Laplace problems

The error for the solution to Laplaces equation with Dirichlet boundary conditions as well as the Laplace transmission problem can be seen in Figure 9, with an error around 10^{-14} to 10^{-16} for 100 panels. The resulting field plot for the solution of the LT problem can be seen in Figure 10

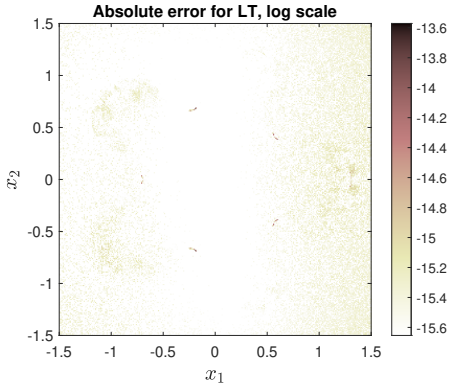
The error in the solution of the Exterior Helmholtz problem with Dirichlet boundary conditions can be seen in Figure 11. In Figure 11(a) we can see that the method PI-p for discretizing the boundary leads to a very low error far from the boundary, around 10^{-16}. while the method PI-z performs worse in this regard, with a far field error around 10^{-9} for 50 panels, as seen in Figure 11(b).

The error for the solution of the exterior Helmholtz problem with Neumann boundary conditions can be seen in Figure 12. Pair 3 leads to high accuracy far from the boundary, around 10^{-16}, as seen in Figure 12(a), while using Pair 1

has dire consequences for the performance of the implementation, with a far field error around 10^{-6} for 50 panels, see Figure 12(b).



(a) Absolute error in $\log_{10}$ scale for the solution of the ILD problem, 100 panels.



(b) Absolute error in $\log_{10}$ scale for the solution of the LT problem, 100 panels.

Figure 9: Field error plots for ILD and LT.

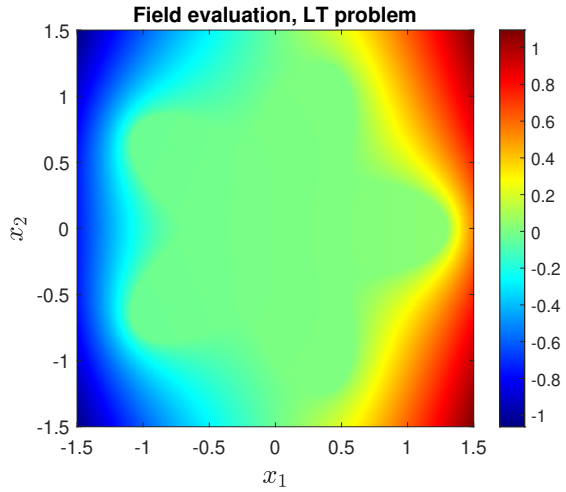
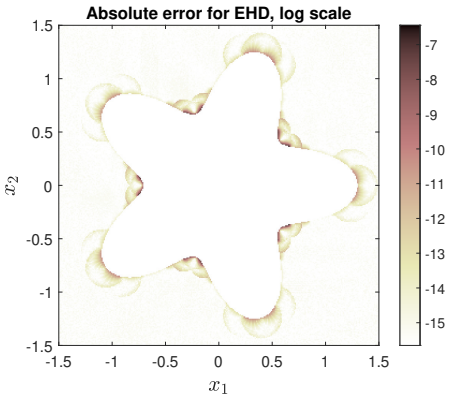
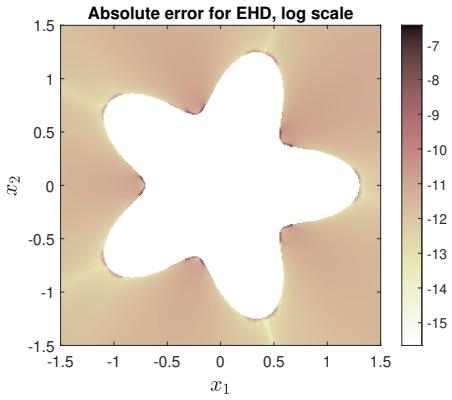


Figure 10: Field plot for the solution of the LT problem, 100 panels.

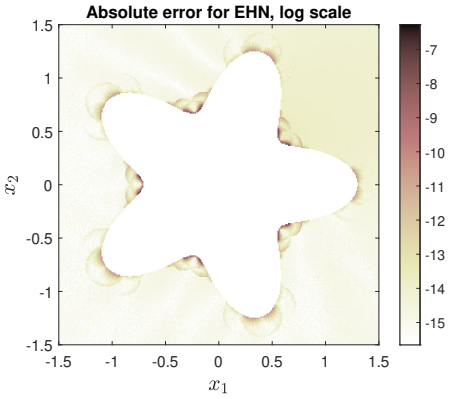


(a) Absolute error in $\log_{10}$ scale for the solution of the EHD problem, 50 panels. The method used for the logarithmic singular parts is PI-p. Wavenumber used was $k = 10$.

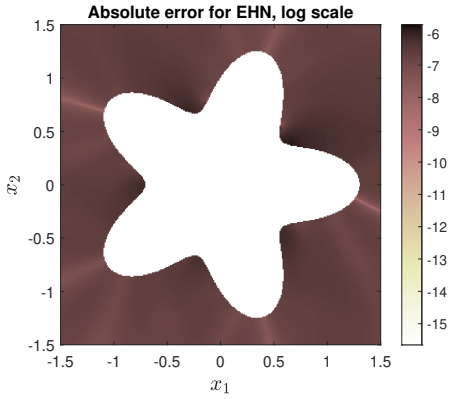


(b) Absolute error in $\log_{10}$ scale for the solution of the EHD problem, 50 panels. The method used for the logarithmic singular parts is PI-z. Wavenumber used was $k = 10$.

Figure 11: Field error plots for EHD.



(a) Absolute error in $\log_{10}$ scale for the solution of the EHN problem, 50 panels. Pair 3 used. Wavenumber used was $k = 10$.



(b) Absolute error in $\log_{10}$ scale for the solution of the EHN problem, 50 panels. Pair 1 used. Wavenumber used was $k = 10$.

Figure 12: Field error plots for EHN.

9 Discussion

First, note that the tests conducted were all successful in the sense that all methods did converge to the correct answer, whenever the exact analytic solution was simple to compare to. There is little to say about the tests of the interior Laplace equation with Dirichlet boundary conditions and the Laplace transmission problem, except that they indicate that there are no glaring errors in the implementation of the Neumann–Poincaré operator on and off the boundary. The screening function for determining if a point is to the left of

the boundary seems to work. The handling of the logarithmic branch, and the product integration methods both seem to work. Note that for field evaluation close to the boundary, only the method PI-z works, since PI-p requires the target point to be expressed in the parametrization of the boundary, and the regularization methods only work for points on the boundary. An interesting idea in the newer papers [17] and [16] is to apply the method PI-p to points off the boundary by finding a pre-image of the target point. The recent paper [4] develops this idea as well. Close to the boundary in the Helmholtz equation problems, Figures 11 and 12, one can really see that evaluation is worse when comparing to points far from the boundary, so it's natural that much thought is given to ways of improving the accuracy of evaluation close to the boundary.

Things become more interesting when moving towards implementation of operators for which we need to use regularization or product integration on the boundary, since there are several different method choices to compare. In matrix-vector multiplication for both the operators M_{C} and M_{H}, there is an order of convergence differing the parameter versions of PI and LR, from the complex variable versions. One explanation for this could be that the test-function was a complex polynomial, and therefore using a complex monomial basis in this case could have a positive effect on convergence. In the testing of solving a linear system of equations for the Poincaré–Bertrand identity (115), the trend that PI-z performs better than PI-p is even more noticeable, especially for low resolution of the boundary. Eventually, however, as the resolution is increased, the parameter versions of LR and PI reach a better final accuracy. The test function $g(z) = \cos(z)$ was also tried, and the convergence shows the same trends, that the z-versions are better for low resolution, while the parameter versions eventually reach a better final accuracy. The parameter versions also show this sudden drop in error, as can be seen somewhat in Figure 5(b) for LR-p and PI-p. Since the convergence structure was similar for the test using the cosine function, the explanation that the polynomial test function was the cause of difference in convergence seems inadequate, and the differences rather seem inherent to the methods.

For matrix-vector multiplication with the operator M_{L}, the differences are minimal, except maybe that PI-p is slightly better for lower resolution of the boundary. See Figure 6.

The tests of the Calderón identities, presented in Figure 8, show very clearly that Pair 3 is the better implementation. The special thing about Pair 3 is that the hyper-singular operator M_{H} is discretized through the LR-p method, local regularization with interpolation in the parameter variable. Then, it is interesting that LR-p does not stand out in the matrix-vector multiplication case for M_{H}, only when it is used to discretize the Calderón identities. We see generally that the performance of the methods in matrix-vector multiplication with the operators M_{C}, M_{L} and M_{H} does not exactly indicate that the same methods will perform well for implementation of other identities, or for use in a practical setting when solving a differential equation cast as a BIE.

Despite the PI-p and PI-z methods performing similarly for the logarithmic operator M_{L} in matrix-vector multiplication, the parameter version PI-p clearly outperforms PI-z when used to solve the exterior Helmholtz equation with Dirichlet boundary conditions, as the field far from the boundary has a much less significant error at 50 panels. The difference between the methods seems to get smaller as the wavenumber increases from $k = 10$.

For solving the exterior Helmholtz equation with Neumann boundary conditions, Pair 3 does perform much better than Pair 1, reflecting the trends in the tests of discretization of the Calderón identities. So LR-p is a good method to use for the hypersingular operator in a practical setting, as well.

Because we always worked with a smooth boundary and a certain choice of PDEs and integral equations, the implementation of the Cauchy operator on the boundary was never tested in a practical problem. In the matrix-vector multiplication test all methods performed similarly, except a difference in order of convergence between the z-versions and the parameter versions. If one has a non-smooth boundary, then the Cauchy parts of the kernels of K_k, K_k^{A} and T_k would have to be calculated using one of the implementations. For smooth boundaries we could include the Cauchy part in the smooth part G_0.

Using the second version of the function which determines if a target point is close to the boundary or not by use of an ellipse made very little difference in the results compared to the first, and so the two versions seem interchangeable. The two versions both use the idea of comparing the distance from a target point to points on the boundary and dividing by the length of the boundary, so it is not odd that they give similar results. The two methods also need some constant, C in the second version and d_{lim} in the first, and it is difficult to know exactly which value is best to use. There are some circular patterns in Figure 11(a) which are present for the first version but which are not present when using the second version. But these patterns are also not visible when lowering d_{lim} to 0.9, for example. This might indicate that $d_{\mathrm{lim}} = 1.1$ is not ideal, at least not for this particular test.

9.1 Further improvements

There are several ways the package could be expanded and improved. For example, the Fast Multipole Method (FMM), introduced in [8], is often applied to calculate matrix-vector multiplication with the part of discretized operator which corresponds to evaluation far from the boundary. The part of the discretized operator which corresponds to close evaluation is contained in a diagonal of constant width in the operator matrix. The FMM can then reduce the time complexity of the whole matrix-vector multiplication.

Another step which could be added to save time is upsampling of the layer density. The idea is that, often, the boundary is refined enough to give an accurate solution for the layer density $\boldsymbol{\mu}$ after solving the linear system that results from Nyström discretization. But the field evaluation is still inadequate because the boundary isn't refined enough to give an accurate discretization with regards to the kernel of the integral operator. So then two grids are considered, where μ is solved on the coarser one, and then upsampled to the finer one which is then used in field evaluation. The same idea could be used to precondition the Nyström discretization itself by using matrices corresponding to interpolation between these grids, as described in [11, Sec. 5].

As a final suggestion for an improvement, methods could be added to deal with non-smooth or highly curved boundaries in more effective ways. Methods like iterative refinement of the boundary for parts with high curvature or parts close to corners, as described in [20, Sec. 4].

9.2 Conclusions

In conclusion, the results indicate that the resulting package is adequate for solving planar problems for the Laplace and Helmholtz equation. For implementation of logarithmic singular operators on the boundary, the PI-p method is suggested, product integration with interpolation in the parameter variable, as it gives better results in the field far from the boundary. For implementation of hypersingular operators on the boundary, the LR-p method is suggested, local regularization with interpolation in the parameter variable, because the results are far better for this method when it comes to the hypersingular part of the T_k operator.

The package is available on GitHub at `https://github.com/erikandersson98/BIE-CELib`.

Acknowledgement

This work was prepared under partial support by the Swedish Research Council via contract 2021-03720 to Johan Helsing.

References

[1] https://functions.wolfram.com/Bessel-TypeFunctions/BesselY/06/01/04/01/02/0008/, Retrieved May 3, 2023.

[2] Greg von Winckel (2023). Legendre-Gauss Quadrature Weights and Nodes (https://www.mathworks.com/matlabcentral/fileexchange/4540-legendre-gauss-quadrature-weights-and-nodes), MATLAB Central File Exchange. Retrieved June 1, 2023.

[3] Kazunori Ando, Hyeonbae Kang, Yoshihisa Miyanishi, and Mihai Putinar. *Spectral analysis of Neumann-Poincaré operator*. 2020. arXiv: 2003.14387 [math.SP].

[4] Gang Bao, Wenmao Hua, Jun Lai, and Jinrui Zhang. *Singularity swapping method for nearly singular integrals based on trapezoidal rule*. 2023. arXiv: 2305.05855 [math.NA].

[5] Oscar Bruno, Tim Elling, and Catalin Turc. "Regularized integral equations and fast high-order solvers for sound-hard acoustic scattering problems". In: *International Journal for Numerical Methods in Engineering* 91.10 (2012), pp. 1045–1072. DOI: https://doi.org/10.1002/nme.4302.

[6] Elena Cherkaev, Minwoo Kim, and Mikyoung Lim. "Geometric series expansion of the Neumann–Poincaré operator: Application to composite materials". In: *European Journal of Applied Mathematics* 33.3 (2022), pp. 560–585. DOI: 10.1017/S0956792521000127.

[7] David Colton and Rainer Kress. *Inverse Acoustic and Electromagnetic Scattering Theory*. 3rd ed. Vol. 93. Applied Mathematical Sciences. Springer, 2013.

[8] L Greengard and V Rokhlin. "A fast algorithm for particle simulations". In: *Journal of Computational Physics* 73.2 (1987), pp. 325–348. ISSN: 0021-9991. DOI: https://doi.org/10.1016/0021-9991(87)90140-9.

[9] Leslie Greengard and Monique Moura. "On the numerical evaluation of electrostatic fields in composite materials". In: *Acta Numerica* 3 (1994), pp. 379–410. DOI: 10.1017/S0962492900002464.

[10] Shengtun Hao, A. Barnett, P. Martinsson, and P. Young. "High-order accurate methods for Nyström discretization of integral equations on smooth curves in the plane". In: *Advances in Computational Mathematics* 40 (2014). DOI: 10.1007/s10444-013-9306-3.

[11] Johan Helsing. "Integral equation methods for elliptic problems with boundary conditions of mixed type". In: *Journal of Computational Physics* 228.23 (2009), pp. 8892–8907. ISSN: 0021-9991. DOI: https://doi.org/10.1016/j.jcp.2009.09.004.

[12] Johan Helsing and Anders Holst. "Variants of an explicit kernel-split panel-based Nyström discretization scheme for Helmholtz boundary value problems". In: *Advances in Computational Mathematics* 41 (2015), pp. 691–708. DOI: 10.1007/s10444-014-9383-y.

[13] Johan Helsing and Anders Karlsson. "On a Helmholtz transmission problem in planar domains with corners". In: *Journal of Computational Physics* 371 (2018), pp. 315–332. ISSN: 0021-9991. DOI: https://doi.org/10.1016/j.jcp.2018.05.044.

[14] Johan Helsing and Rikard Ojala. "On the evaluation of layer potentials close to their sources". In: *Journal of Computational Physics* 227.5 (2008), pp. 2899–2921. ISSN: 0021-9991. DOI: https://doi.org/10.1016/j.jcp.2007.11.024.

[15] Johan Helsing and Karl-Mikael Perfekt. "On the polarizability and capacitance of the cube". In: *Applied and Computational Harmonic Analysis* 34.3 (2013), pp. 445–468. ISSN: 1063-5203. DOI: https://doi.org/10.1016/j.acha.2012.07.006.

[16] Ludvig af Klinteberg. *Singularity swap quadrature for nearly singular line integrals on closed curves in two dimensions*. 2023. arXiv: 2304.11865 [math.NA].

[17] Ludvig af Klinteberg and Alex Barnett. "Accurate quadrature of nearly singular line integrals in two and three dimensions by singularity swapping". In: *BIT Numerical Mathematics* 61 (July 2020), pp. 1–36. DOI: 10.1007/s10543-020-00820-5.

[18] Andreas Klöckner, Alexander Barnett, Leslie Greengard, and Michael ONeil. "Quadrature by expansion: A new method for the evaluation of layer potentials". In: *Journal of Computational Physics* 252 (2013), pp. 332–349. ISSN: 0021-9991. DOI: https://doi.org/10.1016/j.jcp.2013.06.027.

[19] Matthew J. Morse, Abtin Rahimian, and Denis Zorin. "A robust solver for elliptic PDEs in 3D complex geometries". In: *Journal of Computational Physics* 442 (2021), p. 110511. ISSN: 0021-9991. DOI: https://doi.org/10.1016/j.jcp.2021.110511.

[20] Bowei Wu, Hai Zhu, Alex Barnett, and Shravan Veerapaneni. "Solution of Stokes flow in complex nonsmooth 2D geometries via a linear-scaling high-order adaptive integral equation scheme". In: *Journal of Computational Physics* 410 (2020), p. 109361. ISSN: 0021-9991. DOI: https://doi.org/10.1016/j.jcp.2020.109361.

[21] Hai Zhu and Shravan Veerapaneni. "High-Order Close Evaluation of Laplace Layer Potentials: A Differential Geometric Approach". In: *SIAM Journal on Scientific Computing* 44.3 (2022), A1381–A1404. DOI: 10.1137/21M1423051.

Master's Theses in Mathematical Sciences 2023:E27
ISSN 1404-6342

LUTFNA-3052-2023

Numerical Analysis
Centre for Mathematical Sciences
Lund University
Box 118, SE-221 00 Lund, Sweden

http://www.maths.lu.se/