On the use of rational-function fitting methods for the solution of 2D Laplace boundary-value problems

Amit Hochman, Yehuda Leviatan, and Jacob K. White

Abstract

A computational scheme for solving 2D Laplace boundary-value problems using rational functions as the basis functions is described. The scheme belongs to the class of desingularized methods, for which the location of singularities and testing points is a major issue that is addressed by the proposed scheme, in the context of the 2D Laplace equation. Well-established rational-function fitting techniques are used to set the poles, while residues are determined by enforcing the boundary conditions in the least-squares sense at the nodes of rational Gauss-Chebyshev quadrature rules. Numerical results show that errors approaching the machine epsilon can be obtained for sharp and almost sharp corners, nearly-touching boundaries, and almost-singular boundary data. We show various examples of these cases in which the method yields compact solutions, requiring fewer basis functions than the Nyström method, for the same accuracy. A scheme for solving fairly large-scale problems is also presented.

1. Introduction

Laplace boundary-value problems (BVPs) are prevalent in the applied sciences, and methods for their solution have been the subject of study for decades. Among these, prominent approaches are finite-difference and finite-element methods, integral-equation methods, and, in 2D, conformal-mapping techniques. While the Laplace equation is the simplest elliptic partial differential equation (PDE), improving methods for its solution is nevertheless of continued interest. At the focus of recent research are problems with small radii of curvature, nearly-touching boundaries, and singular, or almost-singular boundary data [1–6].

The technique described in this paper belongs to a class of methods known as desingularized methods, which are used for the solution of linear BVPs. What characterizes these methods is that the basis functions are elementary solutions of the homogenous PDE, and hence the singularities of the basis functions lie outside the BVP domain. The simplest example of a desingularized method is the method of images, which is only applicable in a handful of special cases for which the exact locations and amplitudes of the image sources are known analytically. For more general problems, an approximate solution can be obtained by setting the locations of the sources heuristically and solving for the amplitudes so that the boundary conditions are satisfied at a set of testing points. Various types of image sources and heuristics have been proposed over the years. Vekua [7] and Kupradze [8] are usually credited as the pioneers of these methods, and

other early works include those of Fox, Henrici, and Moler [9], Bergman [10], and Eisenstat [11]. Nowadays, the methods in this category go by various names, such as method of fundamental solutions [12, 13], method of auxiliary sources [14, 15], generalized multipole technique [16], method of fictitious sources [17, 18, Ch. 1.5.6], and source-model technique [19–21].

Desingularized methods share many of the properties of integral-equation methods, and indeed can be viewed as such when the sources are distributed on a curve lying outside of the BVP domain. The basis functions used for the density functions are then Dirac delta functions. As the potentials due to these basis functions are smooth on the boundaries, it is customary to enforce the boundary conditions at a set of testing points. This obviates laborious panel integrals that slow down conventional Galerkin schemes for boundary integral equations, making the cost of forming the system matrix similar to that of a Nyström method. Also, the representation of the solution with elementary sources is very convenient for post-processing tasks such as calculating capacitances, charges, forces, etc., and the potential itself at arbitrary points.

Owing to the above properties, desingularized methods are among the simplest methods for solving linear BVPs. They can also be very accurate, provided the sources and testing points are located "adequately". To substantiate this claim, we show a couple of results obtained with the method proposed in this paper. In Fig. 1a, 30 dipoles are used to solve the Laplace equation outside two closely spaced curves. The solution is required

Solve time: 17.3s
$\Delta E_{\max} = 1.6 \times 10^{-6}$

(a)

Solve time: 5.7s
$\Delta E_{\max} = 6.6 \times 10^{-5}$

(b)
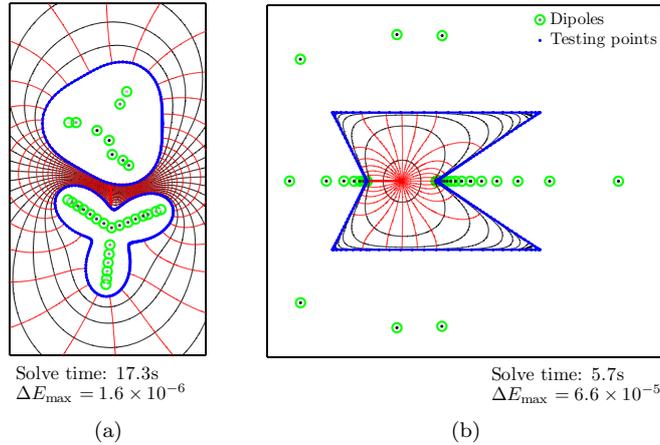
Figure 1: Solutions obtained with a small number of basis functions, for (a) exterior and (b) interior Laplace Dirichlet problems. The number of dipoles is 30 in (a) and 60 in (b). Darker dots correspond to larger dipole amplitudes.

to be -1 on the lower curve and $+1$ on the upper curve, and this boundary condition is satisfied with a maximum error of $\Delta E_{\max} = 1.7 \times 10^{-6}$. In Fig. 1b, an interior Dirichlet problem with two perfectly sharp reentrant corners is solved using 60 dipoles. The boundary condition is due to a monopole located inside the polygon and the maximum error is $\Delta E_{\max} = 2.5 \times 10^{-5}$. The Nyström method, which is spectrally accurate when the boundaries are smooth [22], is the standard against which the proposed method is compared with in this paper. Such a comparison, shown in Fig. 2, reveals that when

the boundary curves nearly touch, a desingularized method can indeed be much more accurate than the Nyström method.
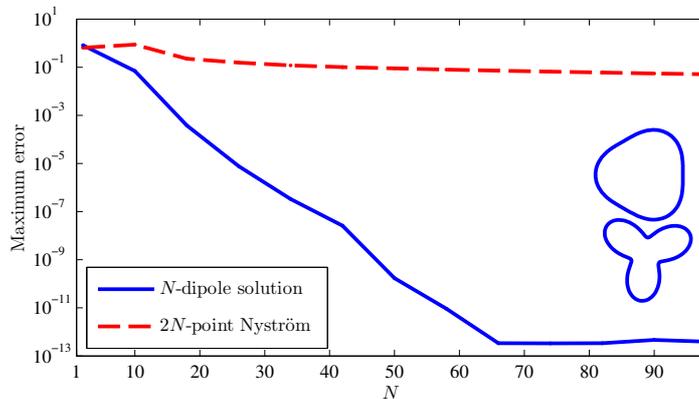
Figure 2: For the problem shown in Fig. 1a, maximum errors on the order of 10^{-12} are obtained with no more than 70 dipoles. Using the Nyström method with the same number of basis functions yields errors of about 5%.

The adequate location of the sources and testing points is a major unresolved issue with desingularized methods, and much work has been devoted to its analysis ([23–27] are just a few references out of a large body of work). Source location affects both the rate of error decay and the effect of round-off errors which may limit the highest accuracy achievable. Also, the effective application of the fast multipole method (FMM) [28] appears to depend strongly on the source location [29]. The problem with applying the FMM to a desingularized method is that the linear systems to be solved are generally ill-conditioned. In [29], the authors proposed a block-diagonal preconditioner that accelerates convergence, provided the distance between the sources curve and the boundary curve is less than a problem dependent critical distance.

Source location schemes that guarantee exponential convergence were developed in [24] and [25, 30], for Laplace and Helmholtz Dirichlet BVPs in simply-connected analytic domains, respectively. In these works, sources are distributed uniformly (in so-called conformal angle) on curves conformal with the boundary, and the resulting error in the solution is shown to decrease exponentially, in exact arithmetic. But placing sources on a conformal curve may be restrictive, and any a-priori source restriction requires a-priori knowledge of singularity locations.

A common approach has been to formulate the source-location problem as an optimization problem. If the number of sources is set in advance, finding the best source locations is a nonlinear least-squares problem, and various attempts have been made to use nonlinear optimization techniques to determine the source locations. Unfortunately, this optimization problem is not convex and has many local minima. Therefore, descent-type methods such as Gauss-Newton and Levenberg-Marquardt are sensitive to the initial source locations [12, 31, 32], while more robust methods that have been tried, such as simulated annealing [33] and genetic algorithms [34], may take a very long time to converge. Moreover, if the positions of the sources are adapted, the positions of the

testing points must be adapted as well. If this is not done, and the sources approach the boundary, the error at the testing points will not be representative of the error on the boundary. Alternatively, if the cost function of the optimization is defined as an integral on the squared error (as in [32], for example), the quadrature for this integral must be adapted along with the source positions. This issue of adapting the testing points has been largely ignored in the literature; in this paper it is addressed by use of rational Gauss-Chebyshev quadrature rules. Perhaps the most practical approaches use geometry-based heuristics [16, 25, 35, 36]. While these can be more than adequate in many engineering applications, they fail to take full advantage of the flexibility in locating the sources and their accuracy may be limited.

In this paper we describe a new method for determining appropriate locations for the sources and testing points. The method is tailored to the 2D Laplace equation, for which the solution can be represented as the real part of a complex potential which we approximate by the sum of a rational function and (for multiply-connected domains) a few logarithmic terms. We use well-established techniques for finding rational-function approximations to the boundary data, and the poles of these rational functions correspond then to the positions of dipole sources.

The major difficulty in applying this idea is that, in a Dirichlet problem, the boundary data corresponds to the real part of the complex potential on the boundary. The imaginary part is determined, up to an additive constant, by the Dirichlet-to-Neumann map. Of course, applying this map is tantamount to solving the Dirichlet problem. Similarly, in a Neumann problem, only the imaginary part of the complex potential on the boundary is known. The solution we propose for this problem is to estimate the missing information (say, the imaginary part in a Dirichlet problem) by solving the BVP with a given set of poles and then using this information to improve the pole locations. Assuming the pole locations are indeed improved, they can be used to find a better estimate of the imaginary part, which in turn is used to improve their locations further. We proceed in this way until the pole positions appear to have converged, the error drops below a preset tolerance, or a maximum number of iterations is reached.

Clearly, the process just described is not guaranteed to converge to the solution that would be obtained were the imaginary part known in advance. It is possible that the estimated imaginary part will be inaccurate, and the poles found based on this estimate will not yield any improvement on the next iteration. When this happens, the improvement in accuracy stalls. It is then usually, though not always, possible to increase the accuracy further by increasing the number of poles. In numerical experiments, we have found that despite this difficulty, errors approaching the machine epsilon can be obtained for various challenging configurations.

2. Dirichlet problem, simply-connected domain

We begin by considering the simplest case, in which we seek a real valued potential, $U(x, y)$, which obeys the Laplace equation. More precisely,

$$\nabla^2 U(x, y) = 0, \quad \text{for all } (x, y) \in \Omega, \tag{1}$$

where Ω is a simply connected domain in $\mathbb{R}^2$ bounded by a curve Γ, and U must also obey a Dirichlet boundary condition,

$$U|_{\text{on } \Gamma} = f. \tag{2}$$

4

The method we propose for solving this problem is to approximate the solution as a weighted combination of dipole potentials, and to use rational function fitting algorithms to determine the dipole locations.

2.1. Dipole representation

As is well-known, the potential U can be sought as the real part of a complex function $W(z)$, where $z = x + iy$ and $W(z)$ is holomorphic and single valued in Ω (it is single valued because Ω is simply connected). The classical single- and double-layer potentials,

$$W_{\text{single}}(z) = \int_{\Gamma} \sigma(z') \log(z' - z) dz', \tag{3}$$

and

$$W_{\text{double}}(z) = \frac{1}{2\pi i} \int_{\Gamma} \frac{\mu(z')}{z' - z} dz', \tag{4}$$

where $\sigma(z)$ and $\mu(z)$ are referred to as the monopole and dipole densities, respectively, have been used extensively to formulate integral equations [22, 37, 38]. For example, a second-kind Fredholm equation is readily obtained by combining the double-layer potential, (4), with the Dirichlet boundary condition, (2), or by combining the single-layer potential, (3), with the Neumann boundary condition.

In desingularized methods, the most common approach is to use the single-layer potential,

$$W_{\Gamma'}(z) = \int_{\Gamma'} \sigma(z') \log(z' - z) dz', \tag{5}$$

where Γ' encloses, but does not intersect, Γ. Enforcing the boundary condition on Γ leads to an ill-conditioned Fredholm equation of the first kind. Nevertheless, as discussed in [25], if $W(z)$ can be continued analytically throughout the region enclosed between Γ and Γ', highly accurate solutions can be obtained, despite this ill-conditioning. The density $\sigma(z)$ is often approximated as a weighted combination of Dirac delta functions, as in

$$\sigma(z) = \sum_{n} \sigma_n \delta(z_n - z), \tag{6}$$

in which case

$$W_{\Gamma'}(z) \approx \sum_{n} \sigma_n \log(z_n - z). \tag{7}$$

Using a double-layer distribution instead of the single-layer distribution in (5) does not yield a second-kind integral equation, as it does in the classical formulations, but this is nevertheless a viable option [16, 39]. The approach in this work is similar, in that we also use dipole potentials as basis functions. However, the source points are not restricted to lie on any given curve, but may be placed anywhere outside Ω. Assuming that $W(z)$ is also holomorphic for $z \in \Gamma$, it can always be expressed as a series of complex dipole potentials,

$$W(z) = \sum_{n=1}^{\infty} \frac{a_n}{z'_n - z}, \quad \forall z \in \Omega, \tag{8}$$

where the z'_n are referred to as the poles of $W(z)$.

According to a theorem by Wolff [40] (see also [41, Ch.1, Theorem 13]), given the appropriate choice of poles, $z'_n \notin \overline{\Omega}$, (8) converges uniformly on any closed set in Ω. In this sense, we may say that the set of single-pole functions is a complete set, which can be used to approximate any function $W(z)$ which is holomorphic for $z \in \overline{\Omega}$. This theorem does not cover the case of $W(z)$ having a branch-point on Γ. Nevertheless, in Section 7.1.4 we show a solution with error $< 10^{-10}$ even for such a case. The convergence theory of rational function approximations to functions with branch-points is rather involved, and we therefore refer the interested reader to a couple of relevant works [42, 43].

Our algorithm is based on approximating the complex potential using N poles plus a constant, or equivalently as an Nth order rational function

$$W(z) \approx a_0 + \sum_{n=1}^{N} \frac{a_n}{z'_n - z} = \frac{P_N(z)}{Q_N(z)} \tag{9}$$

where P_N and Q_N are complex-coefficient polynomials of degree N. To determine $a_0, a_1, \ldots, a_N$, or equivalently, the coefficients of $P_N(z)$, we start with an initial guess for the poles, and associate a complex basis function with each pole,

$$\phi_n^{\text{C}}(z) = \frac{1}{z'_n - z}, \tag{10}$$

with real and imaginary parts,

$$\phi_n^{\text{R}}(z) = \text{Re}\left[\phi_n^{\text{C}}(z)\right], \quad \phi_n^{\text{I}}(z) = \text{Im}\left[\phi_n^{\text{C}}(z)\right]. \tag{11}$$

Note that $\phi_n^{\text{R}}(z)$ and $-\phi_n^{\text{I}}(z)$ are real-valued basis functions that correspond to x- and y-directed dipole potentials. In the special case of the constant term, the complex basis function becomes a real-valued basis function,

$$\phi_0^{\text{C}} = \phi_0^{\text{R}} = 1. \tag{12}$$

Defining $U(z) = \text{Re}[W(z)]$ and $V(z) = \text{Im}[W(z)]$, and using (10)-(12),

$$U(z) \approx \widehat{U}(z) = \sum_{n=0}^{N} a_n^{\text{R}} \phi_n^{\text{R}}(z) - \sum_{n=1}^{N} a_n^{\text{I}} \phi_n^{\text{I}}(z), \tag{13}$$

and its harmonic conjugate,

$$V(z) \approx \widehat{V}(z) = \sum_{n=1}^{N} a_n^{\text{R}} \phi_n^{\text{I}}(z) + \sum_{n=1}^{N} a_n^{\text{I}} \phi_n^{\text{R}}(z), \tag{14}$$

where the basis function coefficients $a_n^{\text{R}} \equiv \text{Re}(a_n)$ and $a_n^{\text{I}} \equiv \text{Im}(a_n)$ are determined by minimizing an error metric on the boundary Γ. More precisely,

$$\mathbf{a} \equiv [a_0^{\text{R}}, a_1^{\text{R}} + ia_1^{\text{I}}, \ldots, a_N^{\text{R}} + ia_N^{\text{I}}]^T \tag{15}$$

is determined by solving the minimization problem

$$\min_{\mathbf{a}} \left\|\widehat{U} - f\right\|_{\Gamma} \equiv \min_{\mathbf{a}} \left\langle \widehat{U} - f, \widehat{U} - f \right\rangle_{\Gamma}^{\frac{1}{2}}, \tag{16}$$

where the norm, $|| \cdot ||_\Gamma$, is induced by a Hermitian inner-product, $\langle \cdot, \cdot \rangle_\Gamma$, the definition of which is deferred to Section 3. The minimization in (16) is equivalent to requiring that the residual, $f - \widehat{U}$, be orthogonal to the span of the basis functions, or equivalently,

$$\left\langle \phi_n^{\mathrm{R}}, f - \widehat{U} \right\rangle_\Gamma = 0, \quad n = 0, 1, \ldots, N, \tag{17a}$$

$$\left\langle \phi_n^{\mathrm{I}}, f - \widehat{U} \right\rangle_\Gamma = 0, \quad n = 1, \ldots, N. \tag{17b}$$

Expanding the inner products and using (13) leads to the normal equations,

$$\begin{bmatrix} \mathbf{\Phi}^{\mathrm{RR}} & \mathbf{\Phi}^{\mathrm{RI}} \\ \mathbf{\Phi}^{\mathrm{IR}} & \mathbf{\Phi}^{\mathrm{II}} \end{bmatrix} \begin{bmatrix} \mathbf{a}^{\mathrm{R}} \\ \mathbf{a}^{\mathrm{I}} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_\phi^{\mathrm{R}} \\ \mathbf{f}_\phi^{\mathrm{I}} \end{bmatrix}, \tag{18}$$

where $\mathbf{\Phi}_{mn}^{\alpha\beta} = \left\langle \phi_m^\alpha, \phi_n^\beta \right\rangle_\Gamma$ and $(\mathbf{f}_\phi^\alpha)_m = \langle f, \phi_m^\alpha \rangle_\Gamma$.

Finding the basis function coefficients by least-squares minimization is common in desingularized methods. What is less common is our use of dipole potentials as basis functions, and also our use of a continuous error metric, as in (16). In Section 3 we will define a Hermitian inner product, and show that under mild conditions, enforcing the residual orthogonality (17) is equivalent to enforcing the boundary conditions at pole-location-related quadrature points.

2.2. Pole relocation

In the next step of the algorithm, the poles are moved to improve the approximation of $W(z)$. This is done by fitting a rational function to the complex function

$$\widehat{W}(z) = f(z) + i\widehat{V}(z), \quad z \in \Gamma \tag{19}$$

where $\widehat{V}(z)$ in (14) is based on the current set of poles. To obtain a "better" set of poles, we seek a rational approximation to $\widehat{W}(z)$ in the form $P(z)/Q(z)$ where both $P(z)$ and $Q(z)$ are polynomials of degree $\leq N$, and then use the zeros of $Q(z)$ as the "better" poles.

This rational fitting is done by minimizing a norm induced by the same inner-product used in (17). Ideally, we would want to minimize $||P/Q - \widehat{W}||_\Gamma$, but this is a nonlinear and non-convex problem. So, instead, we follow the commonly-used strategy of first scaling the residual $P/Q - \widehat{W}$ by Q, and then correcting this scaling by dividing by

$$\widehat{Q}(z) = \prod_{n=1}^{N} (z_n' - z), \tag{20}$$

which is an approximation for $Q(z)$ based on the current set of poles. The minimization problem is then

$$\min_{P,Q} \left\| \frac{P - Q\widehat{W}}{\widehat{Q}} \right\|_\Gamma, \tag{21}$$

where P and Q are represented with polynomial bases, denoted p_n and q_n, $n = 0, 1, \ldots, N$, respectively. To avoid the trivial solution, $Q(z)$ is assumed to be monic in the q_n basis.

The minimization in (21) leads to the orthogonality conditions,

$$\left\langle \frac{p_n}{\widehat{Q}}, \frac{P - Q\widehat{W}}{\widehat{Q}} \right\rangle_\Gamma = 0, \quad \left\langle \frac{q_n\widehat{W}}{\widehat{Q}}, \frac{P - Q\widehat{W}}{\widehat{Q}} \right\rangle_\Gamma = 0, \quad n = 0, 1, \ldots, N. \qquad (22)$$

The choice of polynomial bases determines the conditioning of (22), and, as the monomial basis is a particularly poor choice, other choices have been proposed [44–46].

The above approach to rational fitting has a long history, dating back to Levy [47] and Sanathanan and Koerner [48], and its application to real functions is much older. Usually, however, the minimization problem in (21) corresponds to fitting a rational function at a finite set of points on the imaginary axis or the unit circle. For that more common case, many implementations have been suggested, including vector fitting (VF), iterated rational fitting (IRF) [46, 49], and recently, robust rational least-squares [50]. We use VF as the default method, but in some cases it becomes poorly conditioned and we switch to a variant of IRF, the details of which are given in Appendix A.

The main difference between VF and IRF is the choice of basis polynomials. In VF, the choice is $p_n(z) \equiv q_n(z) \equiv \ell_n(z)$, where

$$\ell_n(z) = \begin{cases} \widehat{Q}(z) & \text{if } n = 0 \\ \dfrac{\widehat{Q}(z)}{(z'_n - z)} & \text{if } 1 \le n \le N. \end{cases} \qquad (23)$$

For $n = 1, 2, \ldots, N$, the $\ell_n(z)$ are, up to a scaling factor, the Lagrange polynomials for interpolating at the z'_n. As these polynomials are zero at all poles but one, they usually lead to a fairly well-conditioned linear system. However, if two or more poles are very close, conditioning may deteriorate. If this happens, we switch to using IRF, in which two polynomial bases are derived. For the $p_n(z)$, we use polynomials orthonormal with respect to the inner-product

$$\langle u, v \rangle_{\Gamma P} = \left\langle \frac{u}{\widehat{Q}}, \frac{v}{\widehat{Q}} \right\rangle_\Gamma, \qquad (24)$$

and for the $q_n(z)$ we use polynomials orthonormal with respect to the inner product

$$\langle u, v \rangle_{\Gamma Q} = \left\langle \frac{\widehat{W}u}{\widehat{Q}}, \frac{\widehat{W}v}{\widehat{Q}} \right\rangle_\Gamma. \qquad (25)$$

As a result of this choice, the linear system to be solved is made-up of two blocks of orthonormal columns, one for P and one for Q (see Appendix A).

Once $Q(z)$ is obtained, its roots are computed and they become the poles for the next iteration. Using these poles, $\widehat{U}(z)$, the approximate solution of (2), is recomputed by solving (18) and then using (13). Similarly, using (14), $\widehat{V}(z)$ is recomputed, $\widehat{W}(z)$ in (19) is updated, and the poles are relocated again.

When computing the roots of $Q(z)$ we may encounter roots inside Ω. Although we do not discard these poles, their corresponding basis functions are not used to form $\widehat{U}(z)$ and $\widehat{V}(z)$. Hence, they are tacitly excluded from (13)-(18). As the algorithm progresses, these poles usually migrate outside of Ω. Any poles that remain in Ω when the algorithm stops are discarded.

2.3. Initial guess for the poles

The proposed algorithm is found to be quite insensitive to the initial pole locations. We set them using the following heuristic, but even random initial locations seem to do fine. The initial guess for the z'_n is obtained by solving (22) assuming $\widehat{W}(z) = f(z)$, which is purely real, $\widehat{Q}(z) = 1$, and then allowing P and Q to be of degree $2N$. Out of the $2N$ zeros of Q we pick N zeros that are outside Ω, and, if there are not enough of these, we supplement them with zeros inside Ω so that we have N zeros in total. These N zeros are used as the poles for the first iteration. The justification for using $2N$ poles along with a purely real $\widehat{W}(z)$ is based on considering the fitting problem when Ω is a disk. For a disk, a purely real $\widehat{W}(z)$ can be constructed with N poles inside Ω and N poles outside Ω. Using only the poles outside, we obtain a function that is holomorphic in the disk and its real part fits $f(z)$ on Γ.

2.4. Algorithm summary

The main steps of the algorithm are as follows:

Algorithm 1 Solve simply-connected Dirichlet problem

1: Set the z'_n to the initial guess, as explained in Section 2.3.
2: **while** stopping criterion not met **do**
3:    Solve (18) for the a_n^{R} and a_n^{I}. Use only $z'_n \notin \Omega$.
4:    Use the a_n^{R} and a_n^{I} to form $\widehat{V}$ according to (14) and $\widehat{W}$ according to (19).
5:    Form $\widehat{Q}$ according to (20), use all z'_n.
6:    Solve (22) for P and Q.
7:    $z'_n = \mathrm{roots}(Q)$.
8: **end while**

---

3. Inner product definition and quadrature rules

In each iteration of Algorithm 1, two norm minimization problems must be solved: one to determine an approximate complex potential given the poles, (16), and one to update the poles using the approximate complex potential, (21). Both problems can be formulated as follows: given a set basis functions $\psi_1, \psi_2, \ldots \psi_{N_\psi}$ defined on Γ, and a function to approximate, g, also defined on Γ, we wish to determine the vector of coefficients $\mathbf{c}$ that solves the norm minimization problem

$$\min_{\mathbf{c}} \left\| \sum_{n=1}^{N_\psi} c_n \psi_n - g \right\|_\Gamma. \tag{26}$$

The unique solution to this problem is given by

$$\mathbf{c} = \mathbf{\Psi}^{-1} \mathbf{g}_\psi, \tag{27}$$

where $\mathbf{\Psi}_{mn} = \langle \psi_m, \psi_n \rangle_\Gamma$ and $(\mathbf{g}_\psi)_m = \langle g, \psi_m \rangle_\Gamma$. It can be easily shown that $\mathbf{\Psi}$ is Hermitian positive definite and hence invertible, provided that the ψ_n are linearly independent

(that this is so for the ϕ_n^{R} and ϕ_n^{I} follows from a trivial modification of Theorem 4 in [8], where it was proved in a 3D context). The inner product used in this work is

$$\langle u, v \rangle_\Gamma = \int_0^1 u\left(z(s)\right) v^*\left(z(s)\right) \lambda(s) ds, \tag{28}$$

where $\lambda(s)$ denotes a positive weight function, and $z(s)$ is a parametrization of Γ. In view of this definition, the inner-product integrals in $\mathbf{\Psi}$ and $\mathbf{g}_\psi$ may be evaluated using a quadrature rule. If (28) can be evaluated with a single quadrature rule for all u's and v's occurring in inner-product integrals in $\mathbf{\Psi}$ and $\mathbf{g}_\psi$, it is simpler and more numerically stable not to use (27) explicitly, but instead to sample $\sqrt{\lambda(s)}g(s)$ and the $\sqrt{\lambda(s)}\psi_n(s)$ at the quadrature nodes, and to set-up an overdetermined linear system for $\mathbf{c}$,

$$\begin{bmatrix} \sqrt{\lambda_1}\psi_1(s_1) & \cdots & \sqrt{\lambda_1}\psi_{N_\psi}(s_1) \\ \vdots & \ddots & \vdots \\ \sqrt{\lambda_{K_\psi}}\psi_1(s_{K_\psi}) & \cdots & \sqrt{\lambda_{K_\psi}}\psi_{N_\psi}(s_{K_\psi}) \end{bmatrix} \begin{bmatrix} c_1 \\ \vdots \\ c_{N_\psi} \end{bmatrix} \approx \begin{bmatrix} \sqrt{\lambda_1}g(s_1) \\ \vdots \\ \sqrt{\lambda_{K_\psi}}g(s_{K_\psi}) \end{bmatrix}, \tag{29}$$

where $s_1, s_2, \ldots, s_{K_\psi}$ are the quadrature nodes and $\lambda_1, \lambda_2, \ldots, \lambda_{K_\psi}$ are the quadrature weights. Equation (29) may also be written as

$$\mathbf{\Lambda}^{\frac{1}{2}}\widehat{\mathbf{\Psi}}\mathbf{c} \approx \mathbf{\Lambda}^{\frac{1}{2}}\widehat{\mathbf{g}}, \tag{30}$$

where

$$\begin{aligned} \mathbf{\Lambda} &= \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_{K_\psi}) \\ \widehat{\mathbf{g}} &= [g(s_1), g(s_2), \ldots, g(s_{K_\psi})]^T \\ \widehat{\mathbf{\Psi}}_{mn} &= \psi_n(s_m). \end{aligned} \tag{31}$$

The least-squares solution to (30) is given analytically by

$$\mathbf{c} = \left(\widehat{\mathbf{\Psi}}^H \mathbf{\Lambda} \widehat{\mathbf{\Psi}}\right)^{-1} \widehat{\mathbf{\Psi}}^H \mathbf{\Lambda} \widehat{\mathbf{g}}, \tag{32}$$

where superscript H denotes Hermitian transpose. It is more numerically stable, however, to obtain this solution by QR decomposition with column pivoting applied to (29). If the quadrature rule is exact, then the solution given in (32) also solves (26) (see, e.g., [22, Ch. 4]).

3.1. Quadrature rules for (16)

The basis functions used to solve (16), restricted to Γ and seen as a function of the parameter s, are the real and imaginary parts of

$$\phi_n^{\text{C}}(s) = \frac{1}{z'_n - z(s)}. \tag{33}$$

Clearly, constructing quadrature rules that will integrate inner products of these functions for arbitrary $z(s)$ is not practical. However, the parametrization of $z(s)$ is ours to choose,

and we choose it to be a piecewise rational function, i.e.,

$$
z(s) = \begin{cases} r_1(s) & \text{if } 0 \le s < h_1 \\ r_2(s) & \text{if } h_1 \le s < h_2 \\ \vdots & \\ r_M(s) & \text{if } h_{M-1} \le s < 1 \end{cases}, \tag{34}
$$

where the $r_m(s)$ are rational functions in s. This choice of parametrization is quite general, as piecewise rational functions can be used to represent any geometry of practical interest, and they are easily generated from Γ using readily available libraries (we use both Matlab's spline toolbox and the Chebfun system [51]). With this choice of parametrization, the $\phi_n^{\mathrm{C}}(s)$ are themselves piecewise-rational functions, each piece having known poles in the complex s plane. Quadrature rules for products of rational functions with known poles are readily available, and this is the motivation for our choice of parametrization[1]. However, the actual basis functions used in (16) are $\phi_n^{\mathrm{R}}(s)$ and $\phi_n^{\mathrm{I}}(s)$, and these are not piecewise-rational functions of s, as they depend on both s and s^*. To overcome this difficulty, we use the Schwartz reflection principle. We write the real basis functions as

$$
\phi_n^{\mathrm{R}}(s) = \frac{1}{2} \begin{cases} \dfrac{1}{z'_n - r_1(s)} + \dfrac{1}{[z'_n - r_1(s)]^*} & \text{if } h_0 \le s < h_1 \\[2ex] \dfrac{1}{z'_n - r_2(s)} + \dfrac{1}{[z'_n - r_2(s)]^*} & \text{if } h_1 \le s < h_2 \\[1ex] \qquad\qquad\qquad \vdots & \\[1ex] \dfrac{1}{z'_n - r_M(s)} + \dfrac{1}{[z'_n - r_M(s)]^*} & \text{if } h_{M-1} \le s < h_M \end{cases}, \tag{35}
$$

where $h_0 = 0$, $h_M = 1$. Note that the expression for ϕ_n^{I} is similar. Although ϕ_n^{R} and ϕ_n^{I} are functions of both s and s^*, for real s they can be written as piecewise rational functions in s alone. These rational functions can be obtained by replacing $[z'_n - r_m(s)]^*$ by $[z'_n - r_m(s^*)]^*$ everywhere in (35), and similarly for ϕ_n^{I}. We then have,

$$
\phi_n^{(\mathrm{R,I})}(s) = \frac{\displaystyle\prod_{l=1}^{L_m} \left(s - \beta_{lmn}^{(\mathrm{R,I})} \right)}{\displaystyle\prod_{l=1}^{L_m} (s - \alpha_{lmn})(s - \alpha_{lmn}^*)} \quad \text{for } s \in [h_m, h_{m+1}), \quad m = 0, 1, \dots, M-1, \tag{36}
$$

where L_m is the degree of the numerator or denominator of $r_m(s)$, whichever is higher. In (36), α_{lmn} is the lth pole associated with the nth basis function, restricted to the mth section of the boundary, i.e., it is the lth zero of $z'_n - r_m(s)$.

---

[1]Another advantage of this parametrization is that determining whether a pole is inside or outside of Ω is easy (see Appendix B).

Quadrature rules that are exact for the inner products of the basis functions are readily obtained using rational Gauss-Chebyshev quadrature rules as building blocks. These rules integrate exactly integrals of the form

$$\int_{-1}^{1} R(s)T^*(s^*)\frac{1}{\sqrt{1-s^2}}ds, \tag{37}$$

where $R(s)$ and $T(s)$ are rational functions that have poles at known locations (and need not be proper) [52]. Once the α_{lmn} are determined, a quadrature rule for each section of the boundary is calculated, in $O(NL_m)$ time, using the Matlab toolbox Rcheb [52]. The rules are then concatenated to obtain a single rule for the entire boundary.

Using these quadrature rules implies a specific choice for the inner-product weight function, denoted $\lambda(s)$ in (28), namely,

$$\lambda(s) = \begin{cases} \lambda_1(s) & \text{if } h_0 \le s < h_1 \\ \lambda_2(s) & \text{if } h_1 \le s < h_2 \\ \vdots & \\ \lambda_M(s) & \text{if } h_{M-1} \le s < h_M \end{cases}, \tag{38}$$

where

$$\lambda_m(s) = \frac{h_{m+1} - h_m}{\sqrt{(h_{m+1} - h_m)^2 - (2s - h_{m+1} - h_m)^2}}. \tag{39}$$

This choice of $\lambda(s)$ is dictated primarily by the ease with which rational Gauss-Chebyshev quadrature rules may be computed. However, it also has the advantage that when the boundary consists of a single rational-function section, using the Chebyshev weight function implies that the approximation will be close to optimal in the max-norm sense, provided the boundary data is smooth enough [52, 53]. When the boundary is composed of more than one section, this is no longer so, but the Chebyshev weight function still helps suppress Runge's phenomenon near $s = 0$ and $s = 1$. Lastly, this choice of weight function makes our error metric dependent on the parametrization $z(s)$. To avoid this, one could choose $\lambda(s) = |dz/ds|$, which would make the inner product a line integral and hence invariant to the choice of parametrization, but this choice would complicate the construction of the quadrature rules. A compromise is to make the weight of each section of the boundary proportional to its length, and this can be done with the chosen weight function by making the size of the interval $[h_m, h_m + 1]$ proportional to the length of the mth section of the boundary.

We still need to ensure that the inner-products between basis functions and the boundary data $f(z(s))$ are calculated accurately. Assuming the boundary data is smooth on each section of the boundary, this can be accomplished by making the quadrature rules exact for improper rational functions with a sufficiently high numerator polynomial degree. In Rcheb, this is done by adding poles at infinity when generating the quadrature rules.

3.2. Quadrature rules for problem (21)

The inner products for this second norm minimization problem are given in (22). All the integrands in this case can be written as rational functions of s, with poles due to

the zeros of $\widehat{Q}\left(z(s)\right)$ as well as the poles of $\widehat{W}\left(z(s)\right)$. Recalling the definition of $\widehat{W}\left(z(s)\right)$ given in (19), its real part is just the boundary data, and hence is assumed piecewise smooth, while the imaginary part has the same poles as those used for (16). To these poles we must add all zeros of $\widehat{Q}\left(z(s)\right)$, which are the zeros of all the $z'_n - r_m(s)$, including any poles $z'_n \in \Omega$. Although the quadrature rules are calculated quickly by Rcheb, it is usually more economical to call it only once, with the poles for (21) and then to solve both equations with the same set of quadrature points.

The proposed scheme for evaluating inner-product integrals with quadrature yields linear matrix equations very much like those of conventional desingularized methods, except that quadrature dictates the location and weight of each testing point. This scheme takes into account the location of the sources, the shape of the boundary, and the variation of the boundary data, and minimizes a continuous error norm.

The total number of testing points used, K, is given by

$$K = M(N_\infty + 1) + (3N_\text{out} + N_\text{in}) \sum_{m=1}^{M} L_m, \tag{40}$$

with the notation

M	number of rational-function sections of the boundary
N_∞	number of infinite poles added
N_out	number of poles outside of Ω
N_in	number of poles inside of Ω
L_m	degree of numerator or denominator of $r_m(s)$, whichever higher.

As the scheme is based on Gaussian quadrature, this number is in a sense optimal for computing with the continuous error norm, and the computation of the testing points is done with a fast and accurate library routine. We have found that, in all but the simplest cases, many uniformly distributed testing points are needed to match the accuracy obtained by the Gaussian quadrature scheme, so the extra computational burden is generally worthwhile. However, the representation of the boundary with piecewise rational functions should be made as simple as possible, i.e., with the fewest sections and lowest orders. If the boundary is broken into many sections, then taking into account all the poles when computing the quadrature rule for each section can be excessive. It is better to take into account only poles that are close to a given section. The rest of the poles can be dealt with more efficiently by a small number of poles at infinity. We refer to this process as pole pruning.

3.2.1. Pole pruning

Adding N_∞ poles at infinity makes the quadrature rule exact for integrands that are products of a rational function and a polynomial of degree $2N_\infty - 1$. Hence, we can omit poles that are distant enough that their corresponding potentials are smooth. To determine which poles are far enough, we need to estimate the error committed in approximating a rational function by a polynomial on the integration interval. As a

prototypical rational function we take a third-order pole function,

$$R_{s_0}(s) = \frac{1}{(s - s_0)^3}, \tag{41}$$

as this is the highest pole order that will appear in the integrands, and estimate the error committed in approximating it by a polynomial, as a function of the pole s_0. The max-norm error of the polynomial that interpolates at Chebyshev nodes is almost minimal and it is the same for all s_0 on the ellipse

$$s(\phi) = \frac{\rho e^{i\phi} + \rho^{-1} e^{-i\phi}}{2}, \quad \phi \in [0, 2\pi], \tag{42}$$

where $\rho > 1$ is the so-called *elliptical radius* (see, e.g., [54]). For any $s_0 \notin [-1, 1]$ we can calculate the elliptical radius corresponding to the ellipse which passes through it,

$$\rho(s_0) = \max\left(\left|s_0 + \sqrt{s_0^2 - 1}\right|, \left|s_0 - \sqrt{s_0^2 - 1}\right|\right). \tag{43}$$

Although there are bounds for the error in approximating a function such as $R_{s_0}(s)$ by a polynomial, we found it more practical to tabulate the degrees of polynomials that yield approximation errors less than ϵ, for relevant values of ϵ and $\rho(s_0)$ (Fig. 3). To obtain
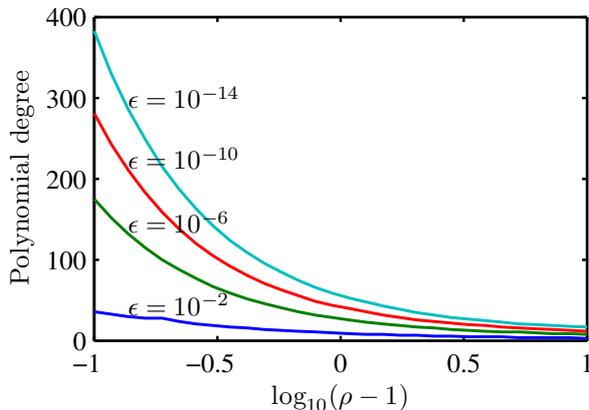
Figure 3: Polynomial degree that yields approximations of $R_{s_0}(s)$ with errors less than ϵ, as a function of ρ. For $\rho \approx 10$, errors under 10^{-14} can be obtained with polynomial degrees under 20.

these polynomial degrees, we again used the Chebfun system, which truncates a Chebyshev expansion once the expansion coefficients drop sufficiently below ϵ. The tabulation is then interpolated or extrapolated as necessary so that we can rapidly calculate the optimal ρ beyond which poles can be discarded and replaced by poles at infinity, without incurring errors much larger than ϵ. As shown in Fig. 3, for all but the most distant poles, the choice of ϵ influences the required polynomial degree appreciably. Hence, if only moderate accuracy is sought, significant reduction may be gained by using a larger

14

ϵ. To take advantage of this, we set ϵ according to

$$\epsilon = \min(10^{-4}, 0.01\Delta E), \tag{44}$$

where ΔE is the normalized error obtained when solving (17) in the previous iteration (a more precise definition is given below in (58)). Thus, if the solution is not very accurate, fewer quadrature nodes are used, but their number is increased if the algorithm is allowed to iterate and higher accuracy is achieved. In the very first iterations, ΔE can be large, so we bound epsilon from above by 10^{-4}. Of course, this rule is heuristic and has been derived from numerical experiments in which it appeared to do well, but it is by no means optimal and a different rule could perform better.

3.2.2. Froissart doublets

Admittedly, the quadrature scheme just described is somewhat complicated, but we have found this complication to be worthwhile. The reason is related to the occurrence of pole-zero pairs, called Froissart doublets [55–57], which almost coincide with the boundary. These pole-zero pairs are a numerical artifact, and they do not contribute to the fidelity of the fit, but instead they fit errors in the data. In our case, the errors in the imaginary part of the data are quite large in the first iterations, so these artifacts occur often. If uniformly distributed testing points are used, the doublets frequently occur closer to the boundary than the separation between testing points. When the poles of these doublets are used to solve (16), the resulting potential is inaccurate between testing points. In turn, the estimate of $\text{Im}[W(z)]$ is inaccurate, and the improvement in accuracy from one iteration to the next may stall. One could try to filter these poles in various ways, for example, based on their small residues and distances from the boundary, but this would limit the obtainable accuracy. By using the proposed quadrature scheme, all the poles outside of Ω, no matter how close they are to its boundary, can be safely used to solve (16).

In Fig. 4a, an artifact caused by a Froissart doublet can be observed near the top of Γ. The pole of the doublet is extremely close to the boundary (the distance is about 10^{-9} of the perimeter of Γ), which explains the appearance of the artifact despite having used 2000 uniformly distributed testing points. When the quadrature scheme is used, with just 203 points, the artifact disappears (Fig. 4b).

4. Dirichlet problem, multiply-connected domain

Other Laplace BVPs can also be solved with the proposed scheme, after appropriate adaptations. As an example of a Dirichlet problem in a multiply-connected domain we consider the exterior Dirichlet problem; multiply-connected interior problems can be treated similarly. In the exterior Dirichlet problem, the potential $U(x, y)$ obeys the Laplace equation everywhere outside of a set of disjoint domains $\Omega_1, \Omega_2, \ldots, \Omega_J$ bounded by contours $\Gamma_1, \Gamma_2, \ldots, \Gamma_J$, respectively. As before, Dirichlet boundary conditions are imposed,

$$U|_{\text{on } \Gamma_j} = f_j, \quad j = 1, 2, \ldots J \tag{45}$$

and in addition, the potential must be bounded at infinity. Of course, the poles must now be restricted to lie inside the Ω_j. Also, the complex potential $W(z)$ is no longer

<div style="text-align:center">(a)   (b)</div>
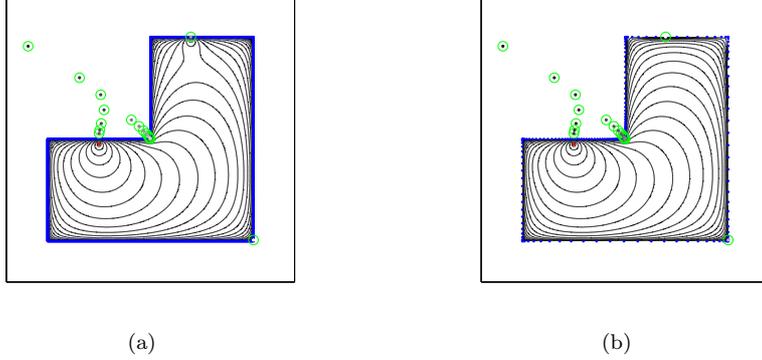
Figure 4: Equipotential lines for a monopole (marked by a red dot) inside an L-shaped region. In (a), 2000 points were uniformly distributed on Γ, in (b), 203 points were distributed by the quadrature scheme.

single-valued, but can be written as the sum of J logarithmic terms and a single-valued holomorphic function $\widetilde{W}(z)$ (see, e.g., [37, §29]). We have,

$$W(z) = \widetilde{W}(z) + \sum_{j=1}^{J} A_j \log\left(z - z_j^{\Omega}\right), \quad z_j^{\Omega} \in \Omega_j, \tag{46}$$

where the A_j may be assumed real, and the z_j^{Ω} can be anywhere inside Ω_j, preferably as far from its boundary as possible. If the A_j were known, the logarithmic terms could be subtracted from the boundary data, and the algorithm for the simply-connected Dirichlet problem could be applied to $\widetilde{W}(z)$. Our approach is therefore to estimate the A_j iteratively, along with the imaginary part of $\widetilde{W}(z)$. On each iteration, we solve

$$\sum_{n=0}^{N_{\text{out}}} a_n^{\text{R}} \phi_n^{\text{R}}(z) - \sum_{n=1}^{N_{\text{out}}} a_n^{\text{I}} \phi_n^{\text{I}}(z) + \sum_{j=1}^{J} A_j \log\left|z - z_j^{\Omega}\right| \approx f(z),$$

$$\text{with } f(z) = f_j(z) \quad \text{for } z \in \Gamma_j, \tag{47}$$

$$\text{subject to } \sum_{j=1}^{J} A_j = 0,$$

for the a_n^{R}, the a_n^{I}, and the A_j. The condition that the sum of A_j's is zero, needed to ensure the solution is bounded at infinity, can be enforced by eliminating one of the A_j, say, $A_J = -\sum_{j=1}^{J-1} A_j$. Instead of (47) we then have,

$$\sum_{n=0}^{N_{\text{out}}} a_n^{\text{R}} \phi_n^{\text{R}}(z) - \sum_{n=1}^{N_{\text{out}}} a_n^{\text{I}} \phi_n^{\text{I}}(z) + \sum_{j=1}^{J-1} A_j \log\left|\frac{z - z_j^{\Omega}}{z - z_J^{\Omega}}\right| \approx f(z). \tag{48}$$

As before, we solve (48) by sampling at quadrature nodes, weighting the equation by the square roots of the quadrature weights, and solving the linear system in the least-

<div style="text-align:center">16</div>

squares sense. In contrast to the previous case, however, the inner products involving the logarithmic basis functions will not be integrated exactly. We must therefore add poles at infinity when deriving the quadrature rules for each section of the boundary. Fortunately, the logarithmic singularity is weak, so when the z_j^Ω are not too close to the boundary, the restriction of the logarithmic term to the boundary curve can be approximated well by piecewise polynomials. The poles at infinity added by the pole-pruning process of Section 3.2.1 usually suffice, although extra poles at infinity can be added if necessary.

Once (48) has been solved, a rational function is fitted to an estimate of the boundary values of $\widetilde{W}(z)$,

$$\widetilde{W}(z) \approx f(z) - \sum_{j=1}^{J-1} A_j \log \left| \frac{z - z_j^\Omega}{z - z_J^\Omega} \right| + i\widehat{V}(z), \quad z \in \bigcup_{j=1}^{J} \Gamma_j = \Gamma. \tag{49}$$

The real part is estimated as $f(z)$, the known values of $\mathrm{Re}[W(z)]$ on the boundaries, with the real part of the logarithmic terms subtracted. The estimate for $\mathrm{Im}[\widetilde{W}(z)]$ is $\widehat{V}(z)$, given by (14), and it remains as it was for the simply-connected problem when $\widetilde{W}(z)$ and $W(z)$ were the same.

5. Exterior Neumann problem

Now consider the case when the normal derivative of the potential on the boundary is prescribed. For brevity, we discuss only the exterior problem, which can be formulated as

$$\nabla^2 U(x, y) = 0, \quad \text{for all } (x, y) \notin \bigcup_{j=1}^{J} \Omega_j,$$

$$\left. \frac{\partial U}{\partial n} \right|_{\text{on } \Gamma_j} = f_j, \tag{50}$$

$$\lim_{|z| \to \infty} U(x, y)|z| < \infty, \quad z = x + iy,$$

where the Ω_j are again simple disjoint domains bounded by the Γ_j. This problem is uniquely solvable if and only if

$$\sum_{j=1}^{J} \int_{\Gamma_j} f_j(z)|dz| = 0. \tag{51}$$

The approach we adopt for this problem follows [37, §32]. As in the exterior Dirichlet problem, we seek the solution as the real part of a function $W(z)$ that is holomorphic outside the Ω_j, and which can be written as a sum of a single-valued part, $\widetilde{W}(z)$, and J logarithmic terms. In this case, however, the logarithmic terms are known in advance:

$$A_j = -\frac{1}{2\pi} \int_{\Gamma_j} f_j(z)|dz|, \tag{52}$$

and they can be subtracted from the boundary data. After this subtraction,

$$\widetilde{f}_j(z) = f_j(z) - \sum_{j=1}^{J} A_j \log \left| z - z_j^J \right|, \tag{53}$$

can be used to write a Neumann boundary condition for $\mathrm{Re}\left[\widetilde{W}(z)\right]$. Denoting $\widetilde{W}(z) = \widetilde{U}(z) + i\widetilde{V}(z)$, this condition reads,

$$\left. \frac{\partial \widetilde{U}}{\partial n} \right|_{\text{on } \Gamma_j} = \tilde{f}_j. \tag{54}$$

If $\widetilde{W}(z)$ is approximated by a weighted combination of dipole potentials, it becomes inconvenient to enforce (54) directly, as the differentiation increases the order of the poles. Instead, we integrate (54) with respect to arc-length and use the Cauchy-Riemann equations to obtain,

$$\widetilde{V}(z) = \int_{z_j}^{z} \tilde{f}_j(z)|dz| + v_j \equiv F_j(z) + v_j, \tag{55}$$

where the integration is along Γ_j, from an arbitrary point z_j to the point z, and the v_j are constants yet to be determined. Finding a harmonic function $\widetilde{V}$ that satisfies the boundary condition (55) is known as a modified Dirichlet [37] or Muskhelishvili [58] problem, and given the condition at infinity in (50), it is solvable only for a unique set of v_j constants. These can be determined along with an approximate solution for $\widetilde{V}$, by solving

$$\sum_{n=1}^{N_{\text{out}}} a_n^{\mathrm{R}} \phi_n^{\mathrm{I}}(z) + a_n^{\mathrm{I}} \phi_n^{\mathrm{R}}(z) - v_j \approx F_j(z), \quad \text{for } z \in \Gamma_j, \tag{56}$$

for the a_n^{R}, the a_n^{I}, and the v_j. As before, (56) is solved in the least-squares sense after sampling the basis functions at quadrature nodes and weighting the equations by the square roots of quadrature weights. Once (56) is solved, the rational fitting algorithm is applied to an estimate of the boundary values of $\widetilde{W}(z)$ given by,

$$\widetilde{W}(z) \approx \sum_{n=1}^{N_{\text{out}}} a_n^{\mathrm{I}} \phi_n^{\mathrm{I}}(z) + a_n^{\mathrm{R}} \phi_n^{\mathrm{R}}(z) + i \left[F_j(z) + v_j \right], \quad \text{for } z \in \Gamma_j. \tag{57}$$

As expected, in the Neumann problem, it is the *imaginary* part of the complex potential that is prescribed by the boundary data (up to J additive constants), while its real part and the J constants are estimated iteratively. Note that the $n = 0$ term corresponding to the constant basis function is omitted from the sum in (57) so as to conform with the boundary condition at infinity in (50).

The formulation just described differs from one that is commonly used for Neumann problems in integral-equation formulations (e.g., in [38]). These formulations typically use a single-layer potential so that enforcing the boundary condition of (50) leads to a second-kind Fredholm equation. We could have formulated our solution similarly, i.e., by

representing the potential as a weighted combination of monopole potentials. Enforcing the boundary condition would then lead to a weighted combination of dipole potentials approximating the normal derivative of U. The chosen formulation does have some advantages, however. First, the functions to be approximated, the $F_j(s)$, are smoother than the $f_j(s)$, so fewer basis functions may be needed for their approximation. Second, a common situation in Neumann problems is that the A_j of (52) are zero, so that the complex potential $W(z)$ is single valued. This occurs when the boundary is composed of a single contour, or when the Ω_j represent impenetrable objects. In these cases, it is convenient to approximate $W(z)$ using single-valued basis functions, instead of having a superposition of non-single-valued functions that can be made single-valued by an appropriate choice of branch cuts, as would occur if monopoles were used. For instance, using the proposed formulation, it is easy to obtain equipotential contours and field lines as the level sets of $\text{Re}\,[W(z)]$ and $\text{Im}\,[W(z)]$, without having to keep track of the proper Riemann-sheet on which to evaluate the logarithmic kernel.

6. Large-scale problems

The proposed algorithm for setting the poles and testing-points works well for moderately-sized geometries, but its computational cost is too high for larger ones. However, it is still possible to tackle large-scale problems by partitioning the geometry into smaller domains and running the algorithm for each of these domains separately, possibly in parallel. Once the poles and testing points are obtained for the entire geometry, we solve the full problem with the locally-optimized poles and testing points. If the cost of solving the full problem becomes prohibitive, it may be possible to use an iterative domain-decomposition approach, along the lines of [59], [60], or [61], which could also be used to improve the locations of the poles and testing points from one iteration to the next.

We propose a scheme for geometries made up of a large number of simple closed curves. The scheme consists of partitioning the boundary curves into clusters so that each curve belongs to a single cluster, as shown in Fig. 5. Various algorithms for obtaining such a partition exist, and we use the simplest one, the k-means clustering, or Lloyd's, algorithm [62]. To keep the partitioning simple, each curve is entirely within one cluster and represented by its centroid, on which the clustering algorithm operates. Then, to each cluster we add a "buffer region," as shown in Fig. 5, so that curves close to the edge of the cluster have all their nearby interactions taken into account. The buffer region is obtained by adding to each curve all the curves with centroids closer than some radius, and this radius is chosen as twice the distance to the nearest centroid among all other curves. Poles that, after the local optimization, are found inside the buffer region are not used when solving the full problem.

7. Numerical results

The proposed method was implemented in Matlab, and results for a number of sample problems were obtained. This implementation, which is freely available for download [63], was run on a 2GHz Intel Xeon machine with 8GB of RAM. The error metric adopted in

19
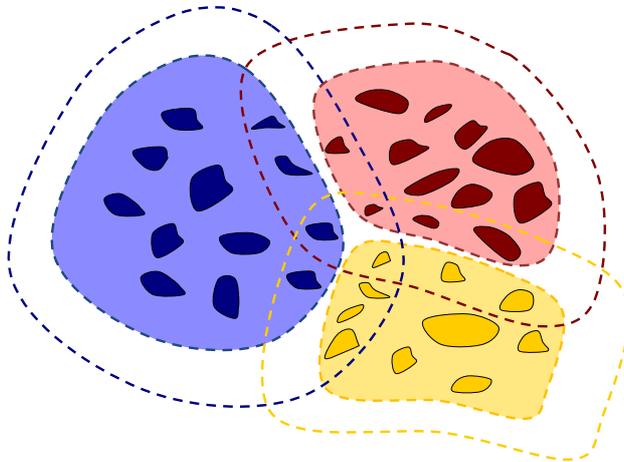
Figure 5: Boundary curves partitioned into three clusters with buffer regions (delimited by dashed lines).

this work is ΔE,

$$\Delta E = \frac{\|\hat{\mathbf{u}} - \mathbf{f}\|_w}{\|\mathbf{f}\|_w}, \tag{58}$$

where $\mathbf{f}$ and $\hat{\mathbf{u}}$ are vectors of samples at the quadrature points of the exact and approximate boundary values, respectively. The $\|\cdot\|_w$ norm includes multiplication by the quadrature weights, so that it is an approximation of the continuous norm. The residual, $\widehat{U} - U$, is itself a harmonic function throughout $\overline{\Omega}$ and hence is largest at the boundary. We also used the maximum-norm error,

$$\Delta E_{\mathrm{max}} = \frac{\max\limits_{z \in \Gamma} \left| f(z) - \widehat{U}(z) \right|}{\|f(z)\|_{L^2(\Gamma)} / \|1\|_{L^2(\Gamma)}}, \tag{59}$$

where $\|1\|_{L^2(\Gamma)}$ is just the length of Γ. In practice, we estimated ΔE_{max} by sampling f and $\widehat{U}$ on a large number of points on the boundary.

7.1. Interior Dirichlet Problem

Problems of increasing difficulty were considered. Where applicable, results were compared to results obtained with the standard Nyström method, implemented along the lines of [38]. However, we have not compared with more sophisticated Nyström schemes, such as those of [1, 2, 5, 6], which have been recently proposed for handling more "difficult" geometries and excitations.

7.1.1. A few well-separated poles

We begin with the simplest problems, in which the exact solution is known in advance to have just a few, well-separated, poles, i.e.,

$$f(z) = \mathrm{Re} \left[\sum_{k=1}^{N_0} \frac{a_k^0}{z_k' - z} \right], \quad z_k' \notin \Omega, \quad z \in \Gamma \tag{60}$$

20

Although this excitation suits our basis functions perfectly, the solution obtained is not always very accurate even if $N = N_0$. If the estimate of $\text{Im}\,[W(z)]$ is inaccurate enough, the poles may be relocated to poor positions, and using these poles will not lead to a more accurate estimate of $\text{Im}\,[W(z)]$. This problem can be alleviated by using $N > N_0$, though the number of poles that must be added beyond N_0 depends on the location of the poles of the solution and also on the shape of the boundary. This dependence is explored below.

A plot of error vs. iteration count is shown in Fig. 6, for an ellipse with unit semi-major axis and an aspect ratio of two, and boundary data as in (60) with $N_0 = 6, a_k^0 = k/N_0$, and $z_k' = 1.5 \exp\left[i\pi(2k + 1)/6\right]$. When $N = N_0$, the error does not decrease
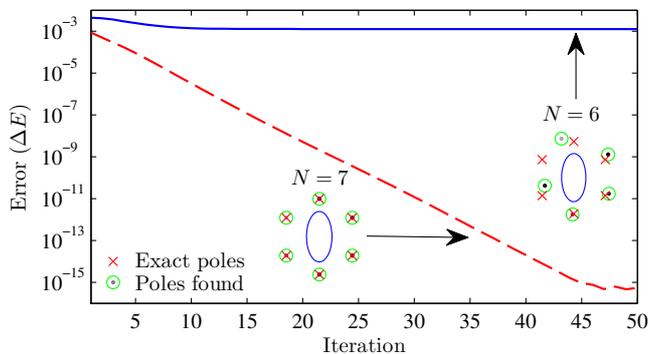
Figure 6: Error vs. iteration for an ellipse and 6-pole boundary data. One of the poles remains inside the ellipse, both for $N = 6$ and $N = 7$.

much below 0.1%, but when $N = N_0 + 1$, the error drops exponentially, approaching the machine epsilon after 45 iterations. The reason for this dramatic difference in accuracy is that when $N = N_0$, a single pole remains inside Ω and the estimate of $\text{Im}[W(z)]$ obtained with the remaining five poles is not accurate enough to make it migrate outside of Ω. Consequently, the improvement in accuracy stalls. In contrast, when $N = N_0 + 1$, the extra pole allows for a better approximation of the imaginary part and six poles appear outside of Ω. They are then relocated until the error approaches the machine epsilon.

A natural possibility is to increase N adaptively, whenever the improvement in accuracy stalls. For example, one can calculate $\sigma_K(\Delta E)$, the standard deviation of ΔE in the previous K iterations, and add a pole whenever

$$\sigma_K(\Delta E) < \mu_K(\Delta E)\epsilon, \tag{61}$$

where $\mu_K(\Delta E)$ is the K-average of ΔE and ϵ is a small number. A more sophisticated strategy might be found along the lines of [57], which could also be used to decrease the number of poles if necessary. We have implemented the simple strategy with $K = 5$ and $\epsilon = 0.5\%$, which appears to yield a practical tradeoff between the conflicting goals of only adding poles when necessary, and keeping the number of iterations moderate. This adaptive scheme, which makes the algorithm fully automatic, is available as part of the downloadable package [63]. However, to keep the results section concise, we only show results for the nonadaptive version.

21

We consider next boundary curves of the form,

$$z(s) = [1/\gamma + (1/\gamma - 1)\cos(2\pi\nu s)]\, e^{j2\pi s}, \quad s \in [0,1] \tag{62}$$

with $\gamma = 1.75$, and $\nu = 2, 4, 6$. A plot of the $\nu = 6$ curve is shown in Fig. 7, together with a set of ten points distributed on a circle of radius R, which represent the poles of the boundary data. This radius was set to 1.1, 1.5, and 3, while the number of poles was
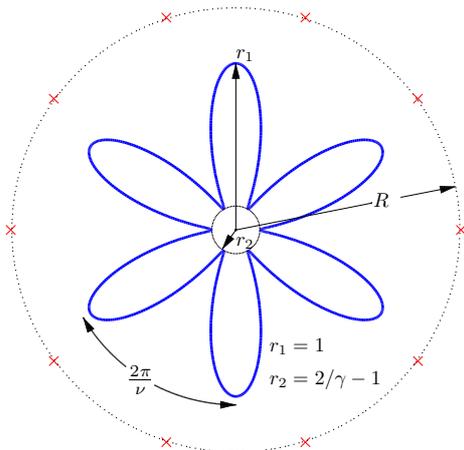
Figure 7: Boundary curve and poles of boundary data used in Table 1.

kept to ten. The algorithm was run for a maximum of 100 iterations and we determined the smallest N required so that $\Delta E < 10^{-14}$ when the algorithm stops. The results of this experiment are summarized in Table 1 where this value of N is given, together with $\Delta E_{\max}$, estimated by evaluating the approximate solution at 20000 boundary points. In

Table 1: Results for boundary curves as in Fig. 7.

ν R	2		4		6	
	$\Delta E_{\max}$	N	$\Delta E_{\max}$	N	$\Delta E_{\max}$	N
1.1	8.8×10^{-15}	10	1.4×10^{-14}	11	1.6×10^{-14}	14
1.5	3.1×10^{-15}	18	5.6×10^{-15}	22	8.3×10^{-15}	18
3	4.3×10^{-15}	25	7.8×10^{-15}	27	1.4×10^{-14}	29

all cases, $\Delta E_{\max}$ is of the same order of magnitude as the tolerance for ΔE, although in a few cases it is slightly larger than 10^{-14}. We see that, in contrast to conventional desingularized- and integral-equation- methods, the proposed method works best when the poles of f are close to the boundary, adding only a few spurious poles during the solution. Evidently, when the poles are further away their locations are more difficult to ascertain. We note that ν has only minor influence on N.

7.1.2. Almost-singular boundary data

When the boundary data is singular, or almost so, the accuracy of most numerical methods suffers. A possible remedy is to add basis functions that can approximate the singular boundary data to the set of basis functions being used (for an example in the context of desingularized methods, see [64]). In the proposed method this happens automatically, as the basis functions are chosen from a large over-complete set. In the next example, the boundary is an ellipse with unit semi-minor axis, and an aspect ratio of two, and the boundary data is,

$$f(z) = \operatorname{Re}\left\{\exp\left[\frac{1}{z - 1.01}\right]\right\}, \quad z \in \Gamma. \tag{63}$$

This implies that the exact complex potential has an essential singularity very close to the boundary. Nevertheless, errors on the order of $\Delta E_{\max} = 10^{-11}$ are obtained with no more than 25 poles (Fig. 8). These errors could not be reduced further, and in fact, as
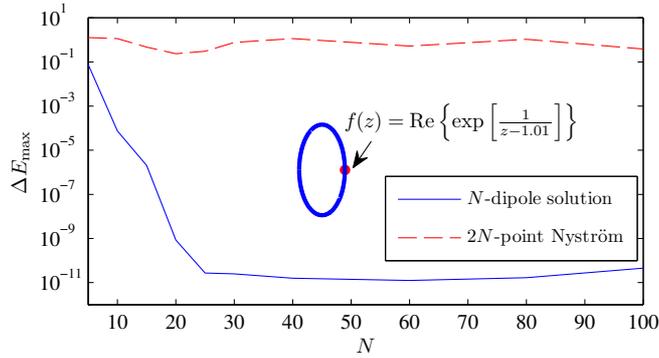
Figure 8: Almost-singular boundary data. Comparison of the error vs. number of basis functions ($2N$) obtained with the proposed method and the Nyström method.

more poles are added the error increases slightly. Compared to the proposed algorithm, the Nyström method yields very large errors, and its convergence rate is very slow. For 200 collocation points the Nyström method yields $\Delta E_{\max} = 0.38$, and, outside the range of the figure, 1000 collocation points still yield only $\Delta E = 0.05$.

Of course, for the same number of basis functions, the computation time of the Nyström method is shorter by a factor related to the number of pole relocation iterations. Even so, the Nyström method is more time-consuming in this example for any reasonable accuracy. For example, for $\Delta E = 0.05$, the Nyström method takes about 6s, while the proposed method takes 2.5s and it uses five poles. For higher accuracies the differences increase and it quickly becomes impractical to use a direct solver with the Nyström method. If the boundary data is smoother, for example, if the essential singularity is moved to $z = 1.1$, the Nyström method can be faster. It requires 0.9s and $N = 400$ for $\Delta E = 10^{-5}$, whereas the proposed method requires 2.5s and five poles to achieve the same accuracy.

At each point on the error curves in Fig. 8 we calculated the error of an N-pole solution and a $2N$-point Nyström solution, so as to match the number of basis functions. The

maximum error for both the Nyström method and the proposed method was calculated by evaluating the approximate potential on $8N$ boundary points. To obtain this potential in the Nyström method we used a $8N \times 8N$ collocation matrix, which multiplies the double-layer density interpolated trigonometrically for its values at the $8N$ points.

7.1.3. Almost-sharp corners

The influence of the shape of the boundary in the previous examples is limited to determining where the solution is sampled; it does not influence the location of poles of the solution as these are set by the boundary data. In the next example, we consider a Green's-function type problem, where a boundary curve of form (62) with $\nu = 2$ is excited by a monopole potential,

$$f(s) = \log|z(s)|. \tag{64}$$

As the parameter $\gamma \to 2$ in (62) the problem becomes more difficult, because the minimal radius of curvature decreases, and also the sharp tips of the boundary approach the monopole and each other. A solution of this problem with $\gamma = 1.9$, obtained with 35 poles is shown in Fig. 9. The inhomogeneous distribution of poles and testing points is
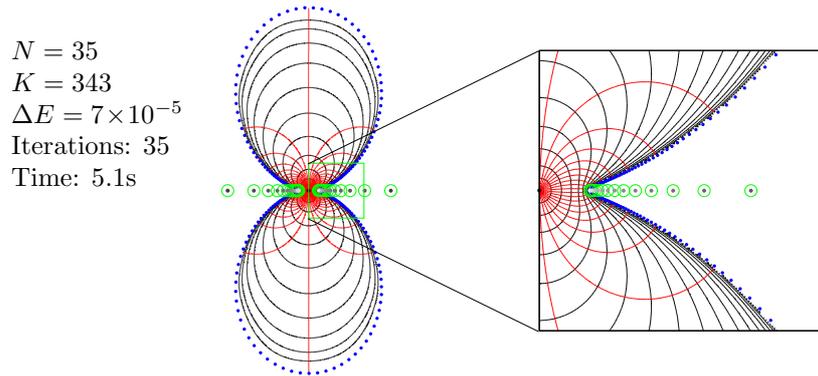
Figure 9: Equipotential contours (logarithmically spaced) and field lines for a monopole inside the curve: $z(s) = [1/1.9 + (1/1.9 - 1)\cos(4\pi s)]\, e^{j2\pi s}$. The poles of the solution are marked by green circles, with centers colored according to the magnitude of the residues (darker means larger residues). The testing points are marked by blue dots.

clearly necessary in order to approximate the rapidly varying field where the radius of curvature is small. To calculate the testing points, the boundary was parameterized by a single polynomial in s, of degree 34, which approximates $f(s)$ to machine precision. Equipotential contours and field lines are also shown in Fig. 9. These were obtained as the level sets of $\mathrm{Re}\,[W(z)]$ and $\mathrm{Im}\,[W(z)]$ by use of Matlab's 'contour' command. The evaluation of $W(z)$ throughout Ω is trivial, as it only requires the summation of the partial-fraction expansion for $W(z)$. This may be contrasted with the more elaborate schemes needed to evaluate integral equation solutions throughout the domain, especially close to the boundary (see, e.g., [1]).

Plots of ΔE_{max} vs. N are shown in Fig. 10 for curves with $\gamma = 1.5, 1.75, 1.9$, together with results obtained with the Nyström method. The points for the Nyström method

24
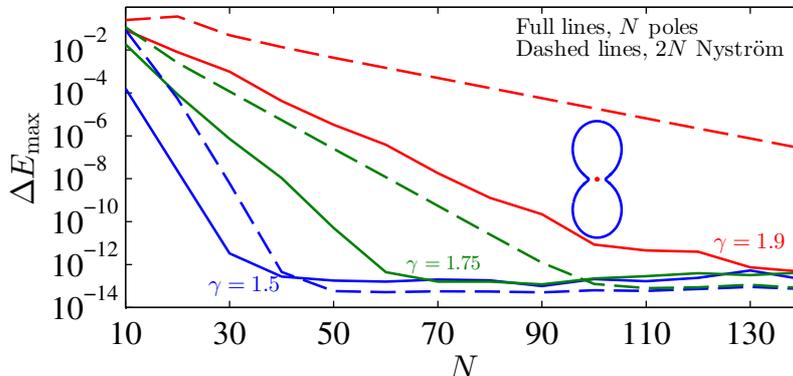
Figure 10: Comparison of the proposed algorithm and the Nyström method for boundary curves with decreasing minimal radius of curvature. The difference between the methods is more significant for smaller minimal radius of curvature.

were distributed uniformly in the parameter s of (62), implying a higher density near the origin, which helps in resolving the rapid variation in the double-layer density there. The proposed algorithm was allowed to run for a maximum of 100 iterations, though the poles stopped moving after roughly 50 iterations. The rate of convergence with respect to the number of poles is approximately exponential, and the minimal error obtained is similar to that of the Nyström method. It is also evident that the proposed method generally yields a more accurate solution per basis function than the Nyström method, and that the difference between the two increases with γ.

7.1.4. Perfectly sharp corners

When the boundary has sharp corners, the potential may have singularities at the corners, making it impossible to represent with simple poles alone. It is still possible, however, to obtain very good approximate solutions with a moderate number of poles. Such an example is shown in Fig. 1b, and another one is shown in Fig. 11. As can be observed in both figures, the poles and testing points cluster near the reentrant corners, and appear to be approximating a branch-cut which exists in the analytical continuation of the solution (see [43] for a discussion of the curves on which poles cluster when approximating algebraic functions by rational functions). The distribution of poles near the reentrant corner is examined in Fig. 12, where the poles are plotted in descending order of distance from the corner, for a solution with 82 poles of the L-shaped boundary. The distances, which are normalized with respect to the distance of the exciting line-source from the corner, are seen to decrease at an almost constant exponential rate, with the closest poles clustered even more densely. The closest pole is placed at a normalized distance of about 10^{-11}, indicating that poles must be located very precisely if high accuracy is to be achieved. Moreover, it seems inevitable that the testing points must also be adapted so that they are located with comparable precision. One might wonder whether the sharp turn in the equipotential lines near the corner is achieved with poles having large residues of differing phases. As shown in Fig. 12, this is not the case. Although the residues have absolute values that span 19 orders of magnitude, the smallest

25

$$N = 30$$
$$M = 230$$
$$\Delta E_{\max} = 1.8 \times 10^{-5}$$
Iterations: 60
Time: 3.2s

$$\Delta E_{\max} = 1.8 \times 10^{-5}$$
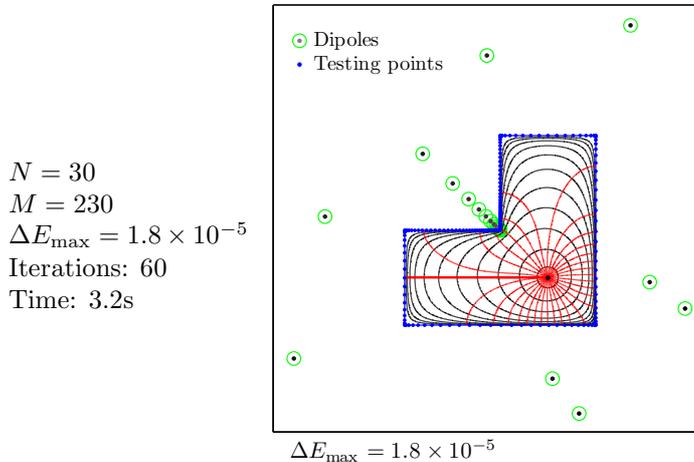
Figure 11: Equipotential contours (logarithmically spaced) and field lines for a monopole inside an L-shaped domain.

ones correspond to the closest poles.

We now examine the convergence of the algorithm with N (Fig. 13). As seen in Fig. 13a, the dependence of the error on N is not as smoothly exponential as in the previous examples. This more uneven error decrease is due to inaccuracies in the imaginary part that result in poles converging to locations inside Ω. For example, when $N = 20$ four of the poles remain inside Ω, while when $N = 30$ all poles move outside of Ω. As a result, ΔE decreases by more than two orders of magnitude when going from $N = 20$ to $N = 30$. As shown in Fig. 13b, the condition number increases with N, but the norm of the vector of residues (normalized with respect to N) decreases. This is similar to [25], in which such behavior was observed when the sources enclosed all the singularities of the analytic continuation of the potential. It is remarkable that this occurs in the present example as the potential cannot be continued analytically beyond the corners.

7.2. Exterior Dirichlet problem

The exterior Dirichlet problem differs from the interior problem by the presence of logarithmic terms in the former. Also, the region in which the poles may be placed in the exterior problem is bounded, which probably makes their location easier. A typical difficulty in exterior problems occurs when the boundary is composed of two or more curves that nearly touch. An example of this sort consists of two circular curves with boundary values -1 on the lower curve and $+1$ on the upper one, and relative separations $d/R = 0.01, 0.1, 0.5$, as shown in Fig. 14. This example has an analytic solution [65] to which we may compare the results. Specifically, seen as a two-conductor electrostatic problem, a global parameter of interest is the capacitance, proportional to the magnitude of the logarithmic terms, and a local parameter of interest is the charge density, proportional to the normal derivative of the potential. As can be seen in Fig. 14, for both of these parameters, the proposed method requires far fewer degrees of freedom than the Nyström method, and this difference increases as the conductors approach.
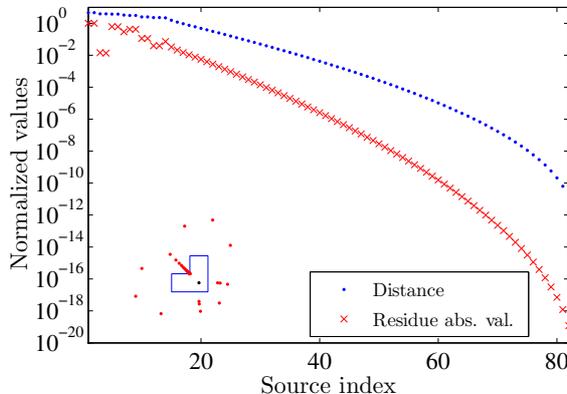
Figure 12: Distances to reentrant corner, normalized with respect to the distance of the exciting line-source from the corner, and corresponding absolute value of residues, normalized with respect to the largest one.

Also of note is that the proposed method is far less sensitive to the separation between conductors. To a good approximation, the N needed to achieve a given error scales like $1/d$ when using the Nyström method, but it scales like $\log(1/d)$ when using the proposed method.

7.3. Large-scale problems: exterior Neumann example

In the last example, we solve a Neumann problem of potential flow around impenetrable ellipses of random radius, eccentricity, and orientation, many of which are very close to one another. This geometry was generated by a simple modification of the method described in [1] for generating a distribution of random circles. The velocity field far from the ellipses is assumed uniform, of unit amplitude, and in the x direction. The corresponding BVP is then given by (50) with $f_j(z) = -\cos[\theta_j(z)]$ where $\theta_j(z)$ is the angle between the outward-pointing normal to the jth curve and the x axis.

We use the partitioning scheme of Section 6 and set the number of clusters so that each cluster has only a few ellipses. Thus, the run-time and memory requirements for each cluster are kept very moderate. For example, for a geometry with 100 ellipses, we used ten clusters, with the largest cluster having 13 ellipses (not including the ones added as buffers). For each cluster, we used three poles per ellipse (including the ones used as buffers), and stopped the algorithm if the error dropped below 1% or 100 iterations were exhausted. The algorithm was run in parallel on four cores of a desktop PC, using Matlab's parallel computing toolbox, and the run-time for determining the pole and testing point locations was 87s. We then assembled a linear system of size 4832×706 with the poles and testing-points of all clusters and solved it by QR decomposition in 5s. The error of this solution was $\Delta E = 0.013$, slightly higher than the error tolerance for each cluster. The largest geometry solved, which consisted of 300 ellipses, is shown in Fig. 15, together with computed flow lines. Here we used 35 clusters, and the run-time for determining the pole and testing points was 9 minutes. The resulting linear system was of size 14581×2150. Although fairly large, this system has, on average, 7.2 degrees

27
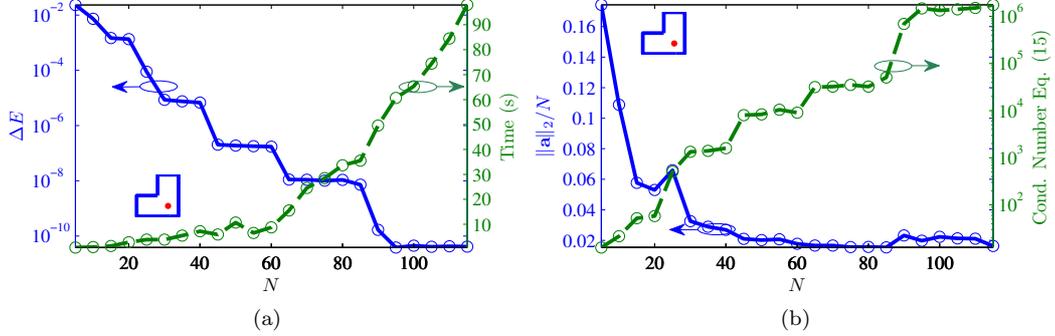
Figure 13: L-shaped domain, monopole excitation. Despite singularities on the boundary, errors smaller than 10^{-10} can be obtained with no more than 100 poles (a). Also, although the condition number increases with N, the norm of the vector of residues (normalized with respect to N) decreases with N (b).
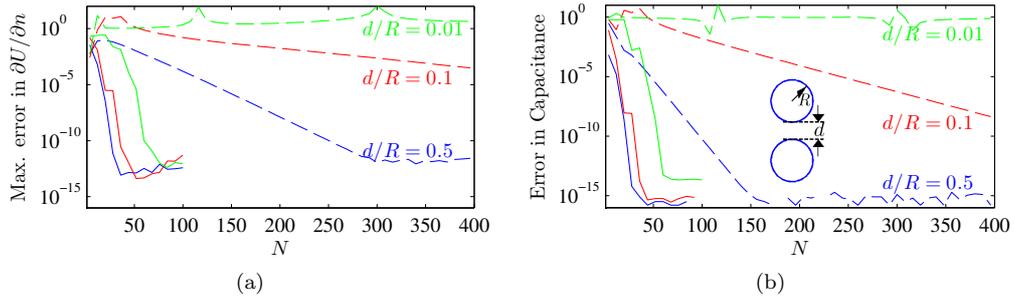
Figure 14: Two closely-spaced curves. Full lines, N-pole solution. Dashed lines, $2N$ Nyström solution. The accuracy of the proposed method is far less sensitive to the separation between circles than the Nyström method.

of freedom per curve, and it was solved in 2 minutes by QR decomposition. The error tolerance for each cluster was again $\Delta E < 0.01$ and the error of the full solution was $\Delta E = 0.014$.

8. Summary

In this paper, we describe an algorithm for solving 2D Laplace BVPs with a desingularized method. The algorithm is based on rational-function fitting techniques, and it uses rational Gauss-Chebyshev quadrature rules to locate testing points at which the boundary conditions are enforced in a least-squares sense. To use these rules, the boundary curves are parameterized by piecewise-rational functions, which can be used to approximate quite arbitrary boundary curves. The algorithm can be viewed as an adaptive desingularized method, in which the sources are not restricted to curves conformal with the boundary curves, but can be placed anywhere outside the BVP domain. This
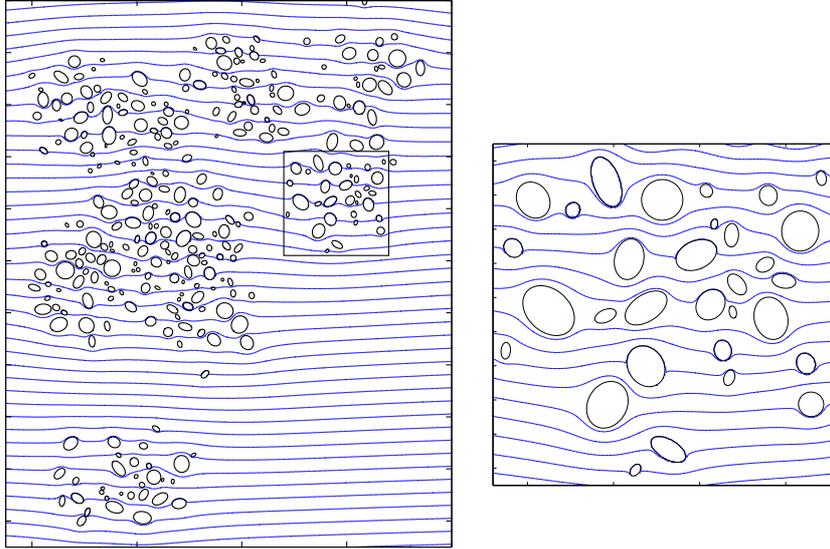
Figure 15: Stream lines around 300 ellipses. Calculation using the partitioning scheme took 11 minutes on a single 2Ghz PC with a four-core processor.

added flexibility allows the method to adapt to difficult boundary geometries and almost-singular boundary data, yielding solutions with errors approaching the machine epsilon for a relatively small number of degrees of freedom. Examples shown include boundaries with sharp and almost-sharp corners, boundary data that is derived from functions with essential singularities, and boundary curves that nearly touch. We also described a partitioning scheme, in which pole optimization is limited to curves within a cluster, that can be used to tackle large-scale problems.

As the proposed method relies on a correspondence between analytic functions in the complex plane and harmonic functions, and on properties of rational functions, it may be difficult to generalize to 3D Laplace problems. On the other hand, generalizing the method for 2D Helmholtz or Biharmonic BVPs appears more straightforward. The general solution of the Biharmonic equation can be expressed in terms of two functions that are analytic throughout the BVP domain, and rational functions could be used for their approximation. For Helmholz BVPs, it may be possible to exploit a connection between the singularities of the analytic continuations of solution to Helmholtz and Laplace BVPs with the same boundary data. For exterior problems, it was shown in [66] that the convex hull of singularities of Helmholtz and Laplace BVPs are the same, assuming the same boundary data. Therefore, the location of poles in a Laplace problem can serve as a guide for the location of sources in a Helmholtz problem (see [67] for an example of this idea). In particular, locating the Helmholtz sources so that they enclose the convex hull of singularities of the Laplace problem guarantees that all of the singularities of the Helmholtz problem are enclosed. According to the conjectures in [25], which have recently been proved [30], this guarantees exponential convergence.

9. Acknowledgements

Appendix A. Iterated rational fitting

The IRF method we use is based on a variant of [46] presented in [49]. As explained in Section 2.2, we seek a rational function P/Q that approximates $\widehat{W}$, and we do so by solving weighted linear least-squares problems of the from (21). Using the quadrature rules of Section 3.2, we discretize the continuous norm in (21) and write the minimization problem as,

$$\min_{\mathbf{a},\mathbf{b}} \sum_{k=1}^{K} \sum_{n=0}^{N} \frac{\sqrt{\lambda_k}}{\widehat{Q}(z_k)} \left| b_n z_k^n - a_n z_k^n \widehat{W}(z_k) \right| \tag{A.1}$$

in which the unknowns are the monomial coefficients of $P(z)$ and $Q(z)$, denoted $\mathbf{b}$ and $\mathbf{a}$, respectively. To avoid the trivial solution, a normalization condition, to be specified below, must be imposed on one of the polynomials.

Let us rewrite (A.1) in matrix form, using the notation

$$\begin{aligned}
\mathbf{\Lambda} &= \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_K), \\
\widehat{\mathbf{Q}} &= \mathrm{diag}\left[\widehat{Q}(z_1), \widehat{Q}(z_2), \ldots, \widehat{Q}(z_K) \right], \\
\widehat{\mathbf{W}} &= \mathrm{diag}\left[\widehat{W}(z_1), \widehat{W}(z_2), \ldots, \widehat{W}(z_K) \right], \\
V_{kn} &= z_k^n, \quad n = 0, 1, \ldots, N, \\
\mathbf{c} &= \begin{bmatrix} \mathbf{b} \\ \mathbf{a} \end{bmatrix}.
\end{aligned} \tag{A.2}$$

We obtain,

$$\min_{\mathbf{c}} \left\| \mathbf{\Lambda}^{\frac{1}{2}} \widehat{\mathbf{Q}}^{-1} \left[\mathbf{V}, -\widehat{\mathbf{W}}\mathbf{V} \right] \mathbf{c} \right\|. \tag{A.3}$$

The conditioning of the Vandermonde matrix, $\mathbf{V}$, can be very poor (sampling points on the real line, for example), or very good (sampling points on the unit circle, for example), and this can affect the conditioning of the minimization problem (A.3). To deal with this problem, we construct two orthonormal bases, one for the column space of $\mathbf{A} = \mathbf{\Lambda}^{\frac{1}{2}} \widehat{\mathbf{Q}}^{-1} \mathbf{V}$ and one for the column space of $\mathbf{B} = \mathbf{\Lambda}^{\frac{1}{2}} \widehat{\mathbf{Q}}^{-1} \widehat{\mathbf{W}} \mathbf{V}$. This can be done with the Arnoldi iteration [68, 69], by noting that the columns of $\mathbf{A}$ form a Krylov subspace,

$$K = \mathrm{span}[\mathbf{d}, \mathbf{Z}\mathbf{d}, \mathbf{Z}^2\mathbf{d}, \ldots, \mathbf{Z}^N\mathbf{d}], \tag{A.4}$$

where $\mathbf{Z} = \mathrm{diag}\,(z_1, z_2, \ldots, z_K)$ and $\mathbf{d} = \mathrm{diag}(\mathbf{\Lambda}^{\frac{1}{2}} \widehat{\mathbf{Q}}^{-1})$, and similarly for the columns of $\mathbf{B}$. Once we have constructed the orthonormal bases,

$$\mathrm{colsp}(\mathbf{P}) = \mathrm{colsp}(\mathbf{A}), \quad \mathbf{P}^{\dagger}\mathbf{P} = \mathbf{I} \tag{A.5}$$

$$\mathrm{colsp}(\mathbf{Q}) = \mathrm{colsp}(\mathbf{B}), \quad \mathbf{Q}^{\dagger}\mathbf{Q} = \mathbf{I}, \tag{A.6}$$

we rewrite (A.3) in the new basis,

$$\min_{\tilde{\mathbf{c}}} \left|\left|[\mathbf{P}, -\mathbf{Q}]\, \tilde{\mathbf{c}}\right|\right|, \tag{A.7}$$

and impose the normalization condition $\tilde{c}_{2N} = 1$, which corresponds to $Q(z)$ being monic in the new basis. The solution of the minimization problem is then obtained by solving the over-determined linear system

$$[\mathbf{P}, -\mathbf{Q}_{1:N-1}]\tilde{\mathbf{c}} = \mathbf{Q}_N \tag{A.8}$$

in the least-squares sense. In (A.8), we denote the column corresponding to the Nth-degree polynomial basis function by $\mathbf{Q}_N$, while the rest of the $\mathbf{Q}$ matrix is denoted by $\mathbf{Q}_{1:N-1}$. The numerical stability of (A.7) is generally much better than that of (A.3), as the unknowns are coefficients of two sets of orthonormal columns.

Once (A.3) is solved, the zeros of $Q(z)$, which are the poles for the next iteration, can be obtained without transforming back to the monomial basis. They are the eigenvalues of a generalized companion matrix [46],

$$\mathbf{C} = \mathbf{H}_{1:N,1:N} - H_{N+1,N}\, \tilde{\mathbf{c}}_{N+1:2N}\, \mathbf{e}_N^T, \tag{A.9}$$

where $\mathbf{e}_N$ is the Nth column of the order-N identity matrix and $\mathbf{H}$ is the upper Hessenberg matrix obtained from the Arnoldi iteration used to generate the orthonormal basis for $\mathrm{colsp}(\mathbf{B})$.

Appendix B. Determining whether a pole is inside D

When a pole is close to the boundary, or the boundary shape is complicated, it may be nontrivial to find out whether a pole is inside D or not. By restricting ourselves to boundaries described by piecewise rational functions, a solution to this problem is readily obtained. Using Cauchy's theorem, a pole z' is inside D if and only if

$$\int_C \frac{1}{z - z'} dz = \int_0^1 \frac{1}{z(s) - z'} \frac{dz(s)}{ds}\, ds = 2\pi i \tag{B.1}$$

This integral cannot be calculated exactly using the rational Gauss-Chebyshev rules because the Chebyshev weight function is absent. However, it can be calculated analytically by breaking it up into rational-function sections, writing the integrands as rational functions, and using a partial-fraction expansion.

References

[1] J. Helsing, R. Ojala, On the evaluation of layer potentials close to their sources, J. Comp. Phys. (5) (2008) 2899–2921.

[2] J. Helsing, R. Ojala, Corner singularities for elliptic problems: integral equations, graded meshes, quadrature, and compressed inverse preconditioning, J. Comp. Phys. (20) (2008) 8820–8840.

[3] T. A. Driscoll, L. N. Trefethen, Schwarz-Christoffel mapping, Cambridge University Press Cambridge, 2002.

[4] D. Marshall, S. Rohde, Convergence of a variant of the Zipper algorithm for conformal mapping, SIAM J. Numer. Anal (6) (2007) 2577–2609.

[5] J. Bremer, V. Rokhlin, Efficient discretization of Laplace boundary integral equations on polygonal domains, J. Comp. Phys. (7) (2010) 2507–2525.

[6] J. Bremer, V. Rokhlin, I. Sammis, Universal quadratures for boundary integral equations on two-dimensional domains with corners, J. Comp. Phys. (22) (2010) 8259–8280.

[7] I. N. Vekua, New Methods for Solving Elliptic Equations, Amsterdam, 1967.

[8] V. D. Kupradze, On the approximate solution of problems in mathematical physics, Russ. Math. Surv. (1967) 58.

[9] L. Fox, P. Henrici, C. Moler, Approximations and bounds for eigenvalues of elliptic operators, SIAM J. Num. Anal. (1) (1967) 89–102.

[10] S. Bergman, J. G. Herriot, Numerical solution of boundary-value problems by the method of integral operators, Numerische Mathematik (1) (1965) 42–65.

[11] S. Eisenstat, On the rate of convergence of the Bergman-Vekua method for the numerical solution of elliptic boundary value problems, SIAM J. Num. Anal. (3) (1974) 654–680.

[12] G. Fairweather, A. Karageorghis, The method of fundamental solutions for elliptic boundary value problems, Adv. Comp. Math. (1) (1998) 69–95.

[13] M. A. Golberg, C. S. Chen, The method of fundamental solutions for potential, Helmholtz and diffusion problems, Bound. Int. Meth. (1999) 103.

[14] D. I. Kaklamani, H. T. Anastassiu, Aspects of the method of auxiliary sources (MAS) in computational electromagnetics, IEEE Antenn. Propag. Mag. (2002) 48–64.

[15] O. M. Bucci, G. D'Elia, M. Santojanni, Non-redundant implementation of method of auxiliary sources for smooth 2D geometries, Electron. Lett. (22) (2005) 1203–1205.

[16] C. Hafner, The Generalized Multipole Technique for Computational Electromagnetics, Artech House, Norwood, MA, 1990.

[17] G. Tayeb, S. Enoch, Combined fictitious-sources-scattering-matrix method, J. Opt. Soc. Am. A (8) (2004) 1417–1423.

[18] D. Maystre, M. Saillard, G. Tayeb, Special methods of wave diffraction, Scattering, Academic Press, London, 2001, Ch. 1.5.6.

[19] Y. Leviatan, Am. Boag, Al. Boag, Generalized formulations for electromagnetic scattering from perfectly conducting and homogeneous material bodies-theory and numerical solution, IEEE Trans. Antenn. Propag. (1988) 1722–1734.

[20] Am. Boag, Y. Leviatan, Al. Boag, Analysis of electromagnetic scattering from linear periodic arrays of perfectly conducting bodies using a cylindrical-current model, IEEE Trans. Antenn. Propag. (1991) 1332–1337.

[21] A. Hochman, Y. Leviatan, Efficient and spurious-free integral-equation-based optical waveguide mode solver, Opt. Express (2007) 14431–14453.

[22] K. E. Atkinson, The numerical solution of integral equations of the second kind, Cambridge University Press, 1997.

[23] Y. Leviatan, Analytic continuation considerations when using generalized formulations for scattering problems, IEEE Trans. Antenn. Propag. (8) (1990) 1259–1263.

[24] M. Katsurada, Charge simulation method using exterior mapping functions, Japan J. Indus. Appl. Math. (1) (1994) 47–61.

[25] A. H. Barnett, T. Betcke, Stability and convergence of the method of fundamental solutions for Helmholtz problems on analytic domains, J. Comp. Phys. (14) (2008) 7003–7026.

[26] F. Chiba, T. Ushijima, Exponential decay of errors of a fundamental solution method applied to a reduced wave problem in the exterior region of a disc, J. Comp. Appl. Math. (2) (2009) 869 – 885.

[27] C. J. S. Alves, On the choice of source points in the method of fundamental solutions, Eng. Anal. Bound. Elemen. (12) (2009) 1348–1361.

[28] L. Greengard, V. Rokhlin, A fast algorithm for particle simulations, J. Comp. Phys. (2) (1987) 325–348.

[29] Y. J. Liu, N. Nishimura, Z. H. Yao, A fast multipole accelerated method of fundamental solutions for potential problems, Eng. Anal. Bound. Elem. (11) (2005) 1016–1024.

[30] U. Kangro, Convergence of collocation method with delta functions for integral equations of first kind, Int. Equ. Op. Theory (2) (2010) 265–282.

[31] R. Mathon, R. L. Johnston, The approximate solution of elliptic boundary-value problems by fundamental solutions, SIAM J. Num. Anal. (4) (1977) 638–650.

[32] M. Nishimura, S. Takamatsu, H. Shigesawa, A numerical analysis of electromagnetic scattering of perfect conducting cylinders by means of discrete singularity method improved by optimization process, Electron. Comm. Japan (10) (1984) 75–81.

[33] A. P. Cisilino, B. Sensale, Application of a simulated annealing algorithm in the optimal placement of the source points in the method of the fundamental solutions, Comput. Mech. (2) (2002) 129–136.

[34] I. I. Heretakis, P. J. Papakanellos, C. N. Capsalis, Analysis of electromagnetic scattering by infinite conducting cylinders of arbitrary smooth cross section using a genetically optimised technique (GA/MAS), J. Electromag. Wave. Appl. (11) (2002) 1555–1572.

[35] S. Eisler, Y. Leviatan, Analysis of electromagnetic scattering from metallic and penetrable cylinders with edges using a multifilament current model, IEE-Proc. H (1989) 431–438.

[36] E. Moreno, D. Erni, C. Hafner, R. Vahldieck, Multiple multipole method with automatic multipole setting applied to the simulation of surface plasmons in metallic nanostructures, J. Opt. Soc. Am. A (2002) 101–111.

[37] S. G. Mikhlin, Integral equations and their applications to certain problems in mechanics, mathematical physics and technology, Pergamon press New York, 1957.

[38] A. Greenbaum, L. Greengard, G. B. McFadden, Laplace's equation and the Dirichlet-Neumann map in multiply connected domains, J. Comp. Phys. (2) (1993) 267–278.

[39] A. H. Barnett, T. Betcke, An exponentially convergent nonpolynomial finite element method for time-harmonic scattering from polygons, SIAM J. Sci. Comp. (3) (2010) 1417–1441.

[40] J. Wolff, Sur les séries $\sum a_k/(z - z_k)$, Comptes rendus Acad. Sci. (1921) 1327–1328.

[41] J. L. Walsh, Interpolation and approximation by rational functions in the complex domain, American Mathematical Society, 1965.

[42] H. Stahl, The convergence of padé approximants to functions with branch points, J. Appr. Theory (2) (1997) 139–204.

[43] J. Borcea, R. Bøgvad, B. Shapiro, On rational approximation of algebraic functions, Adv. Math. (2) (2006) 448–480.

[44] B. Gustavsen, A. Semlyen, Rational approximation of frequency domain responses by vector fitting, IEEE Trans. Power Deliv. (3) (1999) 1052–1061.

[45] D. Deschrijver, B. Haegeman, T. Dhaene, Orthonormal vector fitting: A robust macromodeling tool for rational approximation of frequency domain responses, IEEE Trans. Adv. Packag. (2) (2007) 216–225.

[46] C. Coelho, J. Phillips, L. Silveira, Robust rational function approximation algorithm for model generation, in: Proc. 36th ACM/IEEE Des. Auto. Conf., ACM, 1999, p. 212.

[47] E. Levy, Complex curve fitting, IRE Trans. Auto. Cont. (1) (1959) 37–43.

[48] C. Sanathanan, J. Koerner, Transfer function synthesis as a ratio of two complex polynomials, IEEE Trans. Auto. Cont. (1) (1963) 56–58.

[49] D. Vasilyev, M. Tsuk, J. White, An Equivalence Theorem for Iterated Rational-and Vector-fitting, Prog. Electromag. Res. (2010) 639.

[50] P. Gonnet, R. Pachón, L. N. Trefethen, Robust rational interpolation and least-squares, Elect. Trans. Num. Anal. (2011) 146–167.

[51] L. N. Trefethen, et al., Chebfun Version 4.0, The Chebfun Development Team, http://www.maths.ox.ac.uk/chebfun/ (2011).

[52] J. Van Deun, K. Deckers, A. Bultheel, J. A. C. Weideman, Algorithm 882: Near-best fixed pole rational interpolation with applications in spectral methods, ACM Trans. Math. Soft. (2) (2008) 1–21.

[53] E. W. Cheney, Introduction to approximation theory, American Mathematical Society, 1982.

[54] L. N. Trefethen, Approximation Theory and Approximation Practice, SIAM, 2012.
URL http://www2.maths.ox.ac.uk/chebfun/ATAP/

[55] M. Froissart, Approximation de padé: application à la physique des particules élémentaires, CNRS, RCP, Programme No. 29.

[56] J. Gilewicz, M. Pindor, Padé approximants and noise: rational functions, J. Comp. Appl. Math. (1) (1999) 285–297.

[57] R. Pachón, R. B. Platte, L. N. Trefethen, Piecewise-smooth chebfuns, IMA J. Num. Anal.

[58] W. Pogorzelski, Integral equations and their applications, Pergamon, 1966.

[59] B. Stupfel, A fast-domain decomposition method for the solution of electromagnetic scattering by large objects, IEEE Trans. Antenn. Propag. (10) (1996) 1375–1385.

[60] S. C. Lee, M. N. Vouvakis, J. F. Lee, A non-overlapping domain decomposition method with non-matching grids for modeling large finite antenna arrays, J. Comp. Phys. (1) (2005) 1–21.

[61] C. Geuzaine, A. Vion, R. Gaignaire, P. Dular, R. V. Sabariego, An amplitude finite element formulation for multiple-scattering by a collection of convex obstacles, IEEE Trans. Magnet. (8) (2010) 2963–2966.

[62] S. Lloyd, Least squares quantization in PCM, IEEE Trans. Inf. Theory (2) (1982) 129–137.

[63] http://www.mathworks.com/matlabcentral/fileexchange/?term=laprat.

[64] C. J. S. Alves, V. Leitão, Crack analysis using an enriched MFS domain decomposition technique, Eng Anal. Bound. Elemen. (3) (2006) 160–166.

[65] W. R. Smythe, Static and dynamic electricity, McGraw-Hill, 1950.

[66] R. Millar, The analytic continuation of solutions to elliptic boundary value problems in two independent variables, J. Math. Anal. Appl. (2) (1980) 498–515.

[67] Y. Leviatan, E. Erez, Analysis of electromagnetic scattering by a nearly touching pair of cylinders with the use of a multifilament source model, Microwave Opt. Technol. Lett. (3) (1996) 152–155.

[68] W. E. Arnoldi, The principle of minimized iterations in the solution of the matrix eigenvalue problem, Q. Appl. Math (17) (1951) 17–29.

[69] Y. Saad, Iterative methods for sparse linear systems, SIAM, 2003.