Solving Fredholm second-kind integral equations with singular right-hand sides on non-smooth boundaries

Johan Helsing [a], Shidong Jiang [b,c,*]

[a] *Centre for Mathematical Sciences, Lund University, Box 118, 221 00 Lund, Sweden*
[b] *Department of Mathematical Sciences, New Jersey Institute of Technology, Newark, NJ 07102, USA*
[c] *Center for Computational Mathematics, Flatiron Institute, Simons Foundation, New York, NY 10010, USA*

A R T I C L E   I N F O

A B S T R A C T

A numerical scheme is presented for the solution of Fredholm second-kind boundary integral equations with right-hand sides that are singular at a finite set of boundary points. The boundaries themselves may be non-smooth. The scheme, which builds on recursively compressed inverse preconditioning (RCIP), is universal as it is independent of the nature of the singularities. Strong right-hand-side singularities, such as $1/|r|^\alpha$ with α close to 1, can be treated in full machine precision. Adaptive refinement is used only in the recursive construction of the preconditioner, leading to an optimal number of discretization points and superior stability in the solve phase. The performance of the scheme is illustrated via several numerical examples, including an application to an integral equation derived from the linearized BGKW kinetic equation for the steady Couette flow.

© 2021 Elsevier Inc. All rights reserved.

1. Introduction

Fredholm second-kind integral equations (SKIEs) have become standard tools for solving boundary value problems of elliptic partial differential equations [9,23,26,30,32]. Advantages include dimensionality reduction in the solve phase, elimination of the need to impose artificial boundary conditions for exterior problems, easily achieved high-order discretization, and optimal complexity when coupled with fast algorithms such as the fast multipole method [14].

The present work is about the numerical solution to SKIEs of the form

$$(I + K)\rho(r) = f(r), \quad r \in \Gamma. \tag{1}$$

Here $r \in \mathbb{R}^2$ is a point in the plane; I is the identity operator; ρ is an unknown layer density to be solved for; Γ is a piecewise smooth closed contour (boundary) with a finite number of corners; K is an integral operator on Γ which is compact away from the corners; and f is a right-hand side which is singular at a finite number of boundary points, which may or may not coincide with the corner vertices, but otherwise smooth. The union of corner vertices and boundary points where f is singular is referred to as *singular points*. These points are denoted γ_j, $j = 1, 2, \ldots$. The kernel of K is denoted $K(r, r')$. We also assume a parameterization $r(s)$ of Γ where s is a parameter.

We shall construct an efficient scheme for the numerical solution of (1). The difficulty in this undertaking is that the singularities in f and the non-compactness of K at the γ_j may require a very large number of unknowns for the resolution

of ρ. This, in turn, may lead to high computing costs and also to artificial ill-conditioning and reduced achievable precision in quantities computed from ρ.

The nature of the singularity of f can be rather arbitrary in the applications we consider. There is no analysis of f involved in our work and neither is there any further analysis of K. We only assume that f can be evaluated everywhere at Γ except for at the γ_j. If, however, it is known whether the leading singular behavior of f is homogeneous on Γ in the shape of a wedge and whether K is scale invariant on such Γ, that information can be used to further improve the performance of our scheme.

We shall solve (1) numerically using Nyström discretization based on underlying composite 16-point Gauss-Legendre quadrature and the parameterization $r(s)$ and then accelerate and stabilize the solution process using an extended version of the recursively compressed inverse preconditioning (RCIP) method [16]. The discretization is chiefly done on a coarse mesh with quadrature panels of approximately equal size. The coarse quadrature panels are chosen so that the following holds: all singular points γ_j coincide with panel endpoints; for r on a panel close to γ_j and r' away from γ_j, $K(r(s), r')$ is smooth; for r away from γ_j and r' on a panel close to γ_j, $K(r, r'(s))$ is smooth.

The RCIP method assumes that the SKIE has a panelwise smooth right-hand side and consists of the following steps:

- Transform the SKIE at hand into a form where the layer density to be solved for is panelwise smooth.
- Use Nyström discretization to discretize the transformed SKIE on a grid on a fine mesh, obtained from the coarse mesh by repeated subdivision of the panels closest to each γ_j.
- Compress the transformed and discretized SKIE so that it can be solved on a grid on the coarse mesh without the loss of information. This involves the use of a forward recursion.
- Solve the compressed equation.
- Reconstruct the solution to the original SKIE from the solution to the compressed equation. This involves the use of a backward recursion.

In this paper, we extend the RCIP method to treat *singular* right-hand sides. The method is *universal* since algorithmic steps are completely independent of the nature of the singularities in the right-hand-side function. The only information needed is the locations of singularities. Very strong singularities can be treated in full machine precision. Indeed, we have studied singularities of the form $1/|r|^\alpha$ for a wide range of α and our method works very well even for $\alpha = 1 + 0.3i$.

We further observe that problems involving sources close to corners occur surprisingly often in computational electromagnetics and computational fluid dynamics. Examples include the determination of radiation patterns from 5G base stations placed at street corners [33] and singularity formation in Hele–Shaw flows driven by multipoles [31]. These problems involve *nearly singular* right-hand sides and can be treated easily by the method developed in this paper. The 5G base stations often have antennas with logarithmic singularities in their radiated field.

Examples of problems involving singular sources close to smooth surfaces can be found in the area of Internet of Things (IoT), where the radiation patterns from antennas, integrated in devices, again need to be found numerically for design purposes [10]. The most common types of IoT antennas are stripline and patch antennas. These have a radiating current density very close to a metallic ground plane, which often is a smooth part of the housing of the device. As IoT carrier frequencies get higher, antennas become smaller and the distance between the current and the housing surface shrinks – leading to increased need for computational resolution.

The paper is organized as follows. In Section 2, a new transformation is introduced to treat the singular right-hand side f. Sections 3, 4, 5, and 6 present detailed modifications to the other steps in the list that follow from the introduction of the new transform. Numerical examples are presented in Section 7. Finally, we discuss the application of our new method to the integral equation derived from the linearized Bhatnagar-Gross-Krook-Welander (BGKW) kinetic equation [5,35] for the steady Couette flow.

In order to keep the presentation concise, we concentrate on new development and refer the reader to the recently updated compendium [16] for details on the RCIP method. This compendium, in turn, contains references to original journal papers.

2. Transformed equation with smooth density

This section reviews the original RCIP transformation, discusses its shortcomings for singular right-hand sides f, and presents a new and better transformation. We frequently use the concept of *panelwise smooth functions*. By this we mean functions which can be well approximated by polynomials of degree 15 in s on individual quadrature panels. We also introduce the boundary subsets $\Gamma^{j\star}$, which refer to the four panels that are closest to a point γ_j (two on each side), and the boundary subsets $\Gamma^{j\star\star}$, which refer to the two panels that are closest to a point γ_j (one on each side).

2.1. The original transformation

The original RCIP transformation for SKIEs of the form (1) assumes that f is a panelwise smooth function and relies on a kernel split

$$K(r, r') = K^\star(r, r') + K^\circ(r, r'), \quad r, r' \in \Gamma, \tag{2}$$

and a corresponding operator split

$$K = K^\star + K^\circ. \tag{3}$$

In (3), the operator $K^\star$ denotes the part of K that accounts for self-interaction close to the corner vertices γ_j, and K° is the compact remainder. The split (2) is determined from a geometric criterion: if r and r' both are in $\Gamma^{j\star}$ for some j, then $K^\star(r, r') = K(r, r')$. Otherwise $K^\star(r, r')$ is zero.

The change of variables

$$\rho(r) = (I + K^\star)^{-1} \tilde{\rho}(r), \quad r \in \Gamma, \tag{4}$$

makes (1) with (3) assume the form

$$(I + K^\circ (I + K^\star)^{-1}) \tilde{\rho}(r) = f(r), \quad r \in \Gamma. \tag{5}$$

When f is panelwise smooth, the transformed layer density $\tilde{\rho}$ in (5) will, loosely speaking, also be panelwise smooth. This is so since the action of K° on any function results in a panelwise smooth function. In particular, $\tilde{\rho}$ will be smooth on $\Gamma^{j\star\star}$. The panelwise smoothness of $\tilde{\rho}$ on $\Gamma^{j\star\star}$, inherited from f, is the key property which makes the transformed equation (5) efficient for the original problem (1). The efficiency comes from the fact that a panelwise smooth unknown is easy to resolve by panelwise polynomials.

2.2. A new transformation

When the right-hand side f in (1) is not panelwise smooth, but has singularities at the corner vertices, the transformed layer density $\tilde{\rho}$ in (5) is not panelwise smooth either. In order to fix this problem we now propose a new transformation of (1) which, in addition to the split of K, also splits the right-hand side f and the unknown ρ as

$$f(r) = f^\star(r) + f^\circ(r), \quad r \in \Gamma, \tag{6}$$

$$\rho(r) = v(r) + g(r), \quad r \in \Gamma. \tag{7}$$

Here $f^\star(r) = f(r)$ if $r \in \Gamma^{j\star}$ for some j. Otherwise $f^\star(r)$ is zero. The functions v and g are given by

$$v(r) = (I + K^\star)^{-1} \tilde{v}(r), \tag{8}$$

$$g(r) = (I + K^\star)^{-1} f^\star(r), \tag{9}$$

where $\tilde{v}$ is a new unknown transformed layer density.

Use of (3), (6), (7), (8), and (9) makes (1) assume the form

$$(I + K^\circ (I + K^\star)^{-1}) \tilde{v}(r) = f^\circ(r) - K^\circ (I + K^\star)^{-1} f^\star(r), \quad r \in \Gamma. \tag{10}$$

One can see, in (10), that $\tilde{v}$ is panelwise smooth on $\Gamma^{j\star\star}$. This is so since the right-hand side of (10) is smooth on $\Gamma^{j\star}$.

We remark that if Γ is smooth so that there are no corners, only singularities in f, then the local transformation (8) is not needed and (10) reduces to

$$(I + K) v(r) = f^\circ(r) - K^\circ (I + K^\star)^{-1} f^\star(r), \quad r \in \Gamma. \tag{11}$$

One can also imagine mixed situations where (8) is used only at those γ_j which correspond to corner vertices.

3. Discretization of (5) and (10)

This section summarizes the modifications needed in the RCIP method in order for it to apply to (10) rather than to (5). We first show how RCIP is applied to (5), with detailed references to [16], and then state the modifications needed for (10). For simplicity of presentation it is from now on assumed that there is only one singular point, denoted γ, with an associated four-panel neighboring zone denoted $\Gamma^\star$ as illustrated in Fig. 1. We let $\Gamma^{\star\star}$ refer to the two panels closest to γ (one on each side).

The discretization of (5) takes place on two different meshes: on the coarse mesh and on a fine mesh. The fine mesh is constructed from the coarse mesh by n_{sub} times subdividing the panels closest to γ. See the right image of Fig. 1 for an example. Discretization points on the coarse mesh constitute the *coarse grid*. Points on the fine mesh constitute the *fine grid*. The number of refinement levels, n_{sub}, is chosen so that the operator $(I + K^\star)^{-1}$ in (5) is resolved to a desired precision.

As mentioned in Section 1, our Nyström discretization relies on composite 16-point Gauss–Legendre quadrature. This means that an integral
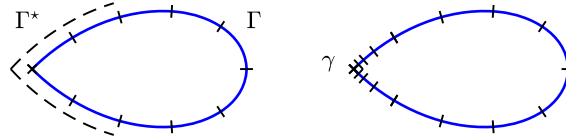
Fig. 1. A contour Γ with a corner at γ of opening angle $\theta = \pi/2$. Left: A coarse mesh with ten quadrature panels on Γ. A subset of Γ, called $\Gamma^\star$, covers the four coarse panels closest to γ. Right: A fine mesh created from the coarse mesh by subdividing the panels closest to γ a number $n_{sub} = 3$ of times.

$$\int_\Gamma h(r)\,d\ell\,,$$

where h is a smooth function and $d\ell$ is an element of arc length, can be approximated by a sum on the coarse grid

$$\int_\Gamma h(r)\,d\ell \approx \sum_j h(r(s_{coa_j}))|\dot{r}(s_{coa_j})|w_{coa_j}\,. \tag{12}$$

Here $\dot{r}(s) = dr(s)/ds$ denotes differentiation with respect to the boundary parameter s and w_{coa_j} are appropriately scaled Gauss–Legendre weights. A formula, analogous to (12) but with subscripts coa replaced with subscripts fin, holds when h is a singular function that needs the fine grid for resolution.

3.1. Smooth right-hand side

It is shown in [16, Appendix B] that the discretization of (5) on the fine grid followed by RCIP-style compression leads to the system

$$\left(\mathbf{I}_{coa} + \mathbf{K}^\circ_{coa}\mathbf{R}\right)\tilde{\boldsymbol{\rho}}_{coa} = \mathbf{f}_{coa}\,. \tag{13}$$

Here $\mathbf{R}$ is a block-diagonal matrix defined as [16, Eq. (26)]

$$\mathbf{R} = \mathbf{P}_W^T \left(\mathbf{I}_{fin} + \mathbf{K}^\star_{fin}\right)^{-1}\mathbf{P}\,, \tag{14}$$

and the subscripts coa and fin indicate what type of mesh is used for discretization. The prolongation matrix $\mathbf{P}$ interpolates piecewise polynomial functions known at the coarse grid to the fine grid. The weighted prolongation matrix $\mathbf{P}_W$ resembles $\mathbf{P}$, but is designed to act on discretized functions multiplied by quadrature weights [16, Eq. (21)]. The interpolation is done with respect to the boundary parameter s. The superscript T denotes the transpose.

3.2. Singular right-hand side

The discretization and compression of (10) can be constructed in a manner completely analogous to that of (5) and leads to the system

$$(\mathbf{I}_{coa} + \mathbf{K}^\circ_{coa}\mathbf{R})\tilde{\mathbf{v}}_{coa} = \mathbf{f}^\circ_{coa} - \mathbf{K}^\circ_{coa}\mathbf{R}_f\mathbf{f}^\star_{coa}\,, \tag{15}$$

where $\mathbf{R}$ is as in (14) and

$$\mathbf{R}_f = \mathbf{P}_W^T \left(\mathbf{I}_{fin} + \mathbf{K}^\star_{fin}\right)^{-1}\mathbf{P}_f\,. \tag{16}$$

The prolongation matrix $\mathbf{P}_f$ interpolates f from the coarse grid to the fine grid and can easily be constructed from computed values of f on the two grids respectively. The matrix $\mathbf{P}_f$ is block-diagonal with blocks being either identity matrices or, for entries corresponding to discretization points on $\Gamma^{\star\star}$, the rectangular rank-one matrix

$$\mathbf{P}_f^{\star\star} = \frac{1}{\mathbf{f}_{coa}^{\star\star H}\mathbf{f}_{coa}^{\star\star}}\mathbf{f}_{fin}^{\star\star}\mathbf{f}_{coa}^{\star\star H}\,. \tag{17}$$

Here $\mathbf{f}_{fin}^{\star\star}$ is a column vector with values of f on the fine grid on $\Gamma^{\star\star}$, $\mathbf{f}_{coa}^{\star\star}$ is a column vector with values of f on the coarse grid on $\Gamma^{\star\star}$, and H denotes the conjugate transpose. Clearly, $\mathbf{P}_f^{\star\star}\mathbf{f}_{coa}^{\star\star} = \mathbf{f}_{fin}^{\star\star}$.

4. Forward recursion for R and R_f

The matrices $\mathbf{R}$ and $\mathbf{R}_f$ of (14) and (16) differ from the identity matrix only in that they each contain a non-trivial 64×64 diagonal block, associated with entries corresponding to discretization points on $\Gamma^\star$. These diagonal blocks can be efficiently constructed via recursions relying on the discretization of K on small meshes on a hierarchy of boundary subsets $\Gamma_i^\star$ around γ. Central to the recursions are two types of overlapping meshes called type b and type c. The type b mesh
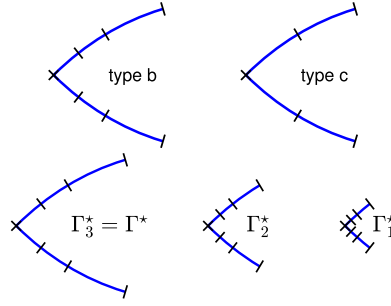
Fig. 2. Top row: meshes of type b and type c on the boundary subset $\Gamma^\star$. The type b mesh has six panels. The type c mesh has four panels. Bottom row: the boundary subsets $\Gamma_3^\star = \Gamma^\star$, $\Gamma_2^\star$, and $\Gamma_1^\star$ along with their corresponding type b meshes for $n_{\text{sub}} = 3$.

contains six quadrature panels. The type c mesh contains four quadrature panels. See Fig. 2 for an illustration and [16, Section 7] for more information.

The recursions for $\mathbf{R}$ and $\mathbf{R}_f$, given below, use local prolongation matrices $\mathbf{P}_{\text{bc}}$, $\mathbf{P}_{W\text{bc}}$, and $\mathbf{P}_{f i \text{bc}}$ similar to the global matrices $\mathbf{P}$, $\mathbf{P}_W$ and $\mathbf{P}_f$ of Section 3. The matrix $\mathbf{P}_{\text{bc}}$ performs polynomial interpolation from a grid on a type c mesh to a grid on a type b mesh on the same level. The matrix $\mathbf{P}_{f i \text{bc}}$ interpolates f from a grid on a type c mesh to a grid on a type b mesh on the same level. The subscript i indicates that $\mathbf{P}_{f i \text{bc}}$ depends on the hierarchical level in which it appears.

The matrices $\mathbf{P}_{\text{bc}}$ and $\mathbf{P}_{W\text{bc}}$ are level independent. Their construction is described in [16, Section 7.1]. The matrix $\mathbf{P}_{f i \text{bc}}$ is constructed analogously to the matrix $\mathbf{P}_f$ of Section 3.

4.1. The recursion for $\mathbf{R}$

It is shown in [16, Appendix D] that the non-trivial diagonal 64×64 block of $\mathbf{R}$ can be obtained from the recursion

$$\mathbf{R}_i = \mathbf{P}_{W\text{bc}}^T \left(\mathbb{F}\{\mathbf{R}_{i-1}^{-1}\} + \mathbf{I}_b^\circ + \mathbf{K}_{ib}^\circ \right)^{-1} \mathbf{P}_{\text{bc}}, \quad i = 1, \ldots, n_{\text{sub}}, \tag{18}$$

with the initializer

$$\mathbb{F}\{\mathbf{R}_0^{-1}\} = \mathbf{I}_b^\star + \mathbf{K}_{1b}^\star. \tag{19}$$

Here $\mathbf{K}_{ib}$ is the discretization of K on a type b mesh on level i in the hierarchy of local meshes around γ. The operator $\mathbb{F}\{\cdot\}$ expands its matrix argument by zero-padding (adding a frame of zeros of width 16 around it). The superscripts $\star$ and $\circ$ denote matrix splits analogous to the split (3).

The forward recursion (18) starts at the finest refinement level, $i = 1$, and ascends through the hierarchy of levels until it reaches the coarsest level $i = n_{\text{sub}}$. The matrix $\mathbf{R}_{n_{\text{sub}}}$ is equal to the non-trivial 64×64 block of $\mathbf{R}$.

4.2. The recursion for $\mathbf{R}_f$

A recursion for the non-trivial 64×64 block of $\mathbf{R}_f$ can be derived in complete analogy with the derivation of (18). See Appendix A for the key steps of the derivation. The result is

$$\mathbf{R}_{fi} = \mathbf{P}_{W\text{bc}}^T \left(\mathbb{F}\{\mathbf{R}_{i-1}^{-1}\} + \mathbf{I}_b^\circ + \mathbf{K}_{ib}^\circ \right)^{-1} \left(\mathbb{F}\{\mathbf{R}_{i-1}^{-1} \mathbf{R}_{f(i-1)}\} + \mathbf{I}_b^\circ \right) \mathbf{P}_{f i \text{bc}}, \quad i = 1, \ldots, n_{\text{sub}}. \tag{20}$$

The recursion (20) can be initialized with

$$\mathbf{R}_{f0} = \mathbf{R}_0 \tag{21}$$

and run in tandem with (18).

4.3. Further improvement of the recursions

From (15) it is evident that the matrix $\mathbf{R}_f$ acts only on one particular known vector, namely on $\mathbf{f}_{\text{coa}}^\star$. Therefore, rather than first finding $\mathbf{R}_f$ via (20) and then computing the vector

$$\mathbf{r}_f^\star = \mathbf{R}_f \mathbf{f}_{\text{coa}}^\star, \tag{22}$$

one can instead modify (20) with (21) so that it produces $\mathbf{r}_f^\star$ directly

$$\mathbf{r}_{fi}^\star = \mathbf{P}_{W\text{bc}}^T \left(\mathbb{F}\{\mathbf{R}_{i-1}^{-1}\} + \mathbf{I}_b^\circ + \mathbf{K}_{ib}^\circ \right)^{-1} \left(\mathbb{F}\{\mathbf{R}_{i-1}^{-1} \mathbf{r}_{f(i-1)}^\star\} + \mathbf{f}_{ib}^\circ \right), \quad i = 1, \ldots, n_{\text{sub}}, \tag{23}$$

$$\mathbf{r}_{f0}^{\star} = \mathbf{R}_{f0}\mathbf{f}_{1b}^{\star}. \tag{24}$$

The recursion (23) with (24) is a bit faster than (20) with (21).

It is also worth noting that all three recursions (18), (20), and (23) can be implemented without the explicit inversion of $\mathbf{R}_{i-1}$. The key to this is to use the Schur–Banachiewicz inverse formula for partitioned matrices [21, Eq. (8)]. Avoiding inversion is particularly important when $\mathbf{R}_{i-1}$ is ill conditioned. Details on an implementation for (18), free from inversion of $\mathbf{R}_{i-1}$, are given in [16, Section 8].

4.4. Efficient initializers

A minor problem with the recursions (18), (20), and (23) is that it could be difficult to determine a suitable recursion length n_{sub} a priori. A too large n_{sub} leads to unnecessary work. A too small n_{sub} fails to resolve the problem. Should it, however, happen that K is scale-invariant on wedges, there may be an easy way around this problem. The key observation is that for large n_{sub} and at levels i such that $n_{\text{sub}} - i \gg 1$, the matrices $\mathbf{K}_{ib}^{\circ}$ have often converged to a matrix $\mathbf{K}_{b}^{\circ}$ (in double precision arithmetic) that is independent of i. This, typically, happens for $n_{\text{sub}} - i > 60$ and means that (18) assumes the form of a fixed-point iteration

$$\mathbf{R}_{*i} = \mathbf{P}_{W\text{bc}}^{T}\left(\mathbb{F}\{\mathbf{R}_{*(i-1)}^{-1}\} + \mathbf{I}_{b}^{\circ} + \mathbf{K}_{b}^{\circ}\right)^{-1}\mathbf{P}_{\text{bc}}, \quad i = 1, 2, \dots. \tag{25}$$

It also means that all $\mathbf{R}_{i}$ in (18) are the same for $n_{\text{sub}} - i \gg 1$. In view of the above one can replace the initializer $\mathbf{R}_{0}$ of (19) with the fixed-point matrix $\mathbf{R}_{*}$ obtained by running (25) until convergence. With the choice $\mathbf{R}_{0} = \mathbf{R}_{*}$, it is enough to take $n_{\text{sub}} = 60$ steps in the recursion (18). See, further, the discussion in [16, Sections 12–13].

In a procedure similar to that just described, it is also possible to replace the initializers $\mathbf{R}_{f0}$ and $\mathbf{r}_{f0}^{\star}$ of (21) and (24) with more efficient initializers. The requirements are, in addition to that K is scale invariant on wedges, that the leading singular behavior of f is homogeneous on wedges. If this holds, and if $n_{\text{sub}} - i \gg 1$, then (20) assumes the form of a linear fixed-point iteration

$$\mathbf{R}_{f*i} = \mathbf{P}_{W\text{bc}}^{T}\left(\mathbb{F}\{\mathbf{R}_{*}^{-1}\} + \mathbf{I}_{b}^{\circ} + \mathbf{K}_{b}^{\circ}\right)^{-1}\left(\mathbb{F}\{\mathbf{R}_{*}^{-1}\mathbf{R}_{f*(i-1)}\} + \mathbf{I}_{b}^{\circ}\right)\mathbf{P}_{f\text{bc}}, \quad i = 1, 2, \dots. \tag{26}$$

The fixed-point matrix $\mathbf{R}_{f*}$ can be found with direct methods solving a Sylvester equation. With the choices $\mathbf{R}_{f0} = \mathbf{R}_{f*}$ and $\mathbf{r}_{f0}^{\star} = \mathbf{R}_{f*}\mathbf{f}_{1b}^{\star}$ it is enough to take $n_{\text{sub}} = 60$ steps in the recursions (20) and (23).

We remark that efficient initializers can be found also under more general conditions on K than scale invariant on wedges. See [17, Section 5.3], for an example.

5. Computing integrals of ρ

Often, in applications, one is not primarily interested in the solution ρ to (1) in itself. Rather, one is interested in computing functionals of ρ of the type

$$q = \int_{\Gamma} h(r)\rho(r)\,\mathrm{d}\ell, \tag{27}$$

where $h(r)$ is a smooth function. The RCIP method offers an elegant way to do this that only involves quantities appearing in the compressed equations (13) and (15).

Assume that f is smooth and consider (13). The quantity q of (27) can then be well approximated by the sum on the coarse grid

$$q \approx \sum_{j} h(r(s_{\text{coa}_{j}}))\hat{\rho}_{\text{coa}_{j}}|\dot{r}(s_{\text{coa}_{j}})|w_{\text{coa}_{j}}, \tag{28}$$

where $\hat{\rho}_{\text{coa}_{j}}$ are elements of the *weight-corrected* density vector

$$\hat{\boldsymbol{\rho}}_{\text{coa}} = \mathbf{R}\tilde{\boldsymbol{\rho}}_{\text{coa}}. \tag{29}$$

See [16, Appendix C] for a proof.

The situation for a singular f and (15) is completely analogous. The expression (28) holds with (29) replaced by

$$\hat{\boldsymbol{\rho}}_{\text{coa}} = \mathbf{R}\tilde{\mathbf{v}}_{\text{coa}} + \mathbf{r}_{f}^{\star}. \tag{30}$$

6. Backward recursion for the reconstruction of ρ

When the compressed equation (15) has been solved for $\tilde{\mathbf{v}}_{\text{coa}}$ one might also be interested in the reconstruction of the discretized solution

$$\boldsymbol{\rho}_{\text{fin}} = \mathbf{v}_{\text{fin}} + \mathbf{g}_{\text{fin}} \tag{31}$$

to the original SKIE (1), compare (7). Such a reconstruction can be achieved by, loosely speaking, running the recursions (18) and (20) backward on $\Gamma^\star$. Outside of $\Gamma^\star$, the coarse grid and the fine grid coincide and it holds that $\boldsymbol{\rho} = \mathbf{v} = \tilde{\mathbf{v}}$, see (7), (8), and (9).

6.1. The recursion for $\mathbf{v}_{\text{fin}}$

We first review the reconstruction of $\boldsymbol{\rho}_{\text{fin}}$ from the solution $\tilde{\boldsymbol{\rho}}_{\text{coa}}$ to (13). The mechanism for this reconstruction was originally derived in [15, Section 7] and is also summarized in [16, Section 10]. The backward recursion reads

$$\vec{\boldsymbol{\rho}}_{\text{coa},i} = \left[\mathbf{I}_{\text{b}} - \mathbf{K}_{ib}^\circ \left(\mathbb{F}\{\mathbf{R}_{i-1}^{-1}\} + \mathbf{I}_{\text{b}}^\circ + \mathbf{K}_{ib}^\circ \right)^{-1} \right] \mathbf{P}_{\text{bc}} \tilde{\boldsymbol{\rho}}_{\text{coa},i}, \quad i = n_{\text{sub}}, \dots, 1. \tag{32}$$

Here $\tilde{\boldsymbol{\rho}}_{\text{coa},i}$ is a column vector with 64 elements. In particular, $\tilde{\boldsymbol{\rho}}_{\text{coa},n_{\text{sub}}}$ is the restriction of $\tilde{\boldsymbol{\rho}}_{\text{coa}}$ to $\Gamma^\star$, while $\tilde{\boldsymbol{\rho}}_{\text{coa},i}$ are taken as elements $\{17 : 80\}$ of $\vec{\boldsymbol{\rho}}_{\text{coa},i+1}$ for $i < n_{\text{sub}}$. The elements $\{1 : 16\}$ and $\{81 : 96\}$ of $\vec{\boldsymbol{\rho}}_{\text{coa},i}$ are the reconstructed values of $\boldsymbol{\rho}_{\text{fin}}$ on the outermost panels of a type b mesh on $\Gamma_i^\star$.

When the recursion is completed, there are no values assigned to $\boldsymbol{\rho}_{\text{fin}}$ at points on the four innermost panels (on $\Gamma_1^\star$ closest to γ) on the fine grid. Reconstructed weight-corrected values of $\boldsymbol{\rho}_{\text{fin}}$ on these panels can then be used, rather than true values, and are obtained from

$$\mathbf{R}_0 \tilde{\boldsymbol{\rho}}_{\text{coa},0} . \tag{33}$$

We now observe that the reconstruction of $\mathbf{v}_{\text{fin}}$ from the solution $\tilde{\mathbf{v}}_{\text{coa}}$ to (15) is identical to the reconstruction just described. This is so since both $\tilde{\rho}$ of (5) and $\tilde{v}$ of (10) are panelwise smooth functions.

6.2. The recursion for $\mathbf{g}_{\text{fin}}$

The backward recursion for $\mathbf{g}_{\text{fin}}$ from $\mathbf{f}_{\text{coa}}^\star$ is analogous to (32), but the vector $\vec{\mathbf{g}}_{\text{coa},i}$, corresponding to $\vec{\boldsymbol{\rho}}_{\text{coa},i}$ in (32), needs a split in a singular and a panelwise smooth part on a type b mesh on each $\Gamma_i^\star$

$$\vec{\mathbf{g}}_{\text{coa},i} = \vec{\mathbf{g}}_{\text{coa},i}^{\text{smo}} + \mathbf{f}_{ib} . \tag{34}$$

The backward recursion can then be written

$$\vec{\mathbf{g}}_{\text{coa},i}^{\text{smo}} = \left[\mathbf{I}_{\text{b}} - \mathbf{K}_{ib}^\circ \left(\mathbb{F}\{\mathbf{R}_{i-1}^{-1}\} + \mathbf{I}_{\text{b}}^\circ + \mathbf{K}_{ib}^\circ \right)^{-1} \right] \mathbf{P}_{\text{bc}} \tilde{\mathbf{g}}_{\text{coa},i}^{\text{smo}}$$

$$- \mathbf{K}_{ib}^\circ \left(\mathbb{F}\{\mathbf{R}_{i-1}^{-1}\} + \mathbf{I}_{\text{b}}^\circ + \mathbf{K}_{ib}^\circ \right)^{-1} \left(\mathbb{F}\{\mathbf{R}_{i-1}^{-1}\mathbf{r}_{f(i-1)}^\star\} + \mathbf{f}_{ib}^\circ \right), \quad i = n_{\text{sub}}, \dots, 1. \tag{35}$$

Here $\tilde{\mathbf{g}}_{\text{coa},i}^{\text{smo}}$ is a column vector with 64 elements. In particular, $\tilde{\mathbf{g}}_{\text{coa},n_{\text{sub}}}^{\text{smo}} = \mathbf{0}$ while $\tilde{\mathbf{g}}_{\text{coa},i}^{\text{smo}}$ are taken as elements $\{17 : 80\}$ of $\vec{\mathbf{g}}_{\text{coa},i+1}$ for $i < n_{\text{sub}}$. The elements $\{1 : 16\}$ and $\{81 : 96\}$ of $\vec{\mathbf{g}}_{\text{coa},i}$ in (34) are the reconstructed values of $\mathbf{g}_{\text{fin}}$ on the outermost panels of a type b mesh on $\Gamma_i^\star$. The reconstructed weight-corrected values of $\mathbf{g}_{\text{fin}}$ on the four innermost panels on the fine grid are obtained from

$$\mathbf{R}_0 \tilde{\mathbf{g}}_{\text{coa},0}^{\text{smo}} + \mathbf{r}_{f0}^\star . \tag{36}$$

7. Numerical examples

We now demonstrate the efficiency of our numerical scheme for (1). The scheme consists of the compressed equation (15), the recursions (18), (23), (32), and (35), and initializers obtained from (25) and (26). The code is implemented in MATLAB, release 2020b, and executed on a 64 bit Linux laptop with a 2.10 GHz Intel i7-4600U CPU. The implementations are standard and rely on built-in functions such as `dlyap` (SLICOT subroutine SB04QD), for the Sylvester equation. Large linear systems are solved using GMRES, incorporating a low-threshold stagnation avoiding technique [19, Section 8] applicable to systems coming from discretizations of SKIEs. The GMRES stopping criterion is set to machine epsilon in the estimated relative residual.

7.1. A transmission problem for Laplace's equation

7.1.1. Test equation and test geometry

We solve the SKIE (1) with an integral operator K defined by its action on ρ as

$$K\rho(r) = 2\lambda \int_\Gamma \frac{\partial G}{\partial v}(r, r')\rho(r')\,d\ell'\,, \quad r \in \Gamma\,. \tag{37}$$

Here λ is a parameter set to $\lambda = 0.5$, $v(r)$ is the exterior unit normal at position r on Γ, $\partial/\partial v = v(r) \cdot \nabla$, $G(r, r')$ is the fundamental solution to Laplace's equation in the plane

$$G(r, r') = -\frac{1}{2\pi} \log |r - r'|\,, \tag{38}$$

and Γ is the closed contour with a corner at $\gamma = 0$ parameterized as

$$r(s) = \sin(\pi s)\,(\cos((s - 0.5)\theta), \sin((s - 0.5)\theta))\,, \quad s \in [0, 1]\,, \tag{39}$$

where θ corresponds to the opening angle of the corner. With the particular choice (37) for K, the SKIE (1) can model a transmission problem for Laplace's equation [16, Section 4].

In our experiments we shall use a coarse mesh that is sufficiently refined as to resolve (1) away from γ, vary the number of distinct recursion steps n_{sub}, and monitor the convergence of the scalar quantity q of (27) with $h(r) = 1$. For comparison we compute q in two ways: first on the coarse grid via (28) and with $\hat{\boldsymbol{\rho}}_{\text{coa}}$ from (30), then on the fine grid via

$$q \approx \sum_j \rho_{\text{fin}_j} |\dot{r}(s_{\text{fin}_j})| w_{\text{fin}_j} \tag{40}$$

and with $\boldsymbol{\rho}_{\text{fin}}$ from (31).

7.1.2. Example with an analytical solution

We start with an example where the corner opening angle is set to $\theta = \pi$. Then Γ of (39) becomes a circle with a circumference of π. The right-hand side of (1) is set to

$$f(r) = \frac{1}{\ell^\alpha(r)} + \frac{1}{(\pi - \ell(r))^\alpha}\,, \quad r \in \Gamma\,. \tag{41}$$

Here α is a possibly complex parameter with $\Re\{\alpha\} < 1$, controlling the strength of the singularity at γ, and $\ell(r)$ is the distance from γ to r measured in arc length along Γ as the circle is traversed in a counterclockwise fashion. The solution ρ to (1) can in this example be computed analytically

$$\rho(r) = f(r) + \frac{2\lambda\pi^{-\alpha}}{(1 - \alpha)(1 - \lambda)}\,, \quad r \in \Gamma\,, \tag{42}$$

as can the quantity q:

$$q = \frac{2\pi^{(1-\alpha)}}{(1 - \alpha)(1 - \lambda)}\,. \tag{43}$$

Note that $\rho(r)$ of (42) diverges everywhere on Γ as $\alpha \to 1^-$, so there is no solution for $\alpha = 1$. Neither is there a finite limit value of q. If $\Im\{\alpha\} \neq 0$ is fixed, however, then there is a limit solution $\rho(r)$ and a finite limit value of q as $\Re\{\alpha\} \to 1^-$.

Fig. 3 shows results obtained with 10 quadrature panels on the coarse mesh on Γ, corresponding to 160 discretization points on the coarse grid, and for various singularity strengths α. For $\alpha < 1$ and not too close to 1, the results produced by our scheme are essentially fully accurate. Values of q computed on the coarse grid (q_{coa}) and on the fine grid (q_{fin}) agree completely – indicating that the reconstruction procedure of Section 6 is stable.

Note, in Fig. 3(d), that for $\alpha = 1 + 0.3i$, which corresponds to a singular right-hand side f that is not even in L^1, the scheme loses only about two digits of accuracy. Note also that, thanks to the use of initializers, a number $n_{\text{sub}} = 60$ of distinct recursion steps is more than enough to make q converge (to the achievable precision) for all values of α tested in Fig. 3.

7.1.3. A one-corner example

We now consider an example where the corner opening angle in (39) is set to $\theta = \pi/2$. The contour Γ then assumes the shape shown in Fig. 1. The right-hand side of (1) is set to

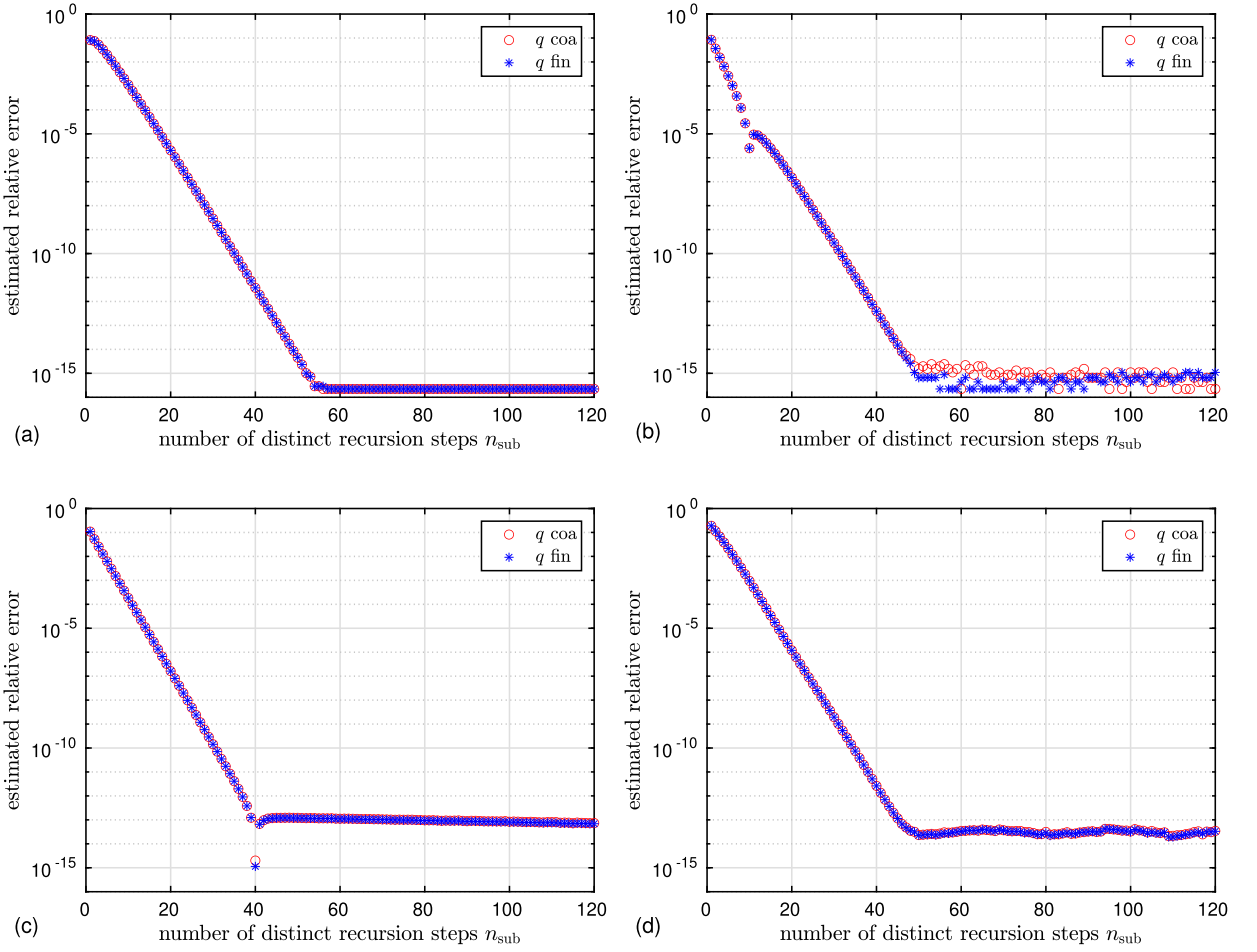$$f(r) = \frac{1}{|r|^\alpha} + \log |r|\,, \quad r \in \Gamma\,. \tag{44}$$

Fig. 3. Convergence of q with the number of distinct recursion levels n_{sub} in the example of Section 7.1.2. The singularity strengths α of (41) are taken as: (a) $\alpha = 0.5$; (b) $\alpha = 0.94$; (c) $\alpha = 0.99$; (d) $\alpha = 1 + 0.3i$.

Fig. 4, analogous to Fig. 3, shows results obtained with 160 discretization points on the coarse grid on Γ and for various singularity strengths α. In the absence of an analytical expression for q we use, as a reference, the value of q_{coa} obtained with $n_{sub} = 500$. For example, with $\alpha = 0.94$ this value is $q \approx 63.53529437281905$ and could be compared to the value $q \approx 63.53529437281894$, obtained with 29,088 discretization points on the fine grid on Γ using the L^1-norm-preserving Nyström discretization of Askham and Greengard [4, Section 4], which extends the L^2-inner-product-preserving discretization of Bremer [6, Section 2], in combination with compensated summation [22,25]. Since the relative difference between these two reference values is on the order of our own error estimate in Fig. 4(b), we believe that our error estimates are reliable also in Fig. 4(c,d), where the norm-preserving discretization cannot be used due to memory constraints.

We add that the execution of our code is very rapid. The computing time, per data point, in Fig. 4 varies with n_{sub} and also differs between q_{coa} and q_{fin}, but it is much less than a second for all the data points shown.

7.2. The exterior Dirichlet Helmholtz problem

The exterior Dirichlet problem for the Helmholtz equation has been considered in [16, Section 18] in detail. Here we use this problem to illustrate that its solution $U(r)$ can be found accurately in the entire computational domain also when the right-hand side $f(r)$ is singular or nearly singular. The integral representation of $U(r)$ and the resulting boundary integral equation (a combined field integral equation) are identical to those in [16, Section 18]. The boundary Γ is given by (39) with $\theta = \pi/2$. We consider two cases

(a) *Singular right-hand side*

The exact solution is

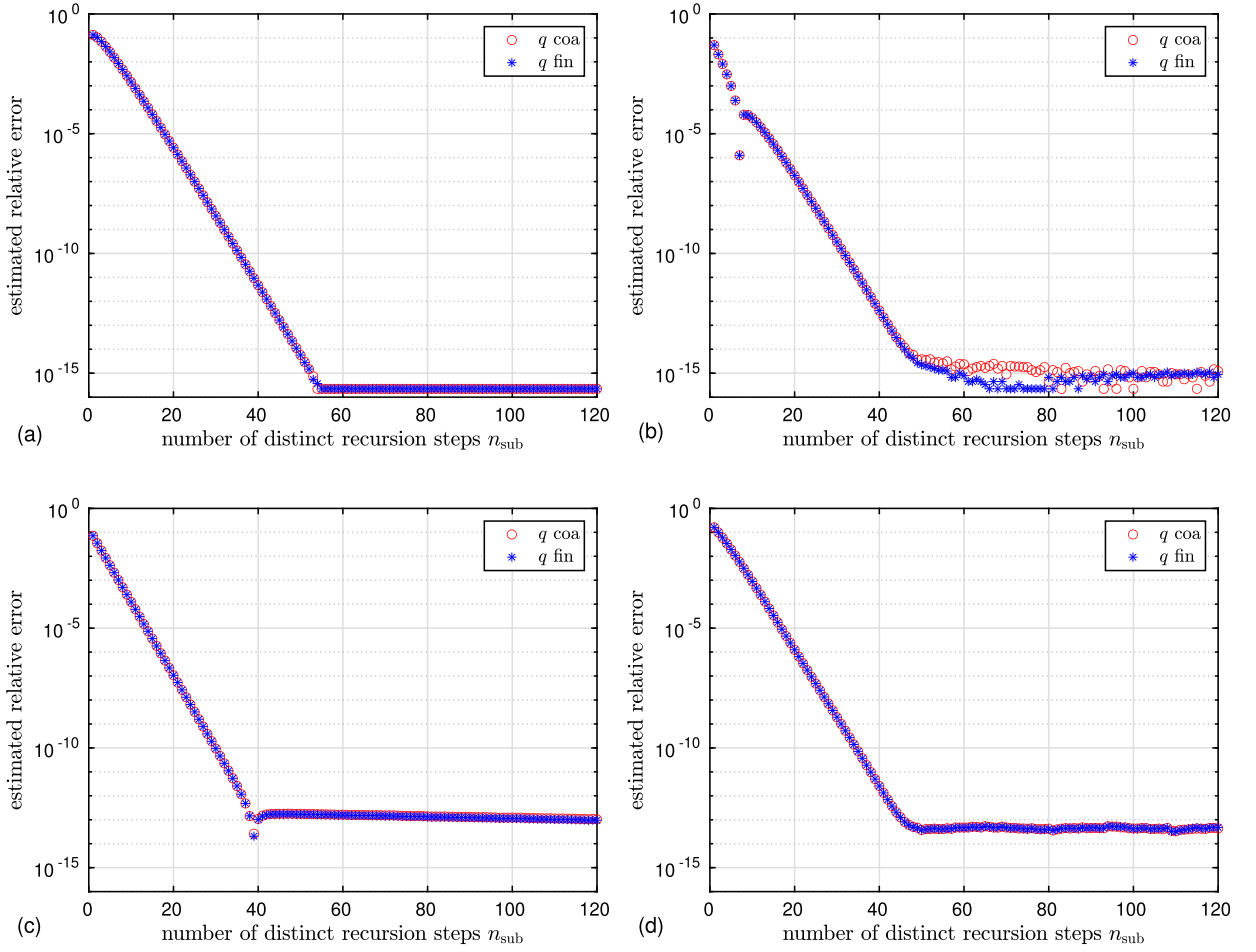$$U(r) = H_0^{(1)}(\omega|r|) \tag{45}$$

Fig. 4. Convergence of q with the number of distinct recursion levels n_{sub} in the example of Section 7.1.3. The singularity strengths α of (44) are taken as: (a) $\alpha = 0.5$; (b) $\alpha = 0.94$; (c) $\alpha = 0.99$; (d) $\alpha = 1 + 0.3i$.

with $f(r)$ being the restriction of $U(r)$ to Γ. Here ω is the wavenumber and $H_0^{(1)}$ is the first-kind Hankel function of order zero, that is, the field is generated by an acoustic monopole right at the corner.

(b) *Nearly singular right-hand side*

The exact solution is

$$U(r) = H_1^{(1)}(\omega|r - r'|)\frac{x - x'}{|r - r'|}, \tag{46}$$

with $f(r)$ being the restriction of $U(r)$ to Γ, and $r' = (10^{-10}, 0)$. That is, the field is generated by an acoustic dipole very close to the corner.

The integral equation [16, Eq. (68)] is solved with $\omega = 10$ using the method in this paper, the singular point is taken as $\gamma = 0$ both for (a) and for (b). For both cases, the coarse grid on Γ has 56 panels, i.e., 896 discretization points; the number of refinement levels is set to $n_{sub} = 112$; and the number of GMRES iterations needed is 18. The field is evaluated using the scheme in [16, Section 20]. The solve phase takes about 4 seconds, while the field evaluation takes about 100 seconds, which can be accelerated via the fast multipole method [8] when necessary. A Cartesian grid of 200×200 equispaced points is placed on the rectangle $[-0.1, 1.1] \times [-0.53, 0.53]$ and evaluations are carried out at those 27,760 grid points that are in the exterior domain. For case (a), 5,088 target points activate local panelwise evaluation for close panels; 920 target points close to the corner vertex require that the solution ρ_{fin} on the fine grid is reconstructed using the backward recursion in Section 6. For case (b), the numbers are 6,364 and 1,326, respectively.

Fig. 5 shows the absolute error in the numerical solution, demonstrating that our scheme achieves high accuracy in the entire computational domain for both *singular* and *nearly singular* right-hand sides. The small difference in achievable accuracy between (a) and (b) can chiefly be explained by that (b) has a stronger singularity in the right-hand side $f(r)$ than has (a) and is, thus, harder to resolve.
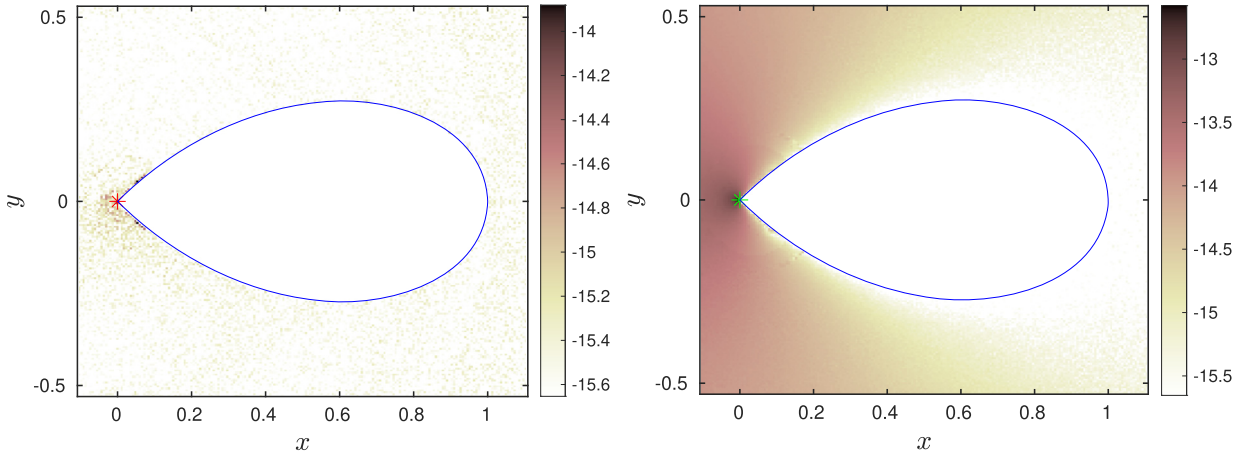
Fig. 5. $\log_{10}$ of absolute error in the solution $U(r)$ to the exterior Dirichlet Helmholtz problem. The blue curve is the boundary Γ. Left: $U(r)$ is an acoustic monopole field with the source at $r' = (0, 0)$ shown as a red star. Right: $U(r)$ is an acoustic dipole field with the source at $r' = (10^{-10}, 0)$ shown as a green star. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

8. Application to the linearized BGKW equation for the Couette flow

In this section, we revisit the integral equation that is derived from the linearized BGKW equation for the steady Couette flow [7,24,27]

$$u(x) - \frac{1}{k\sqrt{\pi}} \int_{-0.5}^{0.5} J_{-1}\left(\frac{|x - y|}{k}\right) u(y)\, dy = f(x), \quad x \in [-0.5, 0.5], \tag{47a}$$

$$f(x) = \frac{1}{2\sqrt{\pi}} \left[J_0\left(\frac{0.5 - x}{k}\right) - J_0\left(\frac{0.5 + x}{k}\right) \right], \tag{47b}$$

where the parameter k is the Knudsen number and J_n is the nth order Abramowitz function defined by

$$J_n(x) = \int_0^\infty t^n e^{-t^2 - x/t}\, dt, \quad n \geq -1. \tag{48}$$

See, for example, [13] and references therein for the properties of Abramowitz functions and an accurate numerical scheme for their evaluation. The BGKW equation has been studied in [24], where it is shown that the solution of (47a) contains singular terms $(x \ln x)^n$ for $n \in \mathbb{N} := \{1, 2, \ldots\}$ at the endpoints. It is known that the kernel function $J_{-1}(x)$ has both absolute value and logarithmic singularities at $x = 0$, and that the right-hand-side function (47b) has $x \ln x$ singularity at the endpoints. Benchmark calculations have been carried out in [24], using dyadic refinement towards the endpoints to treat the singularities of the solution and the right-hand-side function, and generalized Gaussian quadrature [28] to treat the kernel singularities.

We have implemented the method of this paper, based on (10), to solve (47a). We note that both endpoints are singular points and there is only one side to each singular point, since we are dealing with an open arc instead of a closed contour. This leads to straightforward modifications in the method and its implementation. Some implementation details are as follows. First, the kernel-split quadrature is applied to treat the kernel singularity: the correction to the logarithmic singularity was done in [15]; the correction on diagonal blocks for the absolute value singularity can be derived easily in an identical manner; the splitting of the kernel into various parts is done using either the series expansion of the Abramowitz function [1] or the Chebyshev expansion for each part as in [29]. When the Knudsen number is low, the kernel is sharply peaked at the origin. A modified version of the upsampling scheme in [2] is used to resolve the sharp peak of the kernel accurately and efficiently so that the coarse panels only need to resolve the features of the solution. Here the modification is that the exact centering in [2] is not enforced for local adaptive panels for each target. Instead, the so-called level-restricted property [11], i.e., sizes of adjacent panels can differ at most by a factor of 2, is used to ensure the accuracy in the calculation of the integrals. We remark that the cost of the upsampling scheme is $O(\log(1/k))$ as $k \to 0$, the same as that in [2].

Second, it is observed numerically that the condition number of the integral equation (47a) increases as k decreases, reaching about $2 \cdot 10^4$ for $k = 0.003$. On the other hand, the solution $u(x)$ approaches the asymptotic solution $u_{\text{asym}}(x) = x$ as $k \to 0$. Thus, in order to reduce the effect of the ill-conditioning of the integral equation for small values of k, we write $u(x) = w(x) + x$ and solve the following equation for $w(x)$ instead when $k \leq 0.3$:

Table 1
The velocity u at $x = 0.5$ at sample values of the Knudsen number k. Note that in [24], the interval is shifted from $[-0.5, 0.5]$ to $[0, 1]$. Thus, the velocity u at $x = 0.5$ is listed as u at $x = 1$ in Tables 1 and 2 in [24].

k	$u(0.5)$ in [24]	$u(0.5)$ (current)	Error
0.003	$4.978915352789693 \cdot 10^{-1}$	$4.978915352789726 \cdot 10^{-1}$	$6.6 \cdot 10^{-15}$
0.01	$4.930697807742208 \cdot 10^{-1}$	$4.930697807742217 \cdot 10^{-1}$	$1.8 \cdot 10^{-15}$
0.03	$4.800058682766829 \cdot 10^{-1}$	$4.800058682766837 \cdot 10^{-1}$	$1.6 \cdot 10^{-15}$
0.1	$4.412246409722421 \cdot 10^{-1}$	$4.412246409722424 \cdot 10^{-1}$	$6.3 \cdot 10^{-16}$
0.3	$3.672125695500504 \cdot 10^{-1}$	$3.672125695500499 \cdot 10^{-1}$	$1.4 \cdot 10^{-15}$
1.0	$2.518613399894732 \cdot 10^{-1}$	$2.518613399894736 \cdot 10^{-1}$	$1.5 \cdot 10^{-15}$
2.0	$1.852462993740218 \cdot 10^{-1}$	$1.852462993740218 \cdot 10^{-1}$	$1.5 \cdot 10^{-16}$
3.0	$1.504282444992075 \cdot 10^{-1}$	$1.504282444992074 \cdot 10^{-1}$	$3.7 \cdot 10^{-16}$
5.0	$1.126351880294592 \cdot 10^{-1}$	$1.126351880294592 \cdot 10^{-1}$	$2.5 \cdot 10^{-16}$
7.0	$9.171689613521435 \cdot 10^{-2}$	$9.171689613521428 \cdot 10^{-2}$	$7.6 \cdot 10^{-16}$
10.0	$7.292211299328497 \cdot 10^{-2}$	$7.292211299328491 \cdot 10^{-2}$	$7.6 \cdot 10^{-16}$
30.0	$3.381357342231838 \cdot 10^{-2}$	$3.381357342231840 \cdot 10^{-2}$	$6.2 \cdot 10^{-16}$
100.0	$1.343072948081877 \cdot 10^{-2}$	$1.343072948081874 \cdot 10^{-2}$	$1.9 \cdot 10^{-15}$

Table 2
The stress P_{xy} at sample values of the Knudsen number k.

k	P_{xy} in [24]	P_{xy} (current)	Error
0.003	$-1.490909702131201 \cdot 10^{-3}$	$-1.490909702131188 \cdot 10^{-3}$	$8.7 \cdot 10^{-15}$
0.01	$-4.900405009657547 \cdot 10^{-3}$	$-4.900405009657524 \cdot 10^{-3}$	$4.8 \cdot 10^{-15}$
0.03	$-1.413798601526842 \cdot 10^{-2}$	$-1.413798601526841 \cdot 10^{-2}$	$4.9 \cdot 10^{-16}$
0.1	$-4.155607782558620 \cdot 10^{-2}$	$-4.155607782558619 \cdot 10^{-2}$	$3.3 \cdot 10^{-16}$
0.3	$-9.344983511356682 \cdot 10^{-2}$	$-9.344983511356685 \cdot 10^{-2}$	$3.0 \cdot 10^{-16}$
1.0	$-1.694625753368226 \cdot 10^{-1}$	$-1.694625753368225 \cdot 10^{-1}$	$3.3 \cdot 10^{-16}$
2.0	$-2.083322536749375 \cdot 10^{-1}$	$-2.083322536749375 \cdot 10^{-1}$	0.0
3.0	$-2.266437497658084 \cdot 10^{-1}$	$-2.266437497658084 \cdot 10^{-1}$	0.0
5.0	$-2.446632678455994 \cdot 10^{-1}$	$-2.446632678455994 \cdot 10^{-1}$	0.0
7.0	$-2.536943539674479 \cdot 10^{-1}$	$-2.536943539674479 \cdot 10^{-1}$	0.0
10.0	$-2.611624603488405 \cdot 10^{-1}$	$-2.611624603488405 \cdot 10^{-1}$	0.0
30.0	$-2.743853873277227 \cdot 10^{-1}$	$-2.743853873277228 \cdot 10^{-1}$	$2.0 \cdot 10^{-16}$
100.0	$-2.796682147138912 \cdot 10^{-1}$	$-2.796682147138912 \cdot 10^{-1}$	$2.0 \cdot 10^{-16}$

$$w(x) - \frac{1}{k\sqrt{\pi}} \int_{-0.5}^{0.5} J_{-1}\left(\frac{|x-y|}{k}\right) w(y)\, \mathrm{d}y = h(x), \quad x \in [-0.5, 0.5], \tag{49a}$$

$$h(x) = -\frac{k}{\sqrt{\pi}}\left[J_1\left(\frac{0.5-x}{k}\right) - J_1\left(\frac{0.5+x}{k}\right)\right]. \tag{49b}$$

Since $w(x)$ is a small perturbation when k is small, the inaccuracy in the calculation of $w(x)$ has almost no effect on the overall accuracy of $u(x)$. This allows us to achieve the machine precision for all physical quantities of interest for a much wider range of the Knudsen number k than reported in [24].

We have repeated the calculations in [24] using the current method. For all values of k, the computational domain $[-0.5, 0.5]$ is divided into four coarse panels; n_{sub} is set to 41; and the GMRES stopping criterion is set to machine epsilon. Tables 1–3 list the values of the velocity u at $x = 0.5$, the stress P_{xy} [24, Eq. (43)] and the half-channel mass flow rate Q [24, Eq. (42)] at sample values of the Knudsen number k, where the second column contains the values in [24], the third column contains the values using the current method, and the last column shows the relative difference between these values. Here $u(0.5)$ is obtained via the backward recursion in Section 6, P_{xy} and Q are calculated using the velocity on the coarse grid with the kernel-split quadrature applied to treat the logarithmic singularity of $J_0(x)$ at $x = 0$ in P_{xy}. It is clear that the numerical results agree with those in [24] to machine precision for all sample values of k.

The current method is much more efficient as compared with the one used in [24]. For $k = 0.003$, the computation takes about 0.05 second of CPU time; and for $k = 10$, the computational time becomes unmeasurable with Matlab's cputime command for a single run, i.e., less than 0.01 second. In [24], the timing results are 62.8 and 23.9 seconds, respectively. The RCIP method eliminates the dyadic refinement towards the endpoints in the solve phase, and the upsampling scheme allows us to use such coarse panels that they only need to capture the features of the solution. The combination of these two techniques enables us to use the optimal number of discretization points for constructing the system matrix. The full machine precision accuracy of the current method completely removes the need of using quadruple precision arithmetic. Finally, the preconditioner **R** also reduces the number of GMRES iterations. Indeed, the number of GMRES iterations required is at most 6 for all sample values of k, whereas 430 was reported in [24] for $k = 0.003$. All of this results in a significant reduction in the computational cost, i.e., a speedup of at least a factor of 1000.

Table 3
The half-channel mass flow rate Q at sample values of the Knudsen number k.

k	Q in [24]	Q (current)	Error
0.003	$1.242445655299172 \cdot 10^{-1}$	$1.242445655299167 \cdot 10^{-1}$	$4.4 \cdot 10^{-15}$
0.01	$1.225330275292623 \cdot 10^{-1}$	$1.225330275292621 \cdot 10^{-1}$	$1.8 \cdot 10^{-15}$
0.03	$1.180147037188893 \cdot 10^{-1}$	$1.180147037188893 \cdot 10^{-1}$	$1.2 \cdot 10^{-16}$
0.1	$1.057028408172292 \cdot 10^{-1}$	$1.057028408172292 \cdot 10^{-1}$	$2.6 \cdot 10^{-16}$
0.3	$8.560111699820618 \cdot 10^{-2}$	$8.560111699820613 \cdot 10^{-2}$	$4.9 \cdot 10^{-16}$
1.0	$5.804708735555459 \cdot 10^{-2}$	$5.804708735555460 \cdot 10^{-2}$	$2.4 \cdot 10^{-16}$
2.0	$4.281659776113917 \cdot 10^{-2}$	$4.281659776113918 \cdot 10^{-2}$	$1.6 \cdot 10^{-16}$
3.0	$3.489298506190833 \cdot 10^{-2}$	$3.489298506190833 \cdot 10^{-2}$	$2.0 \cdot 10^{-16}$
5.0	$2.627042060967383 \cdot 10^{-2}$	$2.627042060967383 \cdot 10^{-2}$	$1.3 \cdot 10^{-16}$
7.0	$2.147460412330841 \cdot 10^{-2}$	$2.147460412330841 \cdot 10^{-2}$	$1.6 \cdot 10^{-16}$
10.0	$1.714449048590649 \cdot 10^{-2}$	$1.714449048590649 \cdot 10^{-2}$	$2.0 \cdot 10^{-16}$
30.0	$8.043009085700258 \cdot 10^{-3}$	$8.043009085700263 \cdot 10^{-3}$	$6.5 \cdot 10^{-16}$
100.0	$3.226757181742400 \cdot 10^{-3}$	$3.226757181742397 \cdot 10^{-3}$	$9.4 \cdot 10^{-16}$

9. Concluding remarks

Finding efficient solvers for SKIEs on non-smooth boundaries is a field with substantial recent activity. However, almost all existing methods only work well for smooth right-hand sides and may also involve a fair amount of operator-specific analysis and precomputed quantities. In contrast, the method constructed in this paper for singular and nearly singular right-hand sides computes all intermediary quantities needed on-the-fly and only involves a bare minimum of analysis.

The extension of the current method to multiple right-hand sides can be carried out easily. The compressed inverse **R** is independent on f and needs only to be computed once, while the compressed inverse $\mathbf{R}_f$ depends on f and needs to be computed afresh for each new f. The setup cost of $\mathbf{R}_f$ for each additional f can, furthermore, be reduced since several of the matrices entering into the forward recursion for $\mathbf{R}_f$ are independent of f and can be stored after they have been computed on-the-fly. A similar situation holds in the backward recursions for ρ on the fine grid – should that quantity be needed.

In a certain sense, the work completes the RCIP method in two dimensions since the splitting into a smooth part and a local singular part is carried out on both sides of the integral equation. The RCIP method can be generalized to three dimensions. In [20], the RCIP method was extended to solve a boundary integral equation on the surface of a cube. In [18], it was extended to solve boundary integral equations on axially symmetric surfaces. In a recent talk [12], an RCIP-type scheme for discretizing corner and edge singularities in three dimensions was discussed. For general surfaces in three dimensions, the extension of the RCIP is more or less straightforward when the geometry admits a local hierarchical discretization near the corner and edge singularities; and the extension of the work in this paper should follow subsequently.

CRediT authorship contribution statement

Johan Helsing, Shidong Jiang: Equal contribution.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

Appendix A. Derivation of the forward recursion formula (20) for $\mathbf{R}_f$

First, we prove the following lemma.

Lemma 1. *Suppose that* **A** *is an invertible* $m \times m$ *matrix,* **B** *is an* $n \times n$ *matrix,* **U** *is an* $n \times m$ *matrix, and* **V** *is an* $m \times n$ *matrix with* $m \geq n$. *Suppose further that* $\mathbf{UA}^{-1}\mathbf{V}$ *and* $\mathbf{A} + \mathbf{VBU}$ *are both invertible. Then*

$$\mathbf{U}(\mathbf{A} + \mathbf{VBU})^{-1}\mathbf{V} = \left((\mathbf{UA}^{-1}\mathbf{V})^{-1} + \mathbf{B}\right)^{-1}, \tag{A.1}$$

$$\mathbf{U}(\mathbf{A} + \mathbf{VBU})^{-1} = \left((\mathbf{UA}^{-1}\mathbf{V})^{-1} + \mathbf{B}\right)^{-1}(\mathbf{UA}^{-1}\mathbf{V})^{-1}\mathbf{UA}^{-1}. \tag{A.2}$$
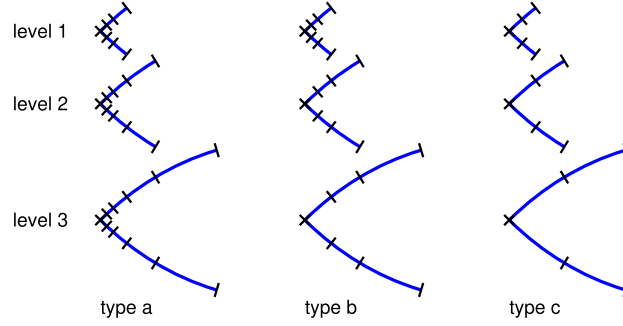
Fig. A.6. Meshes of type a, type b, and type c at different levels. The type a mesh contains $4 + 2i$ dyadic fine panels at level i. The type b mesh always contains six panels, and the type c mesh always contains four panels. At level 1, the type a mesh is identical to the type b mesh. At level n_{sub}, the type c mesh contains four coarse panels on $\Gamma^\star$ in Fig. 1.

Proof. Introduce a complex parameter λ and consider $(\mathbf{A} + \lambda \mathbf{VBU})^{-1}$. When λ is sufficiently small, the following Taylor expansion is valid

$$(\mathbf{A} + \lambda \mathbf{VBU})^{-1} = \mathbf{A}^{-1} - \lambda \mathbf{A}^{-1}\mathbf{VBUA}^{-1} + \lambda^2 \mathbf{A}^{-1}\mathbf{VBUA}^{-1}\mathbf{VBUA}^{-1} - \dots, \tag{A.3}$$

Mutiplying both sides of (A.3) with $\mathbf{U}$ from the left and with $\mathbf{V}$ from the right and regrouping, we obtain

$$\begin{aligned}
\mathbf{U}(\mathbf{A} + \lambda \mathbf{VBU})^{-1}\mathbf{V} &= \mathbf{UA}^{-1}\mathbf{V} - \lambda \left(\mathbf{UA}^{-1}\mathbf{V}\right)\mathbf{B}\left(\mathbf{UA}^{-1}\mathbf{V}\right) \\
&\quad + \lambda^2 \left(\mathbf{UA}^{-1}\mathbf{V}\right)\mathbf{B}\left(\mathbf{UA}^{-1}\mathbf{V}\right)\mathbf{B}\left(\mathbf{UA}^{-1}\mathbf{V}\right) - \dots \\
&= \left(\left(\mathbf{UA}^{-1}\mathbf{V}\right)^{-1} + \lambda \mathbf{B}\right)^{-1},
\end{aligned} \tag{A.4}$$

where the second equality follows from the application of the Taylor expansion in the reverse order. By the cofactor formula of the matrix inverse and the so-called big formula for the matrix determinant [34], both sides of (A.4) are rational functions of λ. Since these two rational functions are equal in a small neighborhood of the origin in the complex plane, they must be equal everywhere in the whole complex plane by analytic continuation [3]. Setting $\lambda = 1$ in (A.4), we establish (A.2) and the invertibility of $(\mathbf{UA}^{-1}\mathbf{V})^{-1} + \mathbf{B}$ simultaneously. (A.2) can be proved in an almost identical manner. $\quad\square$

We now recall the definition of $\mathbf{R}_f$ in (16)

$$\mathbf{R}_f = \mathbf{P}_W^{\mathsf{T}}\left(\mathbf{I}_{fin} + \mathbf{K}_{fin}^\star\right)^{-1}\mathbf{P}_f, \tag{A.5}$$

where $\mathbf{K}_{fin}^\star$ is the system matrix built on a fine mesh that is obtained via n_{sub} level of dyadic refinement of the four coarsest panels (with two panels on each side) near the singular point. Let n_{gl} be the number of Gauss-Legendre nodes on each panel. Then the total number of discretization points on the fine mesh is $n_{gl}(4 + 2n_{sub})$. Clearly, direct application of (A.5) is very expensive, inaccurate, and unrobust. Instead, the forward recursion is used to compute $\mathbf{R}_f$. The forward recursion starts from the finest six panels around the singular point at level $i = 1$, adds one panel on each side as the level goes up, and reaches the full fine mesh at level $i = n_{sub}$. See Fig. A.6 for an illustration of three different types of meshes at different levels, where the type a mesh is needed only in the derivation of the forward recursion. The actual recursion formula involves only type b and type c meshes. To be more precise, at any step in the forward recursion, one only needs to build a system matrix of size $6n_{gl} \times 6n_{gl}$, i.e., on the six panels of a type b mesh, and the type c mesh is used only implicitly in the construction of the prolongation matrix $\mathbf{P}$ and its weighted version $\mathbf{P}_W$. Similar to [16, Eq. (D.1)], we define

$$\mathbf{R}_{fi} := \mathbf{P}_{Wiac}^{T}(\mathbf{I}_{ia} + \mathbf{K}_{ia})^{-1}\mathbf{P}_{fiac}, \qquad i = 1, \cdots, n_{sub}. \tag{A.6}$$

By the action of $\mathbf{P}_{fiac}$ and $\mathbf{P}_{Wiac}^T$, $\mathbf{R}_{fi}$ is always a matrix of size $4n_{gl} \times 4n_{gl}$ at any level i. A derivation similar to that of [16, Eq. (D.6)] leads to

$$\mathbf{R}_{fi} = \mathbf{P}_{Wbc}^{T}\mathbf{P}_{Wiab}^{T}(\mathbf{I}_{ia} + \mathbb{F}\{\mathbf{K}_{(i-1)a}\} + \mathbf{P}_{iab}\mathbf{K}_{ib}^{\circ}\mathbf{P}_{Wiab}^{T})^{-1}\mathbf{P}_{fiab}\mathbf{P}_{fibc}, \tag{A.7}$$

where we have assumed that the low-rank property

$$\mathbf{K}_{ia}^{\circ} = \mathbf{P}_{iab}\mathbf{K}_{ib}^{\circ}\mathbf{P}_{Wiab}^{T} \tag{A.8}$$

holds to machine precision, as in [16, Eq. (D.3)]. Since the type a mesh differs from the type b mesh only on the two panels (i.e., $\Gamma_{ib}^{\star\star}$) closest to the singular point in the type b mesh, the only diagonal blocks in $\mathbf{P}_{iab}$ and $\mathbf{P}_{Wiab}$ that are not identity
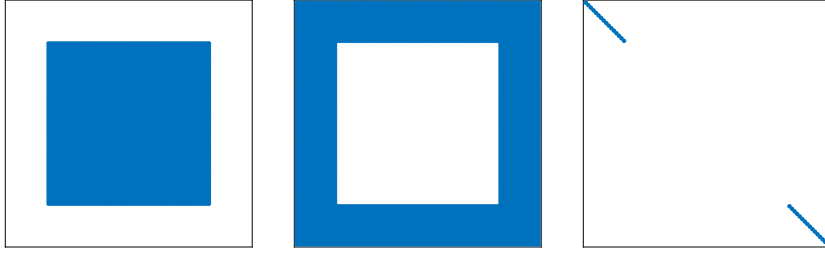
Fig. A.7. Nonzero patterns of $\mathbb{F}\{\mathbf{R}_{i-1}^{-1}\}$ and $\mathbb{F}\{\mathbf{R}_{f(i-1)}\}$ (left), $\mathbf{K}_{ib}^{\circ}$ (center), and $\mathbf{I}_{b}^{\circ}$ (right). Note that the patterns depend on the fact that when constructing the system matrix in the forward recursion, the sources and targets are arranged in the order from the top panel to the bottom panel on a type b mesh, see Fig. A.6.

matrices are $\mathbf{P}_{iab}(\mathbf{I}_{ia}^{\star\star}, \mathbf{I}_{ib}^{\star\star})$ and $\mathbf{P}_{Wiab}(\mathbf{I}_{ia}^{\star\star}, \mathbf{I}_{ib}^{\star\star})$, respectively. Here $\mathbf{I}_{ia}^{\star\star}$ and $\mathbf{I}_{ib}^{\star\star}$ contain indices corresponding to discretization points in $\Gamma_{ib}^{\star\star}$ on type a and type b meshes, respectively. Now it is straightforward to verify that (A.8) is equivalent to

$$\mathbf{K}_{ia}(\mathbf{I}_{ia}^{\star\star}, \mathbf{I}_{ib}^{\circ}) = \mathbf{P}_{iab}(\mathbf{I}_{ia}^{\star\star}, \mathbf{I}_{ib}^{\star\star})\mathbf{K}_{ib}(\mathbf{I}_{ib}^{\star\star}, \mathbf{I}_{ib}^{\circ}),$$
$$\mathbf{K}_{ia}(\mathbf{I}_{ib}^{\circ}, \mathbf{I}_{ia}^{\star\star}) = \mathbf{K}_{ib}(\mathbf{I}_{ib}^{\circ}, \mathbf{I}_{ib}^{\star\star})\mathbf{P}_{Wiab}^{T}(\mathbf{I}_{ib}^{\star\star}, \mathbf{I}_{ia}^{\star\star}),$$

(A.9)

where $\mathbf{I}_{ib}^{\circ}$ contains indices corresponding to discretization points on Γ_{ib}°, that is, on the two outermost panels from γ. Since $\Gamma_{ib}^{\star\star}$ is always well-separated from Γ_{ib}° for any level i, (A.9) is equivalent to stating that the kernel function $K(r, r')$ is smooth when $r \in \Gamma_{ib}^{\star\star}$ and $r' \in \Gamma_{ib}^{\circ}$, or vice versa, and thus the interaction between $\Gamma_{ib}^{\star\star}$ and Γ_{ib}° can be discretized to machine precision with n_{gl} points on each panel in $\Gamma_{ib}^{\star\star}$ provided that n_{gl} is not too small. This holds for all kernels we have encountered in practice, including, say, highly oscillatory ones, if the coarse mesh is chosen in such a way that the oscillations of the kernel are well-resolved. We emphasize that this is a property of the kernel function $K(r, r')$ and the type a and b meshes. It has nothing to do with the singularity of the right-hand side f.

Applying (A.2) to part of the right-hand side of (A.7), we obtain

$$\mathbf{P}_{Wiab}^{T}(\mathbf{I}_{ia} + \mathbb{F}\{\mathbf{K}_{(i-1)a}\} + \mathbf{P}_{iab}\mathbf{K}_{ib}^{\circ}\mathbf{P}_{Wiab}^{T})^{-1} =$$
$$\left[\left(\mathbf{P}_{Wiab}^{T}(\mathbf{I}_{ia} + \mathbb{F}\{\mathbf{K}_{(i-1)a}\})^{-1}\mathbf{P}_{iab} \right)^{-1} + \mathbf{K}_{ib}^{\circ} \right]^{-1}$$
$$\cdot \left(\mathbf{P}_{Wiab}^{T}(\mathbf{I}_{ia} + \mathbb{F}\{\mathbf{K}_{(i-1)a}\})^{-1}\mathbf{P}_{iab} \right)^{-1} \mathbf{P}_{Wiab}^{T}(\mathbf{I}_{ia} + \mathbb{F}\{\mathbf{K}_{(i-1)a}\})^{-1}.$$

(A.10)

Recall that [16, Eq. (D.10)] states that

$$\mathbb{F}\{\mathbf{R}_{i-1}\} + \mathbf{I}_{b}^{\circ} = \mathbf{P}_{Wiab}^{T}(\mathbf{I}_{ia} + \mathbb{F}\{\mathbf{K}_{(i-1)a}\})^{-1}\mathbf{P}_{iab}.$$

(A.11)

We observe that $\mathbb{F}\{\mathbf{R}_{i-1}\}$ places $\mathbf{R}_{i-1}$ in the center $4n_{gl} \times 4n_{gl}$ block with zero padding in a $6n_{gl} \times 6n_{gl}$ matrix, and the nonzero blocks of $\mathbf{I}_{b}^{\circ}$ are the identity matrix of size n_{gl} in the top and bottom diagonal blocks (Fig. A.7). Thus,

$$\mathbb{F}\{\mathbf{R}_{i-1}^{-1}\} + \mathbf{I}_{b}^{\circ} = \left(\mathbb{F}\{\mathbf{R}_{i-1}\} + \mathbf{I}_{b}^{\circ} \right)^{-1}$$
$$= \left(\mathbf{P}_{Wiab}^{T}(\mathbf{I}_{ia} + \mathbb{F}\{\mathbf{K}_{(i-1)a}\})^{-1}\mathbf{P}_{iab} \right)^{-1}.$$

(A.12)

Similarly,

$$\mathbb{F}\{\mathbf{R}_{f(i-1)}\} = \mathbb{F}\{\mathbf{P}_{W(i-1)ac}^{T}(\mathbf{I}_{(i-1)a} + \mathbf{K}_{(i-1)a})^{-1}\mathbf{P}_{f(i-1)ac}\}$$
$$= \mathbf{P}_{Wiab}^{T}(\mathbf{I}_{ia} + \mathbb{F}\{\mathbf{K}_{(i-1)a}\})^{-1}\mathbf{P}_{fiab} - \mathbf{I}_{b}^{\circ},$$

(A.13)

where the second equality uses $\left(\mathbf{P}_{Wiab}^{T}\mathbf{P}_{fiab} \right)^{\circ} = \mathbf{I}_{b}^{\circ}$. Combining (A.7), (A.10)–(A.13), we obtain

$$\mathbf{R}_{fi} = \mathbf{P}_{Wbc}^{T} \left(\mathbb{F}\{\mathbf{R}_{i-1}^{-1}\} + \mathbf{I}_{b}^{\circ} + \mathbf{K}_{ib}^{\circ} \right)^{-1} \left(\mathbb{F}\{\mathbf{R}_{i-1}^{-1}\} + \mathbf{I}_{b}^{\circ} \right) \left(\mathbb{F}\{\mathbf{R}_{f(i-1)}\} + \mathbf{I}_{b}^{\circ} \right) \mathbf{P}_{fibc}.$$

(A.14)

Finally, we arrive at (20) by observing that

$$\left(\mathbb{F}\{\mathbf{R}_{i-1}^{-1}\} + \mathbf{I}_{b}^{\circ} \right) \left(\mathbb{F}\{\mathbf{R}_{f(i-1)}\} + \mathbf{I}_{b}^{\circ} \right) = \mathbb{F}\{\mathbf{R}_{i-1}^{-1}\mathbf{R}_{f(i-1)}\} + \mathbf{I}_{b}^{\circ},$$

(A.15)

due to the nonzero patterns of the involved matrices shown in Fig. A.7.

References

[1] M. Abramowitz, Evaluation of the integral $\int_0^\infty e^{-u^2-x/u}du$, J. Math. Phys. Camb. 32 (1953) 188–192.
[2] L. af Klinteberg, F. Fryklund, A.-K. Tornberg, An adaptive kernel-split quadrature method for parameter-dependent layer potentials, arXiv preprint, arXiv:1906.07713, 2019.
[3] L.V. Ahlfors, Complex Analysis: An Introduction to the Theory of Analytic Functions of One Complex Variable, McGraw-Hill, New York, 1966.
[4] T. Askham, L. Greengard, Norm-preserving discretization of integral equations for elliptic PDEs with internal layers I: the one-dimensional case, SIAM Rev. 56 (4) (2014) 625–641.
[5] P.L. Bhatnagar, E.P. Gross, M. Krook, A model for collision processes in gases. I. Small amplitude processes in charged and neutral one-component systems, Phys. Rev. 94 (3) (1954) 511–525.
[6] J. Bremer, On the Nyström discretization of integral equations on planar curves with corners, Appl. Comput. Harmon. Anal. 32 (1) (2012) 45–64.
[7] C. Cercignani, Rarefied Gas Dynamics: From Basic Concepts to Actual Calculations, Cambridge University Press, Cambridge, UK, 2000.
[8] H. Cheng, W. Crutchfield, Z. Gimbutas, L. Greengard, J. Huang, V. Rokhlin, N. Yarvin, J. Zhao, Remarks on the implementation of the wideband FMM for the Helmholtz equation in two dimensions, in: Inverse Problems, Multi-Scale Analysis and Effective Medium Theory, in: Contemp. Math., vol. 408, Amer. Math. Soc., Providence, RI, 2006, pp. 99–110.
[9] D. Colton, R. Kress, Inverse Acoustic and Electromagnetic Scattering Theory, Springer, New York, NY, 2012.
[10] Y.U. Devi, M. Rukmini, B. Madhav, A compact conformal printed dipole antenna for 5G based vehicular communication applications, Prog. Electromagn. Res. C 85 (2018) 191–208.
[11] F. Ethridge, L. Greengard, A new fast-multipole accelerated Poisson solver in two dimensions, SIAM J. Sci. Comput. 23 (3) (2001) 741–760.
[12] Z. Gimbutas, Edge and corner preconditioners in three dimensions, SIAM Annu. Meet. (2021).
[13] Z. Gimbutas, S. Jiang, L.-S. Luo, Evaluation of Abramowitz functions in the right half of the complex plane, J. Comput. Phys. 405 (2020) 109169.
[14] L. Greengard, V. Rokhlin, A fast algorithm for particle simulations, J. Comput. Phys. 73 (2) (1987) 325–348.
[15] J. Helsing, Integral equation methods for elliptic problems with boundary conditions of mixed type, J. Comput. Phys. 228 (23) (2009) 8892–8907.
[16] J. Helsing, Solving integral equations on piecewise smooth boundaries using the RCIP method: a tutorial, arXiv preprint, arXiv:1207.6737v9, 2018.
[17] J. Helsing, S. Jiang, On integral equation methods for the first Dirichlet problem of the biharmonic and modified biharmonic equations in nonsmooth domains, SIAM J. Sci. Comput. 40 (4) (2018) A2609–A2630.
[18] J. Helsing, A. Karlsson, An explicit kernel-split panel-based Nyström scheme for integral equations on axially symmetric surfaces, J. Comput. Phys. 272 (2014) 686–703.
[19] J. Helsing, R. Ojala, On the evaluation of layer potentials close to their sources, J. Comput. Phys. 227 (5) (2008) 2899–2921.
[20] J. Helsing, K.-M. Perfekt, On the polarizability and capacitance of the cube, Appl. Comput. Harmon. Anal. 34 (3) (2013) 445–468.
[21] H.V. Henderson, S.R. Searle, On deriving the inverse of a sum of matrices, SIAM Rev. 23 (1) (1981) 53–60.
[22] N.J. Higham, Accuracy and Stability of Numerical Algorithms, SIAM, 2002.
[23] G.C. Hsiao, W.L. Wendland, Boundary Integral Equations, Applied Mathematical Sciences, vol. 164, Springer-Verlag, Berlin, 2008.
[24] S. Jiang, L.-S. Luo, Analysis and solutions of the integral equation derived from the linearized BGKW equation for the steady Couette flow, J. Comput. Phys. 316 (2016) 416–434.
[25] W. Kahan, Further remarks on reducing truncation errors, Commun. ACM 8 (1965) 40.
[26] R. Kress, Linear Integral Equations, third edition, Applied Mathematical Sciences, vol. 82, Springer–Verlag, Berlin, 2014.
[27] W. Li, L.-S. Luo, J. Shen, Accurate solution and approximations of the linearized BGK equation for steady Couette flow, Comput. Fluids 111 (2015) 18–32.
[28] J. Ma, V. Rokhlin, S. Wandzura, Generalized Gaussian quadrature rules for systems of arbitrary functions, SIAM J. Numer. Anal. 33 (3) (1996) 971–996.
[29] A.J. MacLeod, Chebyshev expansions for Abramowitz functions, Appl. Numer. Math. 10 (1992) 129–137.
[30] P.-G. Martinsson, Fast Direct Solvers for Elliptic PDEs, CBMS-NSF Regional Conference Series in Applied Mathematics, vol. 96, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2020.
[31] Q. Nie, F.-R. Tian, Singularities in Hele–Shaw flows driven by a multipole, SIAM J. Appl. Math. 62 (2) (2001) 385–406.
[32] C. Pozrikidis, Boundary Integral and Singularity Methods for Linearized Viscous Flow, Cambridge Texts in Applied Mathematics, Cambridge University Press, Cambridge, 1992.
[33] T.S. Rappaport, G.R. MacCartney, S. Sun, H. Yan, S. Deng, Small-scale, local area, and transitional millimeter wave propagation for 5G communications, IEEE Trans. Antennas Propag. 65 (12) (2017) 6474–6490.
[34] G. Strang, Introduction to Linear Algebra, 5th edition, Wellesley-Cambridge Press, Wellesley, MA, 2016.
[35] P. Welander, On the temperature jump in a rarefied gas, Ark. Fys. 7 (1954) 507–553.