A simplified technique for the efficient and high-order accurate discretization of boundary integral equations in 2D on domains with corners

A. Gillman[1], *S. Hao*[2], *P.G. Martinsson*[2]

Background: This note comments on some recently developed techniques for computing an approximate solution to a Boundary Integral Equation (BIE) like

$$\alpha\, q(\boldsymbol{x}) + \int_\Gamma K(\boldsymbol{x},\boldsymbol{y})\, q(\boldsymbol{y})\, ds(\boldsymbol{y}) = f(\boldsymbol{x}), \qquad \boldsymbol{x} \in \Gamma, \tag{1}$$

where Γ is a piecewise smooth contour in the plane, and where K is a one of the standard kernels of potential theory such as, e.g., the single or double layer kernels associated with the Laplace or Helmholtz equations. A challenge in solving (1) is that its integrand exhibits complicated singular behavior near the corner points of Γ. A classical technique for dealing with this difficulty has been to expand the unknown q near the corner using specialized basis functions that incorporate analytical knowledge about the singularity [2]. Recently, however, a remarkable observation has been made [1, 7, 8] that there exist general purpose techniques that do not require any analytical à priori knowledge other than that the integrand of (1) be absolutely integrable.

In a nutshell, the idea of [1, 7, 8] is to use a standard Nyström discretization of (1) designed for a smooth contour. The discretization should use a panel based (i.e. non-global) quadrature rule such as, e.g., a composite Gaussian rule. Then simply refine the computational mesh near any corner. For any given computational tolerance ε (say $\varepsilon = 10^{-10}$), continue the refinement until the contribution from any panels directly touching a corner is bounded by ε (this is possible since the integrand in (1) is absolutely integrable). Now omit the panels nearest to the corner from the discretization. Observe that on any remaining panel, the function q is smooth enough to be accurately represented by the interpolant implied by the chosen quadrature rule.

The apparent drawback of a simplistic refinement process like the one described is that it can dramatically increase the number of degrees of freedom required in the Nyström discretization. A key insight of [1, 7, 8] is that the "superfluous" degrees of freedom added by the refinement can be eliminated from the linear system via a *strictly local* process. Moreover, this local process can be executed in time that scales linearly with the number of degrees of freedom added. The end result is a linear system discretizing (1) that has about as many degrees of freedom as one would have needed had the corner not been present in the first place. (For the case of regular polygonal domains, the compression can even be performed in sublinear time [8].)

The task of "squeezing out" the degrees of freedom added by the local refinement near the corner is in [1, 7, 8] executed via purpose-built local compression techniques that can be somewhat challenging to implement. The purpose of this note is to demonstrate that this compression step can be executed via the general purpose direct solvers described in [4, 5, 9, 11].

A linear algebraic observation: The compression technique that allows us to eliminate the superfluous degrees of freedom is based on the observation that certain off-diagonal blocks of the coefficient matrix resulting from the discretization of (1) have low numerical rank. Critically, the important ranks do not depend on how many degrees of freedom are used in the refinement near the corner. To illustrate how such rank-deficiencies can be exploited, consider in general the task of solving the linear system

$$\begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{q}_1 \\ \mathbf{q}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \end{bmatrix}, \tag{2}$$

where $\mathbf{A}_{11}$ is of size $n_1 \times n_1$ and $\mathbf{A}_{22}$ is of size $n_2 \times n_2$. Now assume that $\mathbf{A}_{12}$ and $\mathbf{A}_{21}$ each are of rank k. Think of n_1 as a large number (e.g. the number of degrees of freedom used in the refinement of the corner, say $n_1 \sim 10^3$), and k as a small number (in practice, in the range $10 - 50$). Then $\mathbf{A}_{12}$ and $\mathbf{A}_{21}$ admit factorizations

$$\underset{n_1\times n_2}{\mathbf{A}_{12}} = \underset{n_1\times k}{\mathbf{U}_1}\ \underset{k\times n_2}{\mathbf{B}_{12}} \qquad \text{and} \qquad \underset{n_2\times n_1}{\mathbf{A}_{21}} = \underset{n_2\times k}{\mathbf{B}_{21}}\ \underset{k\times n_1}{\mathbf{V}_1^*}, \tag{3}$$

where $\mathbf{U}_1$ and $\mathbf{V}_1$ are well-conditioned matrices. We further assume that the data vector $\mathbf{f}_1$ belongs to the same k-dimensional space as the columns of $\mathbf{A}_{12}$ (if it does not, then the space can be extended as needed),

$$\mathbf{f}_1 = \mathbf{U}_1\, \tilde{\mathbf{f}}_1. \tag{4}$$

[1]Department of Mathematics, Dartmouth College, Hanover, NH, 03755-3551
[2]Department of Applied Mathematics, University of Colorado, Boulder, CO, 80309-0526

When (3) and (4) hold, the linear system (2) with $n_1 + n_2$ unknowns is in a certain sense equivalent to the smaller system

$$\begin{bmatrix} \mathbf{D}_{11} & \mathbf{B}_{12} \\ \mathbf{B}_{21} & \mathbf{A}_{22} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{q}}_1 \\ \mathbf{q}_2 \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{f}}_1 \\ \mathbf{f}_2 \end{bmatrix} \tag{5}$$

with only $k + n_2$ unknowns. In (5), $\mathbf{D}_{11}$ and $\tilde{\mathbf{q}}_1$ are defined by

$$\mathbf{D}_{11} = \left(\mathbf{V}_1^* \mathbf{A}_{11}^{-1} \mathbf{U}_1\right)^{-1} \qquad \text{and} \qquad \tilde{\mathbf{q}}_1 = \mathbf{V}_1^* \mathbf{q}_1. \tag{6}$$

When we say that (2) and (5) are "equivalent" we mean that the solution $\{\mathbf{q}_1, \mathbf{q}_2\}$ of the larger system (2) can be obtained from the solution $\{\tilde{\mathbf{q}}_1, \mathbf{q}_2\}$ of the smaller system (5) via the formula

$$\mathbf{q}_1 = \mathbf{A}_{11}^{-1}\, \mathbf{U}_1\, \mathbf{D}_{11}\, \tilde{\mathbf{q}}_1. \tag{7}$$

To be precise, the equivalence holds when $\mathbf{A}_{11}$ and $\mathbf{V}_1^* \mathbf{A}_{11}^{-1} \mathbf{U}_1$ are both non-singular.

Matrix skeletons: For the low-rank factorizations (3), it is convenient to use a so-called *interpolative decomposition* (ID) [3] in which $\mathbf{B}_{12}$ is a $k \times n_2$ matrix consisting of k rows of $\mathbf{A}_{12}$ and $\mathbf{B}_{21}$ is an $n_2 \times k$ matrix consisting of k columns of $\mathbf{A}_{21}$. The matrices $\mathbf{U}_1$ and $\mathbf{V}_1$ each hold a $k \times k$ identity matrix as a submatrix, and have no entries whose magnitude exceeds 1.

The advantage of using an ID is that the matrices $\mathbf{A}_{12}$ and $\mathbf{A}_{21}$ *need never be formed*. Instead, a local computation determines the index vectors pointing out which columns and rows are needed. and then only those entries need to be computed to create the off-diagonal blocks $\mathbf{B}_{12}$ and $\mathbf{B}_{21}$ in (5). Moreover, when skeletonization is used, the vector $\tilde{\mathbf{f}}_1$ can be formed by evaluating the vector $\mathbf{f}_1$ only at the k nodes associated with the spanning rows of $\mathbf{A}_{12}$.

A strictly local technique for computing $\mathbf{U}_1$ and $\mathbf{V}_1$, and determining the corresponding index vectors, is described in Section 6.2 of [4].

Outline of the solution process: To describe the refinement and the local compression, we consider a contour with a single corner, like the one shown in Figure 1(a). We partition the contour into two disjoint parts, $\Gamma = \Gamma_1 \cup \Gamma_2$, in such a way that Γ_1 is a small piece containing the corner. The piece Γ_2 is smooth, and can be discretized into panels rather coarsely (since we use a high order rule, high accuracy does not require many points). For the piece Γ_1, we use a simplistic refinement strategy where we recursively cut the panel nearest the corner in half. Once the innermost panel is small enough that its contribution can be ignored (recall that we assume that the integrand is integrable), it is simply discarded and the refinement stops. The Nyström discretization now results in a linear system like (2). Observe that the block $\mathbf{A}_{11}$ can be very large since we may need thousands of points to fully resolve the singularity near the corner. The key observation is now that the rank k of $\mathbf{A}_{12}$ and $\mathbf{A}_{21}$ is essentially independent of how finely the corner has been refined. This allows us to employ the direct solver of [4, 5, 9, 11] to compress the corner and eliminate the "extra" degrees of freedom used to resolve the singularity. The output of the compression is (i) a set of k collocation points inside Γ_1 that are automatically picked by the algorithm from among the n_1 points used in the refinement and (ii) a $k \times k$ dense matrix $\mathbf{D}_{11}$ that represents self-interaction among the k remaining points. The cost of this compression is $O(n_1\, k^2)$. Once the compression has been completed, all that remains is to solve the smaller system (5) to obtain the solution $\{\tilde{\mathbf{q}}_1, \mathbf{q}_2\}$, and, if required, reconstructing the full vector $\mathbf{q}_1$ via (7).

For a contour with multiple corners, simply repeat the local compression for each corner. Note that the compression processes are independent, meaning that they can be executed in parallel on a multi-core machine. The authors of [1, 7, 8] have demonstrated the astonishing power of this observation by solving numerical examples involving tens of thousands of corners to close to full double precision accuracy.

Note: *To achieve optimal accuracy, it is important to scale the matrix elements in the Nyström discretization as described in* [2]*. The idea is to scale the vectors in the discretization by the quadrature weights so that for a panel Γ_p corresponding to an index vector I_p, we have $||q||_{L^2(\Gamma_p)} \approx ||\mathbf{q}(I_p)||_{\ell^2}$. The matrix elements in the coefficient matrix are scaled analogously so that $||\alpha\, q(\cdot) + \int_\Gamma k(\cdot, \boldsymbol{y})\, q(\boldsymbol{y})\, ds(\boldsymbol{y})||_{L^2(\Gamma_p)} \approx ||[\mathbf{Aq}](I_p)||_{\ell^2}$. The idea is to not give disproportionate weight to a region of the contour that ends up with a high density of discretization points due to a local refinement.*

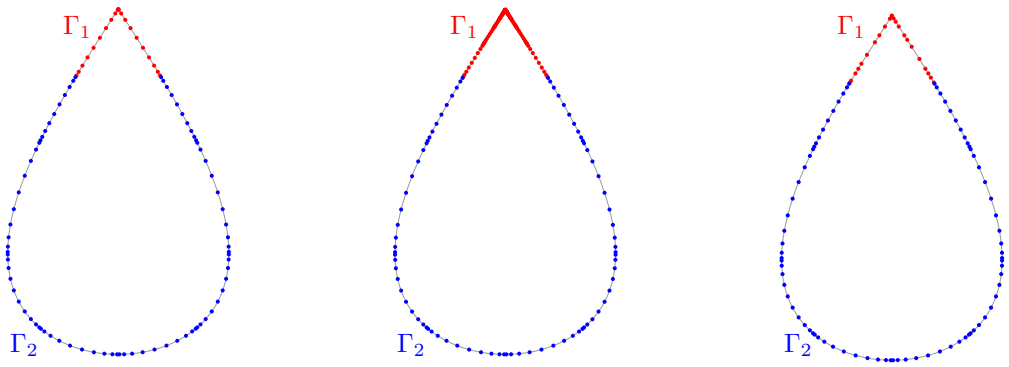

FIGURE 1. The boundary Γ considered in the numerical experiments. (a) The original Gaussian grid before refinement, red nodes in Γ_1 and blue in Γ_2. (b) The locally refined grid. (c) The grid after local compression.

Numerical illustration: Consider the boundary value problem

$$\begin{aligned} \Delta u(\boldsymbol{x}) + \kappa^2 u(\boldsymbol{x}) &= 0 && \boldsymbol{x} \in \bar{\Omega}^{\mathrm{c}} \\ u(\boldsymbol{x}) &= g(\boldsymbol{x}) && \boldsymbol{x} \in \Gamma = \partial\Omega \\ \sqrt{|\boldsymbol{x}|}\left(\tfrac{\partial}{\partial|x|} - \mathrm{i}\kappa\right) u(\boldsymbol{x}) &\to 0 && \text{as } |\boldsymbol{x}| \to \infty \end{aligned} \tag{8}$$

where κ is a constant wavenumber, Ω is a two dimensional bounded domain whose boundary Γ is a piecewise smooth contour, and $\bar{\Omega}^{\mathrm{c}} = \mathbb{R}^2/\bar{\Omega}$. We make the ansatz that the solution u can be represented by

$$u(\boldsymbol{x}) = \int_\Gamma \left(\frac{\mathrm{i}}{4}\frac{\partial}{\partial \boldsymbol{\nu}(\boldsymbol{y})} H_0(\kappa|\boldsymbol{x}-\boldsymbol{y}|)\right) q(\boldsymbol{y})\, ds(\boldsymbol{y}), \qquad \boldsymbol{x} \in \bar{\Omega}^{\mathrm{c}},$$

where $\boldsymbol{\nu}(\boldsymbol{y})$ represents the normal vector to Γ at the point $\boldsymbol{y} \in \Gamma$ and $\frac{\mathrm{i}}{4}H_0(\kappa|\boldsymbol{x}-\boldsymbol{y}|)$ is the fundamental solution to the Helmholtz equation. By enforcing the boundary condition, we obtain the following equation for the unknown boundary charge distribution q:

$$-\frac{1}{2}q(\boldsymbol{x}) + \int_\Gamma \left(\frac{\mathrm{i}}{4}\frac{\partial}{\partial \boldsymbol{\nu}(\boldsymbol{y})} H_0(\kappa|\boldsymbol{x}-\boldsymbol{y}|)\right) q(\boldsymbol{y})\, ds(\boldsymbol{y}) = g(\boldsymbol{x}), \quad \text{for } \boldsymbol{x} \in \Gamma. \tag{9}$$

We discretize (9) via a Nyström technique based on a composite Gaussian quadrature [10] with 10 Gaussian nodes per panel. Note that the kernel of (9) is weakly singular, but high order accuracy can be retained by modifying a small number of matrix entries close to the diagonal, see [6].

All experiments are run on a Lenovo laptop computer with 8GB of RAM and a 2.6GHz Intel i5-2540M processor. The compression technique was implemented rather crudely in MATLAB, which means that significant further gains in speed should be achievable.

A tear shaped geometry with arc-length 4π is considered, cf. Figure 1 (a), for three values of κ (corresponding to domain sizes 0.31, 3.18, and 31.83 wavelengths, respectively). The boundary data is taken to be the incident wave $e^{\mathrm{i}\kappa x_2}$. We measured the error E_{charge} in the computed boundary charge q given by

$$E_{\text{charge}} = \frac{\|q - q_{\text{exact}}\|_{L^2(\Gamma)}}{\|q_{\text{exact}}\|_{L^2(\Gamma)}}$$

and the error E_{pot} in the potential u, evaluated on the boundary of a circle S with radius 3 enclosing Ω,

$$E_{\text{pot}} = \frac{\|u - u_{\text{exact}}\|_{L^2(S)}}{\|u_{\text{exact}}\|_{L^2(S)}}.$$

Since the exact solutions q_{exact} and u_{exact} were not available, we measured against a very highly over-resolved reference solution. Rows 1, 5, and 9 of Table 1 report the errors E_{charge} and E_{pot} when there is no refinement of the corner. In all other experiments, the corner is discretized with 1280 points and is compressed for three prescribed tolerances ϵ. Table 1 reports the size of the original system ($N \times N$), the size of the compressed system ($N_{\text{compressed}} \times N_{\text{compressed}}$), the number of skeleton nodes in the corner k, and the time T_{compress} in

κ	N	ϵ	$N_{\text{compressed}}$	k	T_{compress}	E_{charge}	E_{pot}
1	100	—	—	—	—	$2.3e-01$	$1.3e-04$
1	1380	$1e-7$	125	25	0.66	$8.9e-07$	$1.3e-07$
1	1380	$1e-10$	137	37	0.85	$8.5e-10$	$9.1e-11$
1	1380	$1e-12$	145	45	0.96	$6.6e-12$	$1.1e-12$
10	400	—	—	—	—	$4.4e-01$	$1.4e-04$
10	1680	$1e-7$	427	27	0.64	$1.8e-05$	$5.7e-07$
10	1680	$1e-10$	438	38	0.86	$9.2e-07$	$1.4e-09$
10	1680	$1e-12$	445	45	1.00	$1.5e-11$	$6.4e-13$
100	1600	—	—	—	—	$1.0e-01$	$1.4e-04$
100	2880	$1e-7$	1589	29	0.65	$8.1e-07$	$1.5e-07$
100	2880	$1e-10$	1599	39	0.85	$7.2e-10$	$1.3e-10$
100	2880	$1e-12$	1607	48	1.00	$9.3e-12$	$6.1e-12$

TABLE 1. Results from solving the external Helmholtz problem (8) on the geometry shown in Figure 1(a) for three different values of the wave-number κ. The errors E_{charge} and E_{pot} report the relative errors in the computed charge distribution q, and the evaluated potential, respectively. k is the rank of interaction (to precision ϵ) between the corner piece in Γ_1 and the rest of the contour, and N_{compress} is the size of the compressed system. T_{compress} is the time in seconds required for compressing the corner.

seconds for the compression. We observe that very high accuracy is attained for every wave-number. Moreover, the rank k of interaction between the corner patch (Γ_1) and the rest of the domain (Γ_2) is always small, it depends only weakly on the requested accuracy, and it hardly depends at all on the wave-number.

The technique described can easily be adapted to complicated domains with many corners, in a manner entirely analogous to [1, 7, 8].

As a curiosity, let us finally mention that we also ran numerical experiments on domains whose corners were not re-entrant to Ω^c, but found that for such domains, nine digits of accuracy was obtained *without* local refinement in the corner.

REFERENCES

[1] J. Bremer, *A fast direct solver for the integral equations of scattering theory on planar curves with corners*, Journal of Computational Physics **231** (2012), 1879–1899.

[2] J. Bremer, Rokhlin V., and I. Sammis, *Universal quadratures for boundary integral equations on two-dimensional domains with corners*, Journal of Computational Physics **229** (2010), no. 22, 8259 – 8280.

[3] H. Cheng, Z. Gimbutas, P.G. Martinsson, and V. Rokhlin, *On the compression of low rank matrices*, SIAM Journal of Scientific Computing **26** (2005), no. 4, 1389–1404.

[4] A. Gillman, P. Young, and P.G Martinsson, *A direct solver $o(n)$ complexity for integral equations on one-dimensional domains*, Frontiers of Mathematics in China **7** (2012), 217–247, 10.1007/s11464-012-0188-3.

[5] L. Greengard, D. Gueyffier, P.G. Martinsson, and V. Rokhlin, *Fast direct solvers for integral equations in complex three-dimensional domains*, Acta Numer. **18** (2009), 243–275.

[6] S. Hao, A.H. Barnett, P.G. Martinsson, and P. Young, *High-order accurate nystrom discretization of integral equations with weakly singular kernels on smooth curves in the plane*, 2011, arXiv.org report #1112.6262.

[7] J. Helsing, *A fast and stable solver for singular integral equations on piecewise smooth curves*, SIAM Journal on Scientific Computing **33** (2011), no. 1, 153–174.

[8] J. Helsing and R. Ojala, *Corner singularities for elliptic problems: Integral equations, graded meshes, quadrature, and compressed inverse preconditioning*, Journal of Computational Physics **227** (2008), no. 20, 8820 – 8840.

[9] L. Ho and L. Greengard, *A fast direct solver for structured linear systems by recursive skeletonization*, SIAM J. on Scientific Computing **34** (2012), no. 5, A2507–A2532.

[10] P. Kolm and V. Rokhlin, *Numerical quadratures for singular and hypersingular integrals*, Comput. Math. Appl. **41** (2001), 327–352.

[11] P.G. Martinsson and V. Rokhlin, *A fast direct solver for boundary integral equations in two dimensions*, J. Comp. Phys. **205** (2005), no. 1, 1–23.